

1996

Characterization of gas pipeline defects using optimal radial basis function neural networks

Guoxin Xie

Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Oil, Gas, and Energy Commons](#), [Other Electrical and Computer Engineering Commons](#), and the [Signal Processing Commons](#)

Recommended Citation

Xie, Guoxin, "Characterization of gas pipeline defects using optimal radial basis function neural networks" (1996). *Retrospective Theses and Dissertations*. 16960.

<https://lib.dr.iastate.edu/rtd/16960>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

**Characterization of gas pipeline defects using
optimal radial basis function neural networks**

by

Guoxin Xie

A Thesis Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE

Department: Electrical and Computer Engineering
Major: Electrical Engineering

Signatures have been redacted for privacy

Signatures have been redacted for privacy

Iowa State University
Ames, Iowa

1996

TABLE OF CONTENTS

ACKNOWLEDGMENTS	iv
CHAPTER 1. INTRODUCTION	1
1.1 Background	1
1.2 Research Contribution	3
1.3 Scope of the Thesis	4
CHAPTER 2. MAGNETIC FLUX LEAKAGE (MFL) TECHNIQUE	7
2.1 General Principle of MFL Technique	7
2.2 Application of MFL Technique in Gas Pipeline Defect Characterization	14
2.3 Data Collection and Analysis	17
CHAPTER 3. RADIAL-BASIS FUNCTION (RBF) NEURAL NETWORK ...	21
3.1 Introduction to Neural Networks	21
3.2 RBF Neural Networks	26
3.3 Center Selection	31
3.3.1 Introduction	31
3.3.2 K-Means Algorithm	32
3.3.3 Optimal Adaptive K-Means Algorithm	34
CHAPTER 4. OPTIMAL TIME DOMAIN RBF NETWORKS	38
4.1 Steepest Descent and Optimal RBF Network	38
4.2 Defect Characterization in the Time Domain	42
4.3 Defect Characterization Using Defect Parameters	49
CHAPTER 5. OPTIMAL FREQUENCY DOMAIN RBF NETWORKS	54

5.1 Spectra of MFL Signals and Defect Profiles	54
5.2 Simulated Annealing Algorithm	59
5.2.1 The Metropolis Algorithm	59
5.2.2 The Simulated Annealing Algorithm	60
5.3 Optimal RBF Network Using Simulated Annealing	62
5.4 Modified Optimal RBF Network	68
CHAPTER 6. DISCUSSION AND CONCLUSION	72
6.1 Summary	72
6.2 Discussion and Future Work	73
REFERENCES	77

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Dr. Satish Udpa, for his invaluable guidance, help, encouragement, and understanding. It is you, Dr. Udpa, who helped me realize my dream of becoming an Electrical Engineer. There is an axiom in Chinese: "An advisor is a person who gives you advice in your life, gives you technologies for your work, and solves puzzles of your life." Thank you, Dr. Udpa for being such an advisor.

I sincerely would like to thank Dr. Lalita Udpa for her technical advice and invaluable assistance throughout this project. I also thank her for taking time to read and correct my thesis.

I would like to thank Dr. William Lord for his helpful comments and suggestion in this project. I thank Dr. William Meeker, Jr. for being on my graduate committee. I wish to thank Dr. Shreekanth Mandayam and Dr. Yushi Sun for their most appreciated assistance, advice and encouragement.

My sincere gratitude also goes to people who love me and have sacrificed so much for me. Among whom I especially would like to acknowledge are my wife, Bei Zhang, and my daughter, Jiulin, for their patience, constant support and encouragement, my father for all he has done for me, and my parents-in-law for their encouragement, support and invaluable advice. I do not know how to express my sincere appreciation to them. Without all that they have done for me, I wouldn't be where I am and who I am.

CHAPTER 1. INTRODUCTION

1.1 Background

Natural gas, as one of the main energy sources in the United States, is transported to consumer sites via a vast transmission pipeline network. The gas transmission system in the United States includes more than a million miles of pipelines. Developing and improving technologies to help gas pipeline companies maintain the physical integrity of the transmission systems, prevent pipeline shut down, and reduce maintenance costs is one of the most challenging issues in the industry.

The pipeline industry ensures the integrity and safe operation of pipeline systems through the in-line inspection of gas transmission pipelines. The inspection process can be traced back to the 1960's when Tuboscope introduced the first Linalog[®] corrosion survey service [1]. Currently a variety of nondestructive evaluation (NDE) techniques are either being developed or used for in-line inspection. One commonly used method employs the ultrasonic technique [2]. In this technique, high frequency ultrasonic waves are used to examine the condition of the pipe wall. However, the practical application of this method in a pipeline environment has proved to be problematical. Tests of ultrasonic tools have produced rather unsatisfactory results, which are caused by the complexity of the system, the need for processing enormous amounts of data as well as the need for using of elaborate methods to couple energy between the transducer and the pipe wall. The use of electromagnetic acoustic transducers (EMAT) has been suggested to overcome complications arising out of coupling problems [3, 4].

An alternative to the ultrasonic method and EMATs is based on the magnetic flux leakage (MFL) method. This technology has been applied extensively both in the United States as well as abroad. In this method, the pipeline wall is magnetized using a

permanent magnet or a direct current source. In the presence of surface breaking anomalies, the magnetic flux leaks out into the surrounding region [5, 6]. This leakage flux is picked up by sensors, such as Hall devices. The leakage flux signal contains information relating to the size and shape of the defect. The task of estimating the defect profile on the basis of information contained in the MFL signal is often referred to as an inverse problem. The process of mapping the MFL signal to the corresponding defect profile is illustrated in Figure 1.1. A variety of methods can be used to solve the inverse problem, i.e., to estimate the defect profile from the MFL signal. The techniques used for defect characterization rely largely on calibration methods. Such methods involve, as a first step, a series of experiments or models to establish the relation between the signal features and the defect parameters [7]. The relationship (or calibration curves) is then utilized to estimate the size of the defect. The performance of such approaches can be highly unsatisfactory since the calibration results are applicable only for a limited set of test conditions. Factors such as noise and the effect of variables such as tool velocity and variations in pipewall permeability can seldom be taken into account. The problem becomes even more complicated since the variables act in concert. A variety of statistical tools are being studied to identify the features that carry the most amount of information relating to the defects [7].

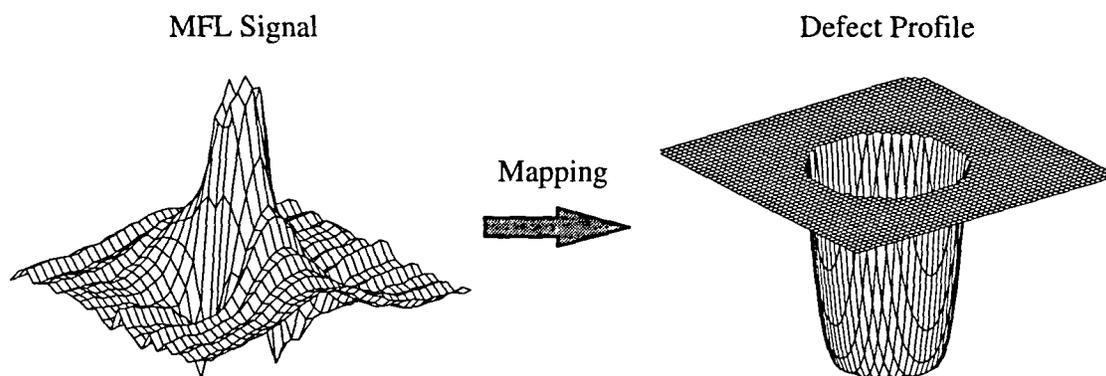


Figure 1.1 Inverse problem of obtaining defect profiles from MFL signals.

An alternative approach that has gained attention in recent years involves the use of artificial neural networks [8]. Such networks have been studied extensively since the last decade and impressive progress has been achieved in their applications. They are increasingly being looked at as signal processing tools for solving inverse problems. The fundamental objective in using artificial neural networks is to mimic the information processing strategies of the human brain and develop systems that have pattern recognition capability. Because of this, neural networks are widely used in many different areas, including nondestructive evaluation (NDE). In this application a neural network is trained initially using a set of well-characterized signals. The trained network is then capable of predicting a defect profile when a novel signal is presented.

Many different neural network paradigms have been proposed [9, 10]. Traditional neural networks, such as multi-layer perceptron (MLP) networks, need a great amount of training time. This has led to the development of more efficient neural networks. An example of such a network is a radial basis function (RBF) network. The architecture of RBF networks is fairly simple compared to other networks [10]. They are used extensively in many different applications.

RBF networks consist of three entirely different layers: an input layer, a hidden layer, and an output layer. As a tool for interpolation in high-dimensional space, an RBF network is able to interpolate between a set of data by fitting a proper hyper-surface in high dimension space. We exploit this property when solving inverse problems that involve mapping MFL signals to defect profiles.

1.2 Research Contribution

Work done to date has shown that RBF networks can be used to predict defect profiles in two dimensions [8]. This thesis extends the concept and focuses its attention on networks that are capable of estimating the full three dimensional profile of the defect. This is important since the defect is truly three dimensional.

The RBF network proposed in this thesis is optimal and has been designed using both steepest descent and simulated annealing algorithms. The initial parameters of the optimal RBF network are obtained using either the K-Means algorithm or an optimal adaptive K-Means method which can determine the number of the cluster centers automatically, based on the spatial distribution of training samples. A second new approach proposed in this thesis aims to improve the performance of the network by using the spectral coefficients of both MFL signals and defect profiles. The method is more efficient in that it avoids the use of a lot of redundant information contained in the spectrum. The technique uses only the low frequency components of MFL signals and defect profiles in the mapping process. An optimal RBF network employing a simulated annealing algorithm using this approach has been presented. The performance is shown to be superior compared to the steepest descent method in the time domain.

1.3 Scope of the Thesis

This thesis focuses on the problem of characterizing MFL signals using an optimal RBF artificial neural network. The research work described in this thesis attempts to improve existing MFL technology through several new methods for signal characterization. Figure 1.2 gives an overview of all the topics addressed in this thesis.

The organization of this thesis can be summarized as follows:

Chapter 1 provides a general introduction, a brief background, and motivation for the work described in this thesis. The fundamental principles understanding the MFL technique and its application to in-line gas pipeline inspection are given in Chapter 2. A description of the defect profiles used to evaluate the algorithm is also presented in this chapter.

Chapter 3 introduces artificial neural networks in general and provides a detailed discussion of RBF networks in particular. Since the choice of center locations and the

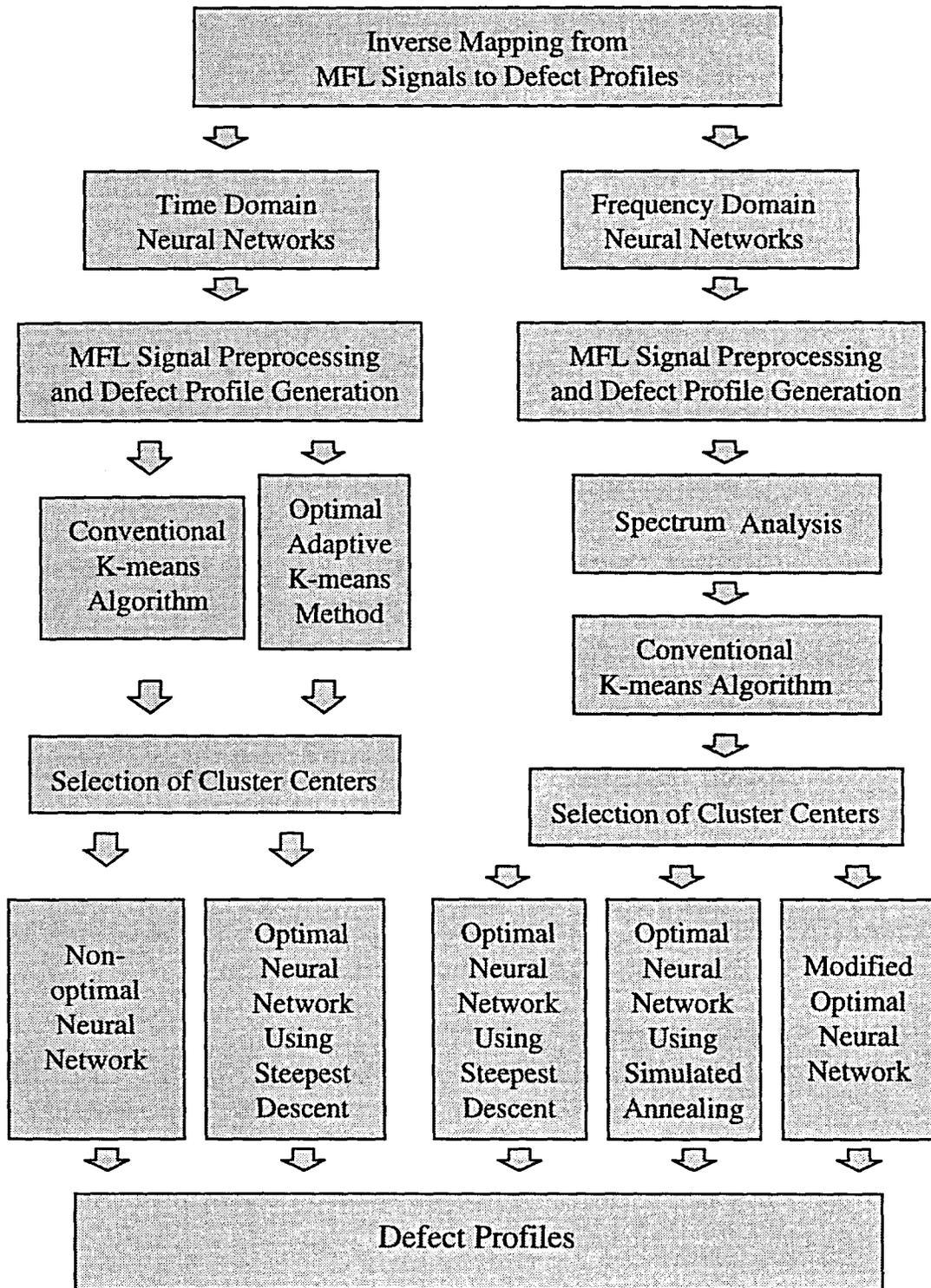


Figure 1.2 An overview of the research topics described in this thesis.

widths of radial basis functions have significant impact on the performance of such networks, two algorithms for selecting centers are described in this chapter. One algorithm uses a conventional K-Means algorithm, while the other uses an optimal adaptive K-Means method.

Chapter 4 describes an optimal RBF network whose parameters are derived using the steepest descent algorithm in the time domain. Two approaches are presented in this chapter. The first one maps MFL signals directly to defect profiles. The second approach, in contrast, maps the MFL signals to five geometrical parameters which can be used to reconstruct defect profiles, since the defects used in this study have regular shapes.

Chapter 5 gives an alternative approach to characterize MFL signals which is carried out in the frequency domain. In this chapter, a procedure for mapping the spectra of MFL signals to the spectra of defect profiles is presented. Methods used for deriving the RBF network parameters include simulated annealing, as well as the steepest descent method. The simulated annealing procedure is discussed in detail in this chapter.

Chapter 6 provides a summary of the research described in the thesis, as well as a discussion of potential areas of future work relating to defect characterization.

CHAPTER 2. MAGNETIC FLUX LEAKAGE (MFL) TECHNIQUE

2.1 General Principle of MFL Technique

Nondestructive evaluation methods are used to inspect and evaluate engineering structure in a manner that does not adversely affect the materials serviceability. Since the tests are nondestructive, every component can be inspected to ensure uniform quality. There are two roles that NDE techniques can play. One is related to their use in manufacturing environments for ensuring the quality of products, and as a tool for process control. The other is related to their role in preventing failures in aging structures by detecting flaws in their incipient stage. This thesis is primarily concerned with the latter role. A diverse set of NDE methods have developed to meet a variety of applications, such as ultrasonic and electromagnetic techniques [11]. Figure 2.1 illustrates the major components of NDE systems. Major areas of research emphasis include the development of solutions to forward and inverse problems. The forward models are concerned with the prediction of the transducer response to a known defect shape and a known excitation function, while the inverse problem is related to the task of characterizing the structural defect based on information contained in the transducer response.

Electromagnetic nondestructive testing is one of the most commonly used NDE methods in the metals industry to solve a variety of material and product evaluation problems [12]. Electromagnetic NDE techniques use the effects of electromagnetic induction, electromagnetic field, or varying currents for probing, measuring, or inspecting the material under test. One of the most commonly used electromagnetic NDT techniques is the magnetic flux leakage (MFL) method which uses electromagnetic sensors to detect the magnetic leakage field caused by the presence of defects in a magnetized test specimen [5].

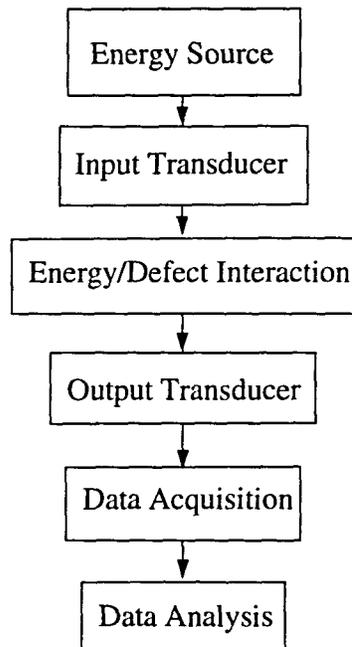


Figure 2.1 A schematic diagram of a “generic” NDE system [11].

The magnetic flux leakage field method is based on the fact that a near surface discontinuity in the geometry or a change in the magnetic properties of a magnetized specimen produces a localized perturbation in the magnetic field just outside the surface of the specimen. Therefore, the presence of a defect, nonmagnetic impurity or other localized anomalies can be detected by observing the magnetic field at the surface of a specimen.

It is known that the presence of a defect in a magnetized ferromagnetic bar will cause the flux to leak out into the surroundings [5, 6]. A qualitative explanation of the origin of the flux leakage field is illustrated in Figure 2.2 [5]. When a uniform ferromagnetic bar without any defect is magnetized with a uniform magnetic field \mathbf{H} , the magnetic flux density, \mathbf{B} , inside the bar is constant. At the operating point near or beyond the saturation point, the permeability of the bar usually is much larger than that of the surroundings. Therefore, the magnetic inductance \mathbf{B} at the outer surface is negligible.

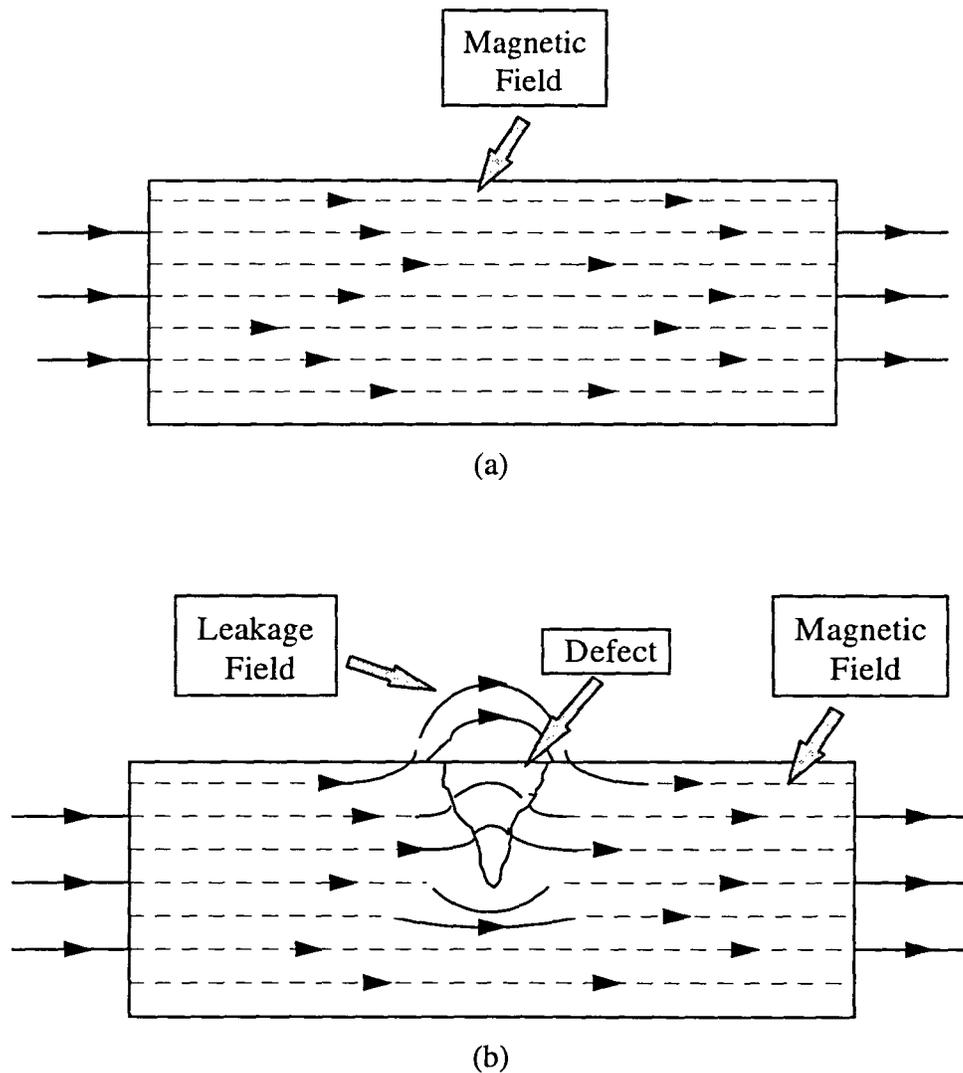


Figure 2.2 (a) The magnetic field in a uniformly magnetized bar (b) The magnetic field in a uniformly magnetized bar with a defect [5].

When a defect is present inside the bar, the flux density below the defect increases. The increase in \mathbf{B} causes a great increase in \mathbf{H} , since the operating point is close to or beyond the saturation point. Consequently the permeability, μ , will decrease greatly, and the permeability of the surrounding non-magnetic material is no longer small compared to that of the ferromagnetic bar. Hence the magnetic flux “leaks” out.

In order to gain a better understanding of the magnetic flux leakage theory, consider two of Maxwell's equations describing static magnetic fields. In this situation the magnetic field can be described by:

$$\nabla \times \mathbf{B} = \mathbf{J} \quad (2.1)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (2.2)$$

where \mathbf{B} is the magnetic-flux density, and \mathbf{J} is the current density.

From a macroscopic viewpoint, these two equations should be replaced by [13]:

$$\nabla \times \mathbf{H} = \mathbf{J} \quad (2.3)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (2.4)$$

$$\mathbf{H} = \mathbf{B} / \mu_0 - \mathbf{M} \quad (2.5)$$

where \mathbf{M} is the average macroscopic magnetization or magnetic moment density, \mathbf{H} is the magnetic field and χ is susceptibility. The constitutive relation between \mathbf{H} and \mathbf{B} for isotropic diamagnetic and paramagnetic materials can be expressed using the following linear equation:

$$\mathbf{B} = \mu \mathbf{H} \quad (2.6)$$

where μ is the magnetic permeability that is characteristic of the medium. However, for ferromagnetic substances, the above equation is not true, and must be replaced with the following nonlinear functional relationship:

$$\mathbf{B} = f(\mathbf{H}) \quad (2.7)$$

A typical B-H curve embodying this relationship is shown in Figure 2.3.

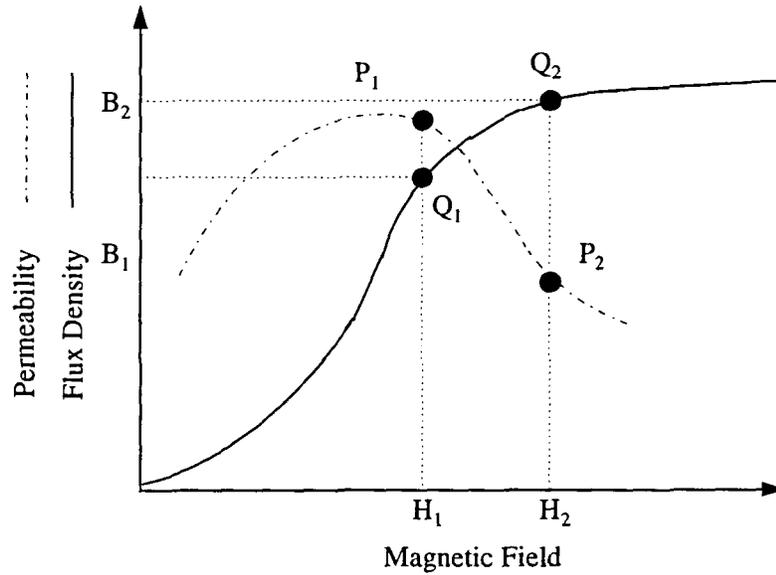


Figure 2.3 The nonlinear relationship between flux density and magnetic field [5].

By applying Stokes theorem to equation (2.3) and Gauss's theorem to equation (2.4), the boundary conditions for \mathbf{B} and \mathbf{H} are given by [13]:

$$\mathbf{n} \times (\mathbf{H}_2 - \mathbf{H}_1) = 0 \quad (2.8)$$

$$\mathbf{n} \cdot (\mathbf{B}_2 - \mathbf{B}_1) = 0 \quad (2.9)$$

where \mathbf{n} is the normal to the surface, and the surface current density is assumed to be zero.

From equation (2.8) and (2.9), it can be seen that the normal components of the magnetic-flux density \mathbf{B} , and the tangential components of magnetic field \mathbf{H} are continuous, i.e.,

$$B_{1n} = B_{2n} \quad (2.10a)$$

$$H_{1t} = H_{2t} \quad (2.10b)$$

In order to look at the MFL phenomena in great detail, let A denote the cross sectional area of the bar, and a be the cross sectional area of the defect. Therefore, the cross sectional area without the defect in the vicinity of the defect is reduced to $(A - a)$. Let the magnetic flux density in the defect-free portion of the bar be \mathbf{B}_1 , when the bar is placed in a uniform magnetizing field \mathbf{H} . The corresponding operating point is at Q_1 , and the corresponding value of permeability is P_1 (shown in figure 2.3 [5]). The total flux is then given by

$$\Phi_1 = B_1 \cdot A \quad (2.11)$$

From the continuity equation (2.10), the flux density at the outer surface of the bar, B_L , is

$$B_L = B_1 \cdot \frac{\mu_s}{\mu_1} \quad (2.12)$$

Since usually μ_s is much smaller μ_1 when the operating point is close to the saturation point, it can be seen that the flux leakage is negligible.

If the magnetic flux \mathbf{B}_1 passes through a region with a defect, the flux density in this region increases to

$$B_2 = B_1 \cdot \frac{A}{A - a} \quad (2.13)$$

Consequently, the operating point will shift to Q_2 and the corresponding permeability will be reduced to P_2 . Following the same discussion, it can be seen that the flux density at the outer surface of the bar is no longer negligible and therefore can be detected using a Hall device.

As mentioned earlier, inverse problems are concerned with the task of characterizing defects on the basis of information contained in the transducer signal.

Direct methods for solving the inverse problem are limited to an extremely limited class of defects where the shape is known a priori [41].

Alternative approaches that have a phenomenological basis involve the use of a forward model in the feedback arrangement. Here, the error representing the difference between the signal generated by the forward model and the actual measurement is used to update the defect shape in an iterative scheme. Forward models employed to date include both finite element [17] and boundary integral based methods [42]. Such approaches, however, tend to be computationally intensive and difficult to implement in practice.

One of the most popular techniques used in industry involves the use of calibration methods. In this approach, empirical relationships between the defect parameters and the measured leakage field signal feature are established on the basis of data collected from a large number of experiments. These methods are, however, not robust.

These limitations have resulted to the using of finite element method (FEM) which is used to model the magnetic field distribution and estimate the defect characteristics [14]. This method, combined with artificial intelligence technique, reduces the general inverse problem to a tree or state space search [18].

Usually, the forward methods can give us very accurate estimate of defect parameters [19]. However, there is too much computation in this method. This backward has resulted to the use of inverse methods. In this method, empirical relationships between the defect parameters, such as depth, and the features of the measured leakage field, like the amplitude, are built using a great number of experimental data [20, 21]. These relationships can be used to estimate defect profile from the detected leakage field. The biggest advantage of this method is its simplicity. It only requires one time of calculation. However usually the estimated defect profiles are much less accurate than those of forward methods. Recently, some sophisticated inverse methods have been developed, such as neural networks which will be discussed in next chapter.

2.2 Application of MFL Technique in Gas Pipeline Defect Characterization

Natural gas, which is one of the nation's cheapest forms of energy, is transported to consumer sites via a vast transmission pipeline network, shown schematically in Figure 2.4. The major components of this system include approximately 275,000 miles of transmission pipelines, 90,000 miles of gathering lines, underground storage systems, compressor stations, regulator stations, transfer facilities, approximately 836,000 miles of distributions mains, and 280,000 miles of services lines [22, 23]. These components provide gas to about 51 million customers. Safety considerations and a desire to assure uninterrupted energy supply require that the pipelines be inspected periodically. Over \$2.7 billion is spent annually on operating and maintenance activities, reflecting the importance and the need for improving technologies used in in-line inspection [14]. In-line inspection technology is one of the important tools being used by pipeline operators to ensure the integrity and safe operation of the natural gas pipeline system.

Magnetostatic methods of NDE are used extensively for the inspection of ferromagnetic specimens. Magnetic powders were used widely in industry as part of the MFL method to detect flaws and defects. The magnetic powder is used to detect the presence of field concentrations on the surface of the specimen. The presence and shape of a defect are indicated by local aggregations of magnetic powder. However, the depth of the defect cannot be determined. Other sophisticated techniques have to be employed to detect defects more precisely and accurately. An alternate approach that is used extensively in in-line inspection of gas pipelines relies on the use of Hall elements and/or coils for detecting the leakage field.

The magnetic flux leakage (MFL) technique is a two step procedure. First the inspection is carried out using a vehicle called the pig (Figure 2.5). The pig, which moves inside the pipeline under the pressure of natural gas, consists of one or more permanent magnets to magnetize the pipe. The magnetic field is coupled to the pipe

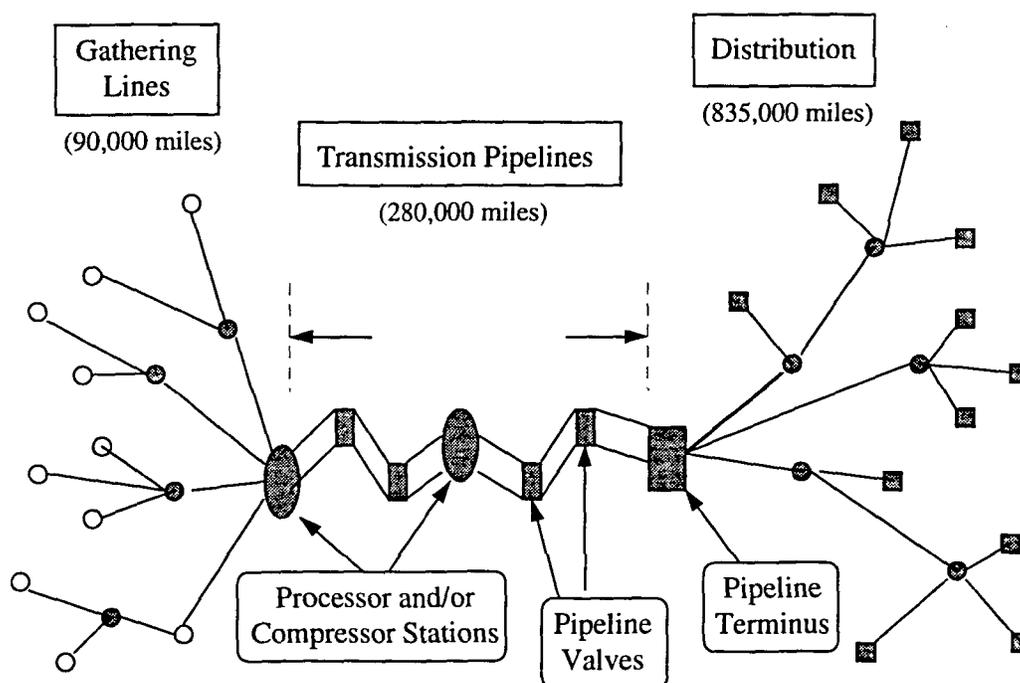


Figure 2.4 Natural gas transmission pipeline system [12].

through brushes which allow intimate contact between the pipe and the magnetic yoke and yet allow the pig to move freely within the pipe. A circumferential array of Hall effect sensors or coils is used to measure the leakage flux. When the pig moves inside the gas pipe, it magnetizes the ferromagnetic pipe wall between the two brushes. The presence of a defect in the pipe-wall results in a redistribution of magnetic field in the vicinity of the defect, and causes the magnetic field to leak out to the surroundings. The axial and radial components of the magnetic leakage flux signal are detected using the Hall effect sensor. Figure 2.6. shows the axial component of such a magnetic leakage field of a defect on gas pipelines.

The next operation required in a pipeline inspection process is to record the detected signal. The pig contains a microprocessor based data acquisition system to digitize and store the data. The data then is subsequently retrieved and analyzed off-line for characterizing the condition of the pipe.

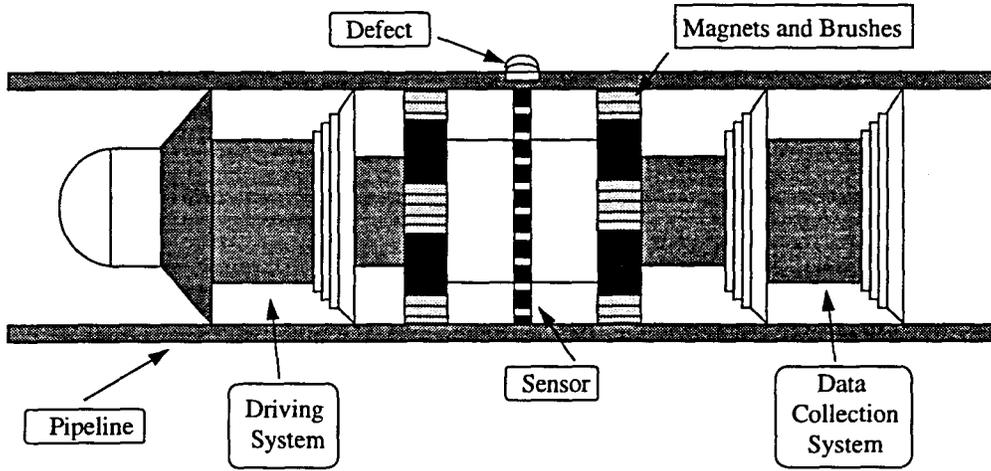


Figure 2.5 A schematic diagram of a pig used in in-line inspection.

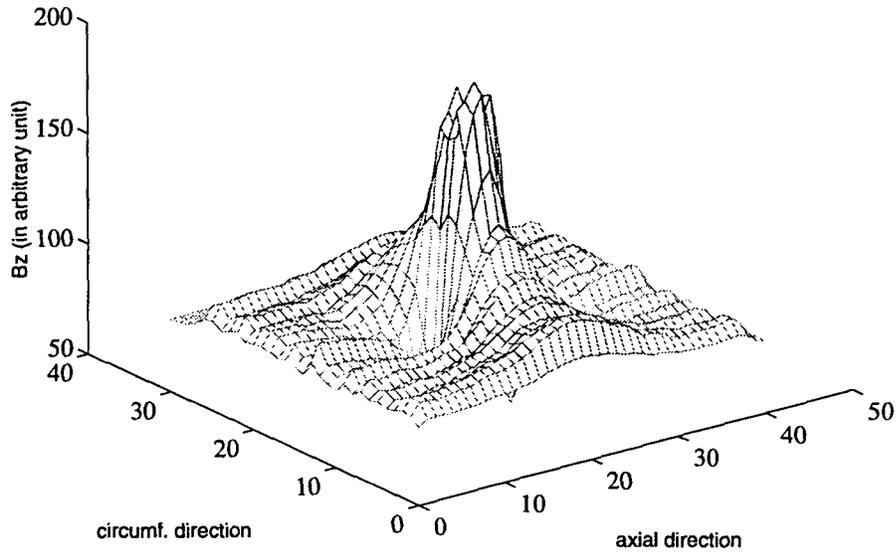


Figure 2.6 Typical axial component of an MFL signal.

2.3 Data Collection and Analysis

The MFL signals used for training and testing the neural network described in this thesis were generated by Vetco Pipeline Services Inc., Houston TX employing the experimental equipment as described previously. A point about these data that needs to be mentioned is the spatial size of the MFL signals. Since defects with different size will produce different kinds of MFL signals they need to be normalized to the same size in order to be analyzed by the RBF neural network. This means some of them have to be zero padded appropriately.

As mentioned previously, the defects used in this research were machined flaws [22]. A generic method was developed to generate these defects. They can be characterized in term of six parameters: length (L), width (W), depth (D), top sharpness (T), bottom sharpness (B), and roundness (R). Figure 2.7 defines these dimensions.

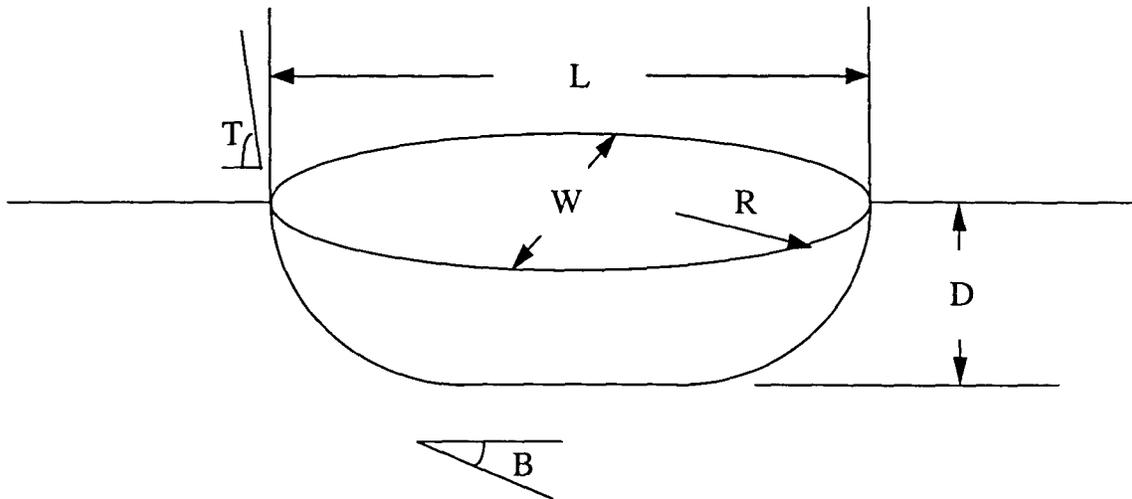


Figure 2.7 Six geometric parameters characterizing metal loss defect.

For simplicity, in this research project, all defects have zero degree bottom sharpness. Therefore, five geometric parameters can define a defect profile. The outer edge of the bottom of the defect is described by the following equation [22]:

$$\left(\frac{x}{BL/2}\right)^R + \left(\frac{y}{BW/2}\right)^R = 1 \quad (2.14)$$

where

(0,0) is the center of the defect bottom,

x is the distance from the defect center in the axial direction,

y is the distance from the defect center in the circumferential direction,

BW is the width of the flat bottom at its widest point,

BL is the length of the defects at its longest point, and

R is the roundness of the defect. ($R = \infty$ for rectangular shape, $R = 2$ for ellipse shape, and $R = 1$ for diamond shape).

Using these five geometrical parameters, a defect profile can be constructed. A typical defect profile reconstructed from the five parameters is shown in Figure 2.8. Table 2.1 describes the geometric parameters of the defects presented in the set.

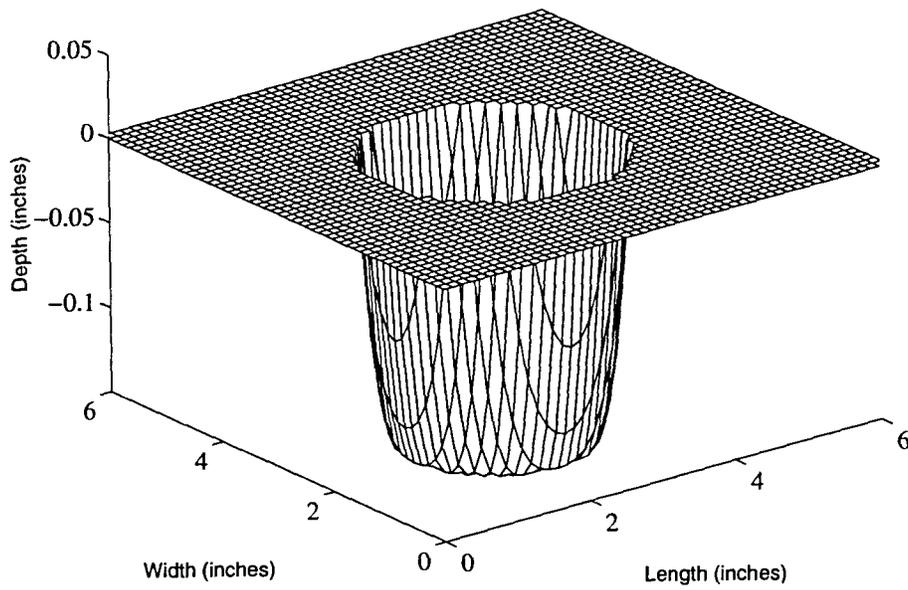


Figure 2.8 Typical defect profile constructed using five geometrical parameters.

Table 2.1 List of defects used [22].

Defect Number	Length (inches)	Width (inches)	Depth (%)	Surface Angle (degree)	Roundness
1	3.0	6.0	65.0	45.0	2.0
2	4.5	6.0	50.0	45.0	2.0
3	2.0	6.0	50.0	45.0	2.0
4	2.0	6.0	80.0	45.0	2.0
5	2.0	1.0	80.0	45.0	2.0
6	4.5	6.0	80.0	45.0	2.0
7	1.0	1.0	65.0	45.0	2.0
8	6.0	6.0	65.0	45.0	2.0
9	6.0	3.0	65.0	45.0	2.0
10	3.0	6.0	35.0	45.0	2.0
11	3.0	3.0	50.0	45.0	2.0
12	6.0	4.5	50.0	45.0	2.0
13	3.0	3.0	80.0	45.0	4.0
14	3.0	6.0	50.0	23.0	2.0
15	3.0	6.0	50.0	45.0	4.0
16	6.0	4.5	80.0	45.0	2.0
17	3.0	3.0	50.0	45.0	2.0
18	3.0	6.0	50.0	90.0	2.0
19	3.0	6.0	50.0	45.0	2.0
20	1.0	1.0	80.0	45.0	2.0
21	1.0	1.0	50.0	45.0	2.0
22	6.0	3.0	80.0	45.0	2.0
23	3.0	3.0	65.0	45.0	2.0
24	3.0	3.0	80.0	45.0	2.0
25	6.0	6.0	80.0	45.0	2.0
26	3.0	3.0	50.0	45.0	2.0
27	6.0	6.0	80.0	90.0	2.0
28	3.0	1.0	80.0	23.0	2.0
29	4.5	3.0	80.0	45.0	2.0
30	3.0	3.0	80.0	90.0	2.0
31	3.0	6.0	20.0	90.0	2.0
32	3.0	3.0	50.0	90.0	2.0
33	3.0	3.0	50.0	45.0	2.0
34	3.0	3.0	80.0	23.0	2.0
35	2.0	3.0	80.0	45.0	2.0
36	3.0	3.0	50.0	45.0	2.0
37	3.0	4.5	80.0	45.0	2.0
38	6.0	6.0	80.0	45.0	4.0
39	3.0	6.0	20.0	45.0	4.0
40	3.0	4.5	50.0	45.0	2.0
41	6.0	6.0	20.0	45.0	4.0
42	3.0	2.0	50.0	45.0	2.0
43	6.0	1.0	80.0	45.0	4.0
44	3.0	3.0	50.0	45.0	4.0
45	3.0	2.0	80.0	45.0	2.0
46	3.0	1.0	80.0	45.0	4.0
47	6.0	1.0	80.0	23.0	2.0
48	6.0	3.0	20.0	45.0	4.0

CHAPTER 3. RADIAL-BASIS FUNCTION (RBF) NEURAL NETWORK

3.1 Introduction to Neural Networks

Neural networks, which have been studied for many years in the hope of achieving human-like performance in the field of speech and image recognition, represent an emerging technology rooted in many disciplines. Such networks are endowed with some unique attributes: universal approximation (input-output mapping), ability to learn from and adapt to their environments, and the ability to invoke weak assumptions about the underlying physical phenomena responsible for the generation of the input data.

In the most general form, a neural network is a machine that is designed to model the way in which the human brain performs a particular task or implement a function of interest. The neural network can be defined as a massively parallel distributed processor that has a natural property for storing experiential knowledge and making it available for use [10]. It simulates the human brain in two respects:

- (1) Knowledge is acquired by the network through a learning process.
- (2) Interneuron connection strength known as synaptic weights are used to store the knowledge.

Models of neural networks are composed of many nonlinear computational elements operating in parallel, and arranged in patterns reminiscent of biological neural networks. The fundamental information-processing unit of a neural network is the neuron, which plays the same function, very often, as that of a neuron in the human brain. Figure. 3.1 shows a nonlinear model of a neuron [10].

Mathematically, a neuron can be described by the following pair of equations:

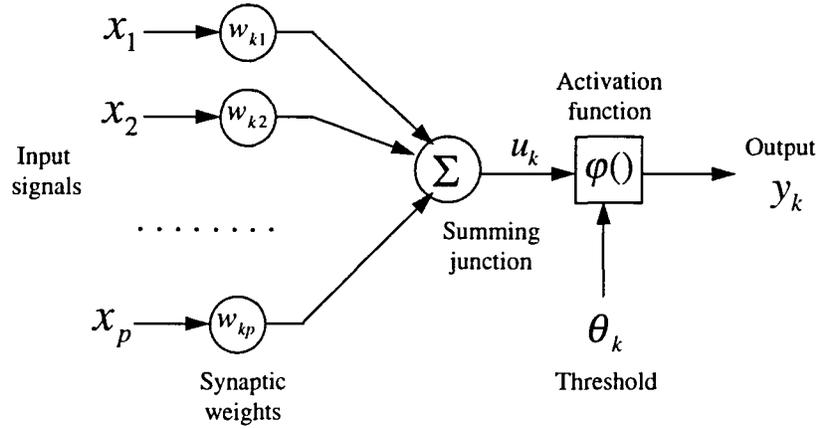


Fig. 3.1 Nonlinear model of a neuron [10].

$$u_k = \sum_{j=1}^p w_{kj} x_j \quad (3.1)$$

$$y_k = \phi(u_k - \theta_k) \quad (3.2)$$

where x_1, x_2, \dots, x_p are the input signals; $w_{k1}, w_{k2}, \dots, w_{kp}$ are the synaptic weights of neuron k , u_k is the linear combined output, θ_k is the threshold, $\phi(\cdot)$ is the activation function, and y_k is the output signal of the neuron. The nonlinear nature of neurons allows neural networks to estimate and derive functions that cannot be explicitly stated using precise mathematical definitions.

Three types of activation functions as shown in Figure 3.2 are employed [10]:

1. Threshold Function (Figure 3.2a)

$$\phi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \quad (3.3)$$

The output of neuron k is

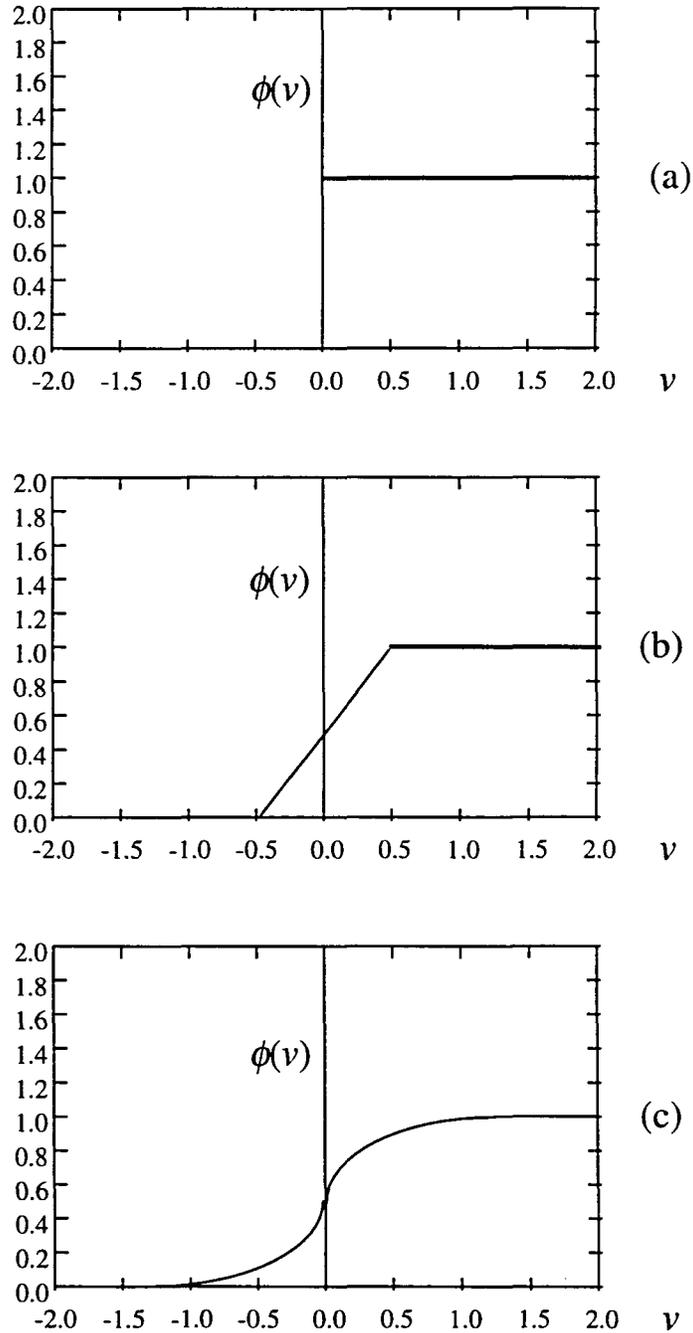


Figure 3.2 Three types of activation function [10]: (a) Threshold. (b) Piecewise-linear. (c) Sigmoid function.

$$y_k = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \quad (3.4)$$

where

$$v_k = \sum_{j=1}^p w_{kj} x_j - \theta_k \quad (3.5)$$

2. Piecewise-linear Function (Figure 3.2b)

This function may be treated as an approximation to a nonlinear amplifier.

$$\phi(v) = \begin{cases} 1 & \text{if } v \geq \frac{1}{2} \\ v & \text{if } -\frac{1}{2} < v < \frac{1}{2} \\ 0 & \text{if } v \leq -\frac{1}{2} \end{cases} \quad (3.6)$$

3. Sigmoid Function (Figure 3.2c)

This is the most commonly used activation function in the construction of a neural network. It has several forms. An example of such a function is:

$$\phi(v) = \frac{1}{1 + \exp(-av)} \quad (3.7)$$

where a is the slope parameter of the sigmoid function.

A neural network consists of multiple sets of neurons that are densely interconnected (Fig. 3.3 shows a typical architecture of a neural network). Several different types of neural network architecture have been proposed and are primarily distinguished by their architecture and the learning procedure. Examples include the Hopfield network, the Hamming network, multilayer perceptron (MLP), radial basis function (RBF) network, and wavelet basis function network. Each neural network has its own set of

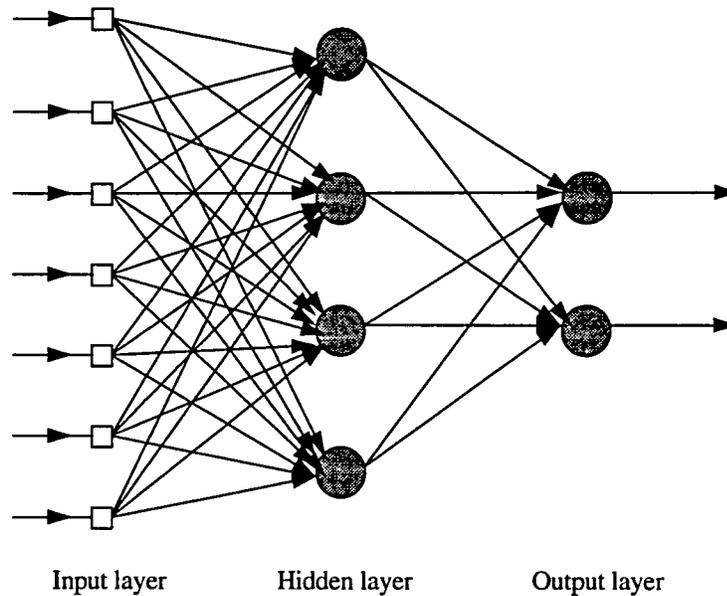


Figure 3.3 The architecture of a typical neural network.

advantages and disadvantages [10]. In this research project we use an RBF network [24].

The ability to learn is one of many interesting properties of a neural network that allows it to improve its performance. To study the issue further, consider a pair of node signals x_j and v_k connected by the weight w_{kj} as shown in Figure 3.1. Let $w_{kj}(n)$ denote the value of the synaptic weight w_{kj} at time n . At this time, an adjustment $\Delta w_{kj}(n)$ is applied to the synaptic weight $w_{kj}(n)$, which updates the value of weights at time $n+1$:

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \quad (3.8)$$

$w_{kj}(n+1)$ can be viewed as the new value the synaptic weight w_{kj} learned from the environment. A variety of learning algorithms have been developed to suit different types of neural networks [9, 10].

3.2 RBF Neural Networks

As one of the most commonly used neural network, an RBF neural network can be described as a tool for curve-fitting (approximation) in a high-dimension space. From this point of view, learning is equivalent to finding a surface in a high-dimension space that provides the best fit to the training data. In other words, the problem is equivalent to the generation of a multi-dimensional surface to interpolate the training data.

The architecture of a radial-basis function (RBF) network in its most basic form consists of three different layers (Figure 3.4). The input layer is made up of source nodes (sensory units). The second layer is a hidden layer of a sufficiently high dimensions, while the output layer supplies the response of the network to the activation patterns applied to the input layer. The transformation from the input space to the hidden-unit space is nonlinear, while the transformation from the hidden-unit space to the output space is linear.

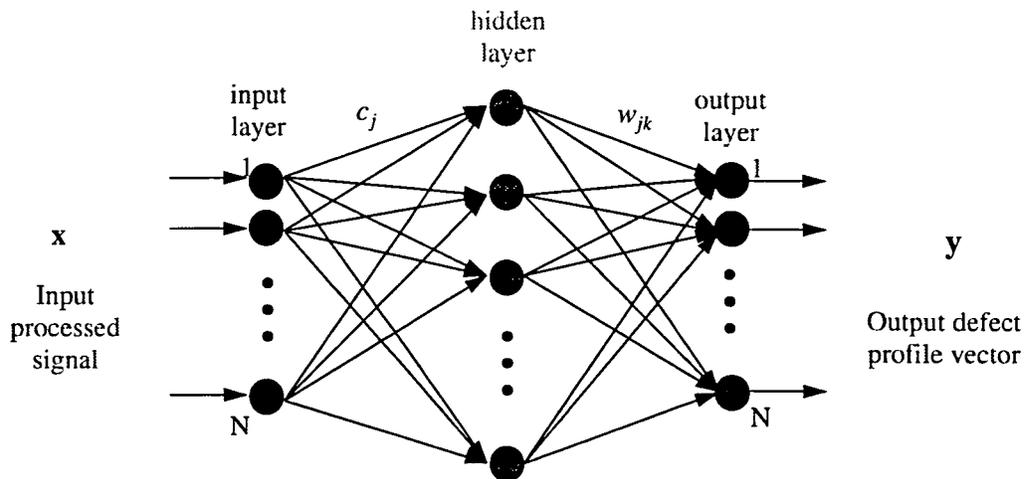


Figure 3.4 Radial basis function (RBF) neural network architecture.

Mathematically, the RBF network can be described as follows [25]: Given a set of N different points $\{\mathbf{x}_i \in R^p | i = 1, 2, \dots, N\}$ and a corresponding set of N real numbers $\{d_i \in R^1 | i = 1, 2, \dots, N\}$, we wish to find a function $F: R^N \rightarrow R^1$ that satisfies the interpolation condition:

$$F(\mathbf{x}_i) = d_i \quad i = 1, 2, \dots, N \quad (3.9)$$

The RBF technique consists of choosing a function $F(\cdot)$ that has the following form [26]

$$F(\mathbf{x}) = \sum_{i=1}^N w_i \phi(\|\mathbf{x} - \mathbf{c}_i\|) \quad (3.10)$$

where $\{\phi(\cdot) | i = 1, 2, \dots, N\}$ is a set of N arbitrary nonlinear functions, serving as radial-basis functions, and $\|\cdot\|$ denotes a norm that is usually taken to be the Euclidean distance. The known data points $\mathbf{c}_i \in R^p, i = 1, 2, \dots, N$ are taken to be the centers of the radial basis functions.

When a Gaussian RBF is chosen, the function $F(\cdot)$ can be written as

$$F(\mathbf{x}_i) = \sum_{j=1}^N w_j \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2\sigma_j^2}\right) \quad (3.11)$$

where \mathbf{c}_j is the center for j th cluster and σ_j is the variance of the j th radial basis function. The variance term σ_j is an important parameter since it determines the localization of the Gaussian radial basis function. Therefore, it is often called the width of a Gaussian radial basis function.

From equations (3.9) and (3.10), we can have the following set of linear equations from which the unknown coefficients (weights) can be obtained:

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1N} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ \phi_{N1} & \phi_{N2} & \cdots & \phi_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \cdots \\ d_N \end{bmatrix} \quad (3.12)$$

where $\phi_{ij} = \phi(\|\mathbf{x}_i - \mathbf{c}_j\|)$, $i, j = 1, 2, \dots, N$.

Let

$$\mathbf{X} = [d_1, d_2, \dots, d_N]^T \quad (3.13)$$

and

$$\mathbf{W} = [w_1, w_2, \dots, w_N]^T \quad (3.14)$$

Equation (3.11) can be simply expressed as

$$\Phi \mathbf{W} = \mathbf{X} \quad (3.15)$$

From this matrix equation, the synaptic weights can be derived using

$$\mathbf{W} = \Phi^{-1} \mathbf{X} \quad (3.16)$$

It has been shown by Micchelli that for all positive integers p and N and for a large class of functions $\phi(\cdot)$ the matrix Φ is nonsingular if the data points are distinct [25].

However, the above discussion is true only for the case in which the number of data points in the high-dimension space is equal to the number of centers. It is possible that the number of data points exceeds the number degrees of freedom, resulting in a problem that is over determined. It is also possible that the number of data points is less than the number of degrees of freedom, in which case the problem becomes under-specified. In both cases, the matrix Φ is not a square matrix, a unique inverse matrix no longer exists, and the task becomes a linear optimization problem. One of the alternatives is to use the Moore-Penrose pseudo-inverse of a matrix [25, 27], i.e.

$$\mathbf{W} = \Phi^+ \mathbf{X} \quad (3.17)$$

where Φ^+ is the pseudo-inverse of matrix Φ , and is defined as

$$\Phi^+ = (\Phi^T \cdot \Phi)^{-1} \cdot \Phi^T \quad (3.18)$$

where Φ^T is the transpose of Φ .

The procedure for training the RBF network involves the presentation of the input-output data pairs shown in Figure 3.5. A strict interpolation procedure may not be a good strategy for training an RBF network for certain classes of tasks due to the problems of poor generalization. The fitted curve usually is not required to pass through all input-output pairs used in the training phase. This is in contrast to recognition problems where the neural networks try to reproduce a pattern seen previously. This kind of learning procedure results in simpler neural networks relative to neural networks that are trained using other learning methods [28].

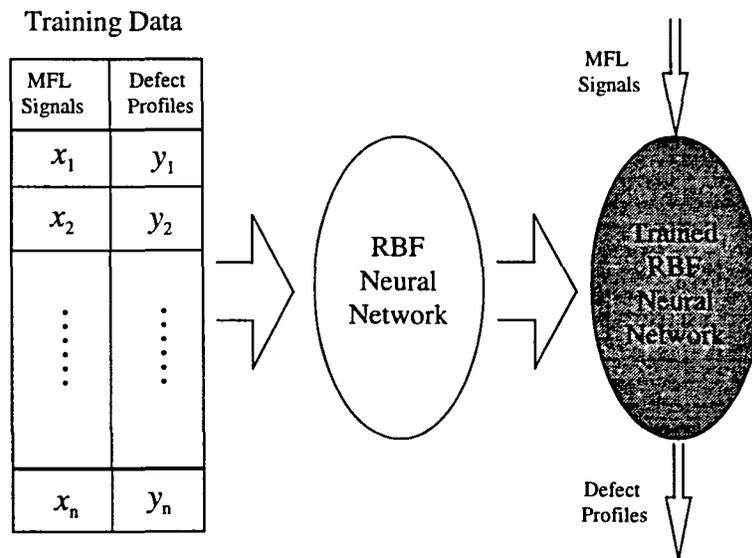


Figure 3.5 Procedure used for training the RBF neural network.

Several learning strategies can be used for training an RBF network, depending on the method used for specifying the centers of the RBF network [10].

The simplest approach is to employ a fixed set of centers. The locations of the RBF centers are chosen randomly from the training data set. A simple choice for the RBF is as follows [10]:

$$F(\|\mathbf{x} - \mathbf{c}\|^2) = \exp\left(-\frac{n}{d^2}\|\mathbf{x} - \mathbf{c}\|^2\right) \quad (3.19)$$

where n is the number of centers and d is the maximum distance between the chosen centers. As long as the centers are known, the weights can be determined by employing the pseudo-inverse method as described previously.

An alternate strategy is to employ a self-organizing or clustering algorithm for selecting the centers. In this approach, the radial basis functions are allowed to move the locations of their centers in a self-organized fashion. This can be accomplished by using the K-Means algorithm to obtain the proper centers. Additional details are provided in the next section.

A third alternative is to employ a supervised scheme for determining the centers. In this approach, we begin by identifying an appropriate cost function. A reasonable choice for the cost function is:

$$E = \frac{1}{2} \sum_{i=1}^N e_i^2 \quad (3.20)$$

where N is the number of training examples, and e is the error function which is defined as:

$$\begin{aligned} e_i &= y_{0i} - y_i \\ &= y_{0i} - \sum_{j=1}^n w_j \phi\left(\frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2\sigma_j^2}\right) \end{aligned} \quad (3.21)$$

The objective is to find the parameters w_j, c_j , and σ_j such that the cost function is minimized.

It is also possible to use a procedure that combines all or some of the above methods. As an example, this thesis reports on a method that uses the K-Means method to select initial centers and employs the third strategy to find the optimal centers, widths, and weights of the radial basis functions.

3.3 Center Selection

3.3.1 Introduction

When the radial basis function (RBF) network is used to characterize the defect profile, the network is trained by finding the centers, variances (widths) and the weights connecting the hidden nodes to the output nodes. Determining the proper number of hidden nodes (cluster centers) for the radial basis functions, their specific location, and widths is of fundamental importance, since they provide the basis for solving the interpolation problem. The obvious choice would be to place one Gaussian radial basis function at every training point, making the matrix Φ in equation (3.14) a square matrix. Assuming the width of each radial basis function (that is the deviation of the radial basis function) is known, the problem becomes trivial. The number of equations and the number of unknowns are both equal to the number of training samples. Consequently the equation can be easily solved for the output weights. However this is often not possible. If the number of the training samples is large, the memory and computational requirements become prohibitive. The computational burden can be reduced by reducing the number of RBF nodes by carefully choosing the centers and widths of the radial basis functions. One of the methods that is commonly used is the K-Means algorithm.

The K-Means algorithm represents one of the most popular methods for clustering data. There exist several varieties of the algorithm. Examples of such varieties include Forgy's method [29], Macqueen's algorithm [30], and the Spath's algorithm [31]. All of

these approaches represent competitive learning algorithms that cluster samples based on the Euclidean distance. The motivation for using distance functions as a classification criterion follows naturally from the fact that the most obvious way of establishing a measure of similarity between sample vectors, is to determine the distance from each other.

However, the performance of conventional K-Means algorithm is influenced by the choice of the preset number of the cluster centers. This limits the capability of K-Means clustering algorithm. In this thesis, an optimal adaptive K-Means algorithm is used. The algorithm allows the number of cluster centers to be determined automatically based on the spatial distribution of the training data points. A brief description of the classical K-Means algorithm will be given in the following section before the new adaptive approach is presented.

3.3.2 K-Means Algorithm

The objective of the K-Means algorithm is to partition the domain of training vectors (samples) into m regions, and then find a reference vector that best represents the pattern vectors in each region. When the training samples are linearly separable, the K-Means algorithm offers reasonable results. The clustering technique partitions the domain of interest by minimizing a performance index which is defined as the sum of the squared distances from all points in a cluster domain to the cluster center. The K-Means clustering procedure consists of the following steps [32]

Step 1. The m initial cluster centers $\mathbf{c}_1(1), \mathbf{c}_2(1), \dots, \mathbf{c}_m(1)$ are chosen arbitrarily. They are usually the first m samples of the given sample set.

Step 2. At the k th iterative step, the samples $\{\mathbf{x}\}$ are distributed among the m cluster domains, using the following relation,

$$\mathbf{x} \in S_j(k) \quad \text{if} \quad \|\mathbf{x} - \mathbf{c}_j(k)\| < \|\mathbf{x} - \mathbf{c}_i(k)\| \quad (3.22)$$

for all $i = 1, 2, \dots, m, i \neq j$, where $S_j(k)$ denotes the set of samples whose cluster center is $\mathbf{c}_j(k)$.

Step 3. The new cluster centers $\mathbf{c}_j(k+1)$, $j = 1, 2, \dots, m$, are computed such that the sum of the squared distances from all points in $S_j(k)$ to the new cluster center is minimized. In other words, the new cluster center $\mathbf{c}_j(k+1)$ is computed so that the performance index

$$J_i = \sum_{\mathbf{x} \in S_j(k)} \|\mathbf{x} - \mathbf{c}_j(k+1)\|^2, \quad j = 1, 2, \dots, m \quad (3.23)$$

is minimized. $\mathbf{c}_j(k+1)$, the new cluster center which minimizes this performance index, is given by

$$\mathbf{c}_j(k+1) = \frac{1}{n_j} \sum_{\mathbf{x} \in S_j(k)} \mathbf{x}, \quad j = 1, 2, \dots, m \quad (3.24)$$

where n_j is the number of samples in $S_j(k)$.

Step 4. If $\mathbf{c}_j(k+1) = \mathbf{c}_j(k)$ for $j = 1, 2, \dots, m$, the algorithm has converged and the procedure is terminated. Otherwise go to Step 2.

The performance of the K-Means algorithm depends largely on the number of cluster centers specified, the choice of initial cluster centers, the order in which the training samples are presented, and the spatial properties of the training data [32]. The K-Means algorithm can be expected to yield reasonable results when the data points are linearly separable [33]. Figure 3.6 shows an example where the data is linearly separable.

Once the centers of radial-basis function are determined, the training of the RBF network is straightforward as described in the previous chapter. However, the problem associated with the K-Means method is that the number of the cluster centers m must be

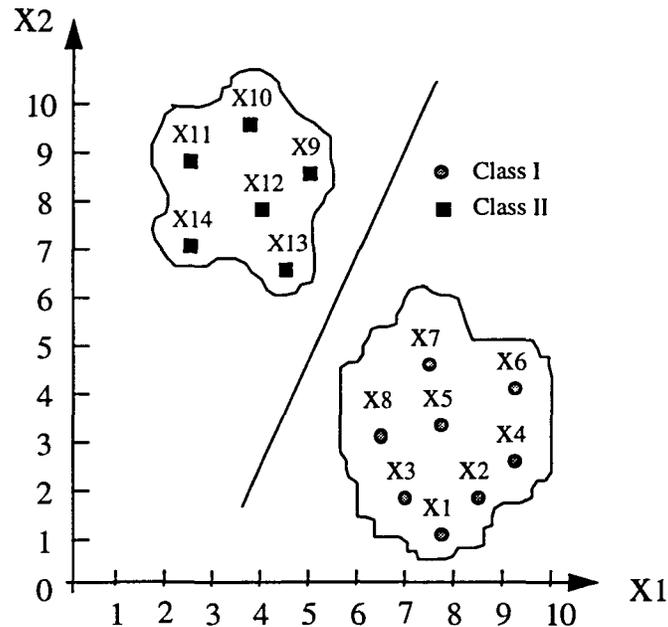


Figure 3.6 Linearly separable classes.

prespecified. Since they are not known *a priori*, various values of m as well as different choices of initial cluster centers have to be tried.

3.3.3 Optimal Adaptive K-Means Algorithm

Optimal adaptive K-Means clustering algorithms have been used as a mechanism for training neural networks, such as radial-basis function networks [34]. In order to determine the number of clusters automatically, we propose a new algorithm based on a modification of the conventional K-Means algorithm. Figure 3.7 shows a flow chart describing the steps involved in the new approach.

The basic idea of this algorithm is to automatically adjust the number of clusters based on the spatial distribution of the samples. The procedure starts by arbitrarily selecting samples as cluster centers and applying the K-Means algorithm to obtain the initial centers. Let d be the minimum inter-cluster distance among the clusters. Define the

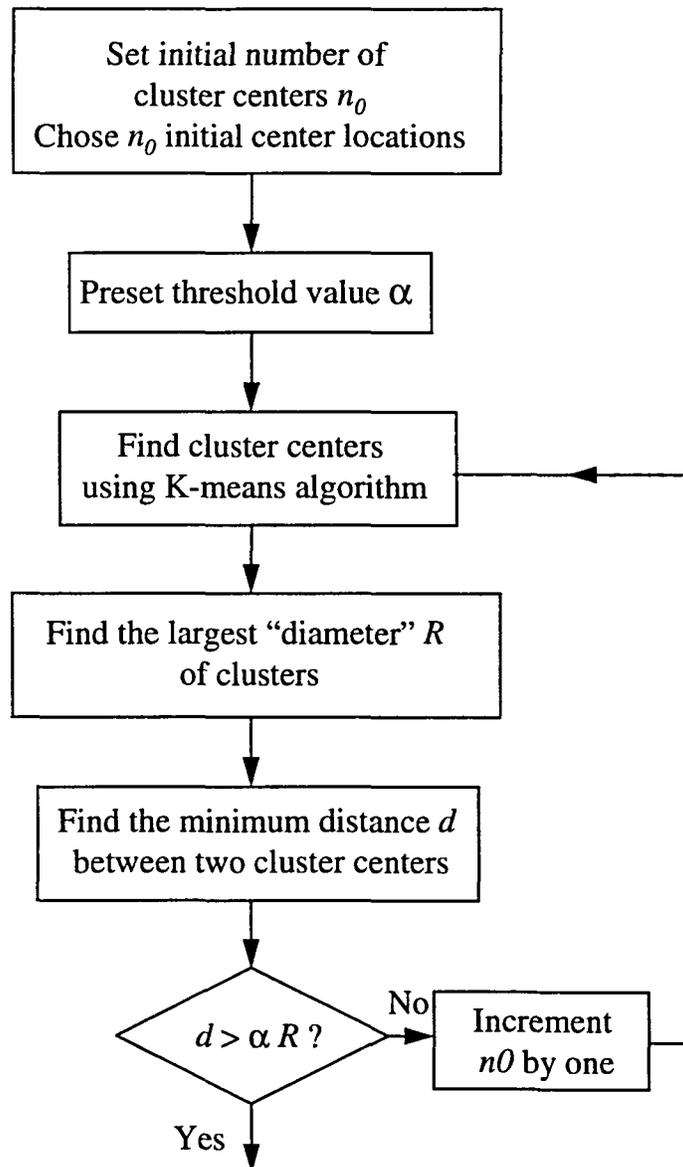


Figure 3.7 Flow chart for adaptive K-Means algorithm.

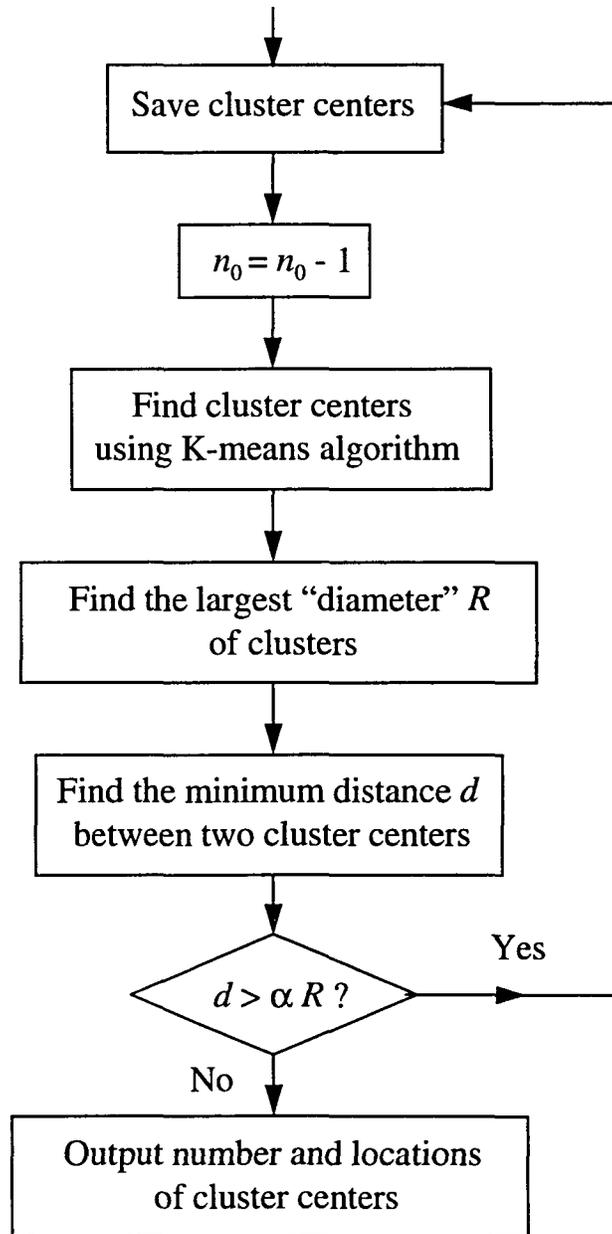


Figure 3.7 (continued)

“diameter” of a cluster as the maximum distance between two samples within the cluster. Let R be the largest diameter of clusters. If d is less than αR , where α is a preset threshold value, the number of the cluster centers is increased by one. Otherwise it is reduced by one. The K-Means algorithm is then applied to obtain the new cluster centers. The values of d and R are recalculated, and the recursive procedure continues until the number of the clusters does not change. For this project, the preset threshold, α , was chosen to be between 1.5 and 2.0.

CHAPTER 4. OPTIMAL TIME DOMAIN RBF NETWORKS

4.1 Steepest Descent and Optimal RBF Network

As discussed in Chapter 3, the RBF network parameters can be determined directly by calculating the pseudo-inverse after the cluster centers and the widths of the Gaussian radial basis functions are determined. This is the simplest approach for solving the problem of mapping MFL signals to defect profiles. Figure 4.1(a) shows a typical prediction of a defect shown in Figure 4.1(b). Figure 4.1(c) illustrates a one dimensional slice of both the predicted and true profiles along the axial direction. The MFL signals and defect profile data used were provided by Vetco Pipeline Services, which were generated under actual experimental conditions. The data used for testing were not part of the training data set. The maximum error percentages of depth, length, and width prediction are about 27%, 100%, and 100%, respectively. This result indicates that the difference between the true defect profile and the predicted profile is very large, and that the performance of the network is not satisfactory. In order to improve the performance of the RBF network, we need to optimize the network using appropriate techniques. This chapter presents several optimal approaches for determining the cluster centers (c), the widths (σ), and the synaptic weights (w).

Optimal approaches typical involve minimization of an appropriate cost function with respect to the parameters that one wishes to determine. A proper choice for the cost function is:

$$E = \frac{1}{2} \sum_i (y_{oi} - y_i)^T (y_{oi} - y_i)$$

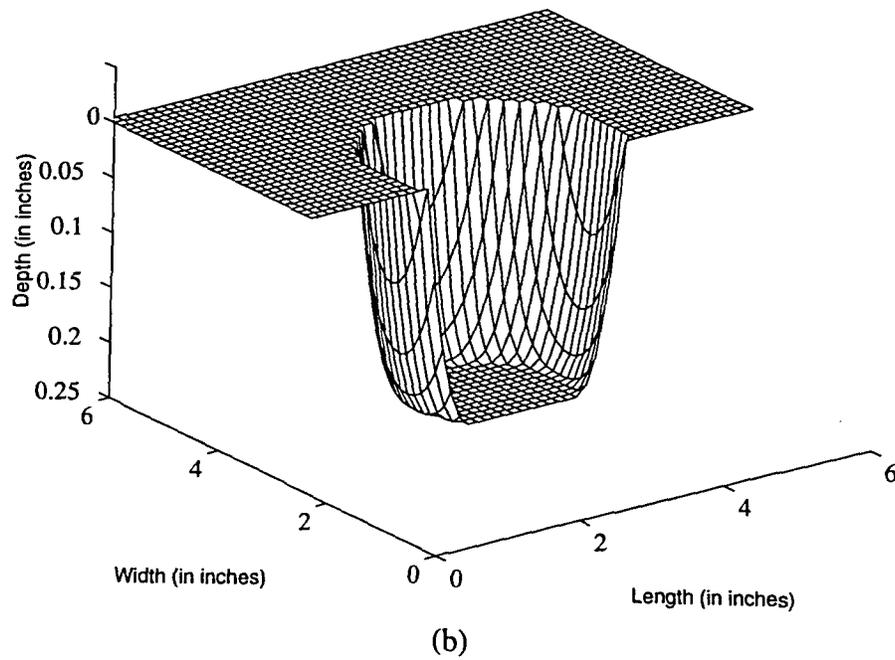
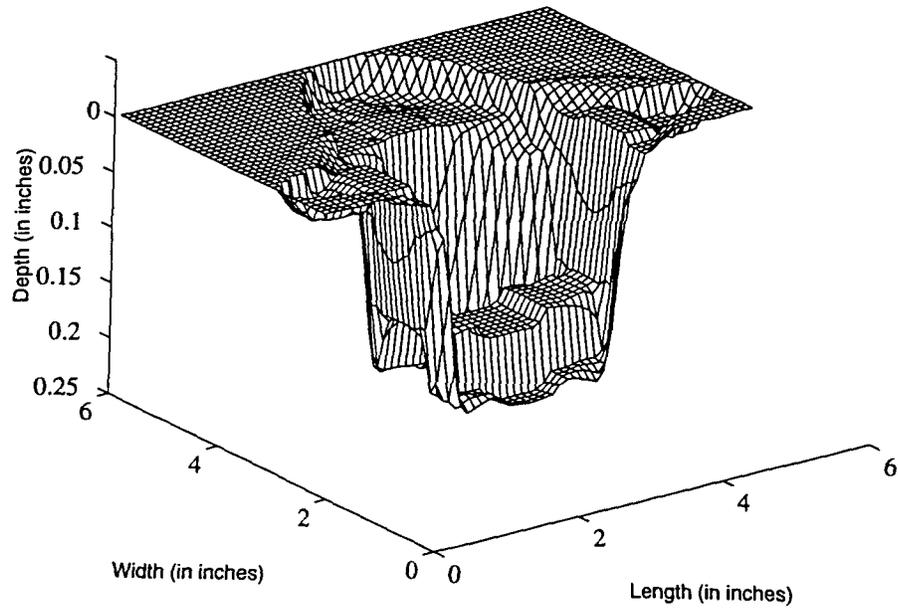


Figure 4.1 (a) Defect profile predicted using a non-optimal RBF network.
(b) Corresponding true defect profile.

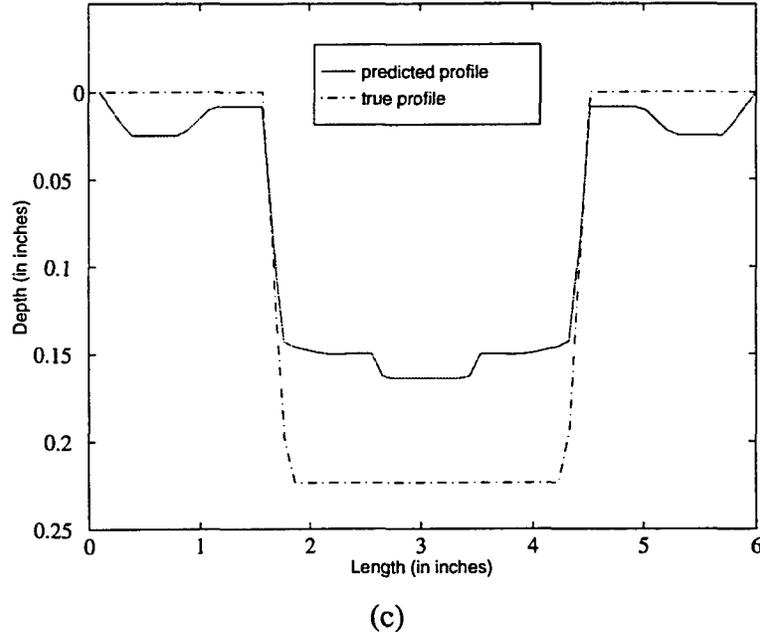


Figure 4.1 (continued) (c) 1-D section along axial direction.

$$= \frac{1}{2} \sum_i \left(\mathbf{y}_{0i} - \sum_j \mathbf{w}_j \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2\sigma_j^2}\right) \right)^T \left(\mathbf{y}_{0i} - \sum_j \mathbf{w}_j \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2\sigma_j^2}\right) \right) \quad (4.1)$$

where \mathbf{y}_{0i} is the i th true defect profile, \mathbf{y}_i is the i th predicted profile, and \mathbf{x}_i is the i th MFL signal.

The task can be characterized as one of solving a nonlinear, unconstrained optimization problem. Many algorithms can be used to minimize the cost function. Examples of such methods include the direct search methods and the gradient descent methods [35, 36]. In this research work, the steepest descent method is used rather than the direct search methods, since the direct search methods are usually less efficient.

The gradient of a function $y(\mathbf{x})$ with respect to the vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is defined as

$$\nabla y(\mathbf{x}) = \frac{dy(\mathbf{x})}{d\mathbf{x}} = \begin{pmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{pmatrix} \quad (4.2)$$

This equation tells us that the gradient of a scalar function $y(\mathbf{x})$ is a vector, and each component of the gradient gives the maximum rate of change of the function in the direction of that component. Consequently, the gradient points in the direction of the maximum rate of increase of the function $y(\mathbf{x})$, and the negative of the gradient points in the direction of maximum rate of decrease of the function $y(\mathbf{x})$. From this, it can be seen that an iterative scheme for minimizing the cost function can be devised. This constitutes the central idea behind the steepest descent algorithm. The steepest descent algorithm can be stated as follows [35, 36]:

Step 1. Start with an initial point \mathbf{x}_1 .

Step 2. Find the search direction $\mathbf{s}_1 = -\nabla y(\mathbf{x}_1)$.

Step 3. Find the step size Δx so as to minimize $y(\mathbf{x}_1 + \Delta x \cdot \mathbf{s}_1)$.

Step 4. Calculate the next point \mathbf{x}_2 using $\mathbf{x}_2 = \mathbf{x}_1 + \Delta x \cdot \mathbf{s}_1$

Step 5. Terminate the iteration if $\|y(\mathbf{x}_{i+1}) - y(\mathbf{x}_i)\| < \epsilon$, where ϵ is error tolerance. Otherwise return to step 2.

Applying this algorithm, the optimal parameters of the RBF network can be found. The update equation for \mathbf{c} and σ are given as follows:

$$\mathbf{c}_{i+1} = \mathbf{c}_i - \Delta \mathbf{c}_i \cdot \nabla_{\mathbf{c}} E(\mathbf{c}, \sigma) \quad (4.3)$$

$$\sigma_{i+1} = \sigma_i - \Delta \sigma_i \cdot \nabla_{\sigma} E(\mathbf{c}, \sigma) \quad (4.4)$$

where Δc_i and $\Delta \sigma_i$ are the step size of c and σ in i th iteration, and $E(c, \sigma)$ is the cost function defined in equation (4.1).

From the above two equations, the synaptic weights, w , can be updated using

$$\mathbf{w}_{i+1} = [\Phi(\mathbf{c}_{i+1}, \boldsymbol{\sigma}_{i+1})]^+ \cdot \mathbf{y} \quad (4.5)$$

where symbol “+” indicates pseudo-inverse of matrix $\Phi(\mathbf{c}_{i+1}, \boldsymbol{\sigma}_{i+1})$.

Figure 4.2 illustrates the flow chart for optimizing the RBF network parameters using the steepest descent algorithm.

So far, the problem of minimizing the cost function with respect to c and σ has been discussed. However, there is an additional issue of choosing the optimal step sizes, Δc and $\Delta \sigma$, during the iterative process.. The step sizes for updating c and σ can be optimized using the same technique as above. Figure 4.3 demonstrates the flow chart for choosing the optimal step sizes Δc and $\Delta \sigma$.

4.2 Defect Characterization in the Time Domain

There are several approaches for mapping MFL signals to defect profiles. One of the methods is to map MFL signals directly to defect profiles using an optimal neural network. The initial parameters of the optimal RBF network can be obtained using the K-Means method. The RBF network can be optimized using the steepest descent technique presented earlier.

From equation (4.1), the gradient of the cost function E with respect to c and σ are given as follows.

$$\nabla_{\mathbf{c}_k} E = -\sum_{i=1}^n \left\{ \left[\mathbf{y}_{0i} - \sum_{j=1}^m \mathbf{w}_j \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2\sigma_j^2}\right) \right]^T \cdot \mathbf{w}_k \cdot \frac{(\mathbf{x}_i - \mathbf{c}_k)}{2\sigma_k^2} \right\} \quad (4.6)$$

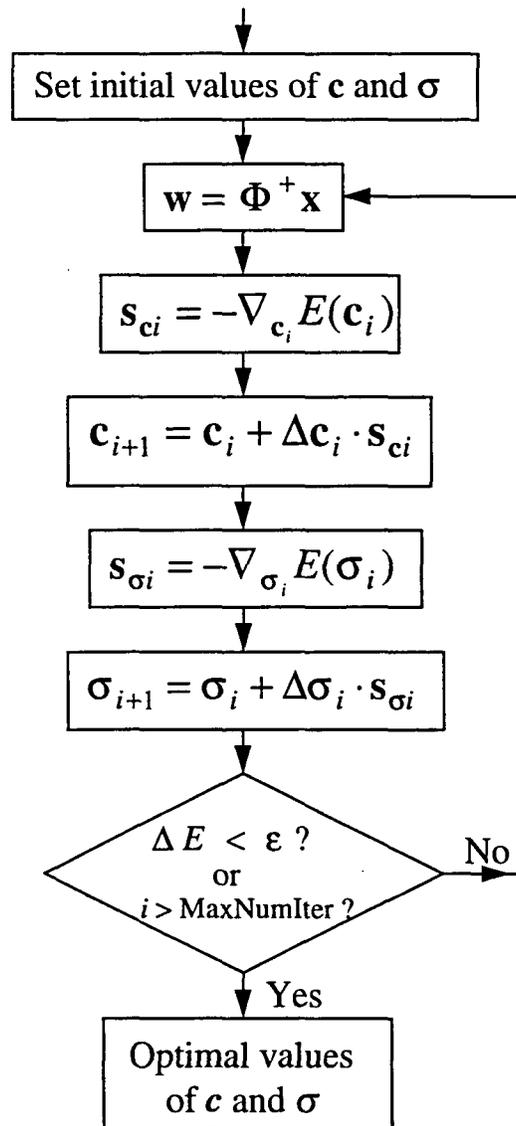


Figure 4.2 Flow chart for optimizing the RBF network using the steepest descent method.

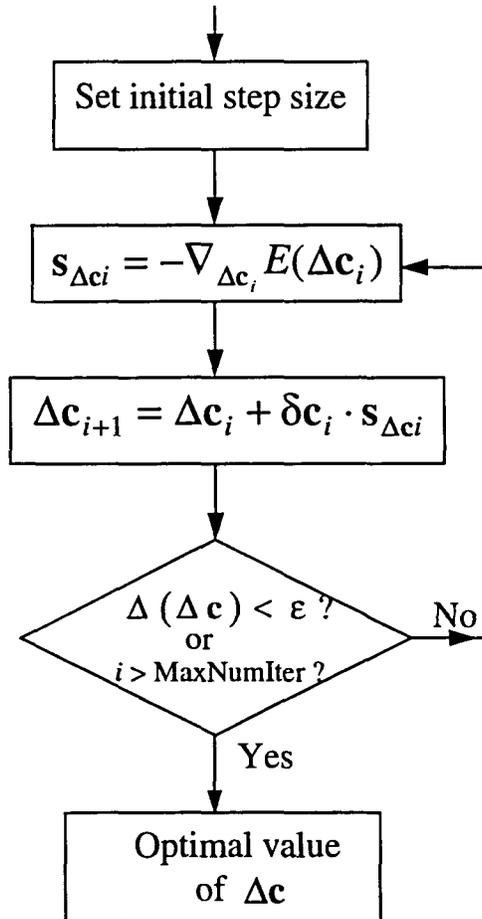


Figure 4.3 Flow chart for optimizing the step size Δc .

$$\nabla_{\sigma_k} E = -\sum_{i=1}^n \left\{ \left[\mathbf{y}_{0i} - \sum_{j=1}^m \mathbf{w}_j \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2\sigma_j^2}\right) \right]^T \cdot \mathbf{w}_k \cdot \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2\sigma_j^2}\right) \cdot \frac{\|\mathbf{x}_i - \mathbf{c}_k\|^2}{\sigma_k^3} \right\} \quad (4.7)$$

$$\nabla_{\Delta \mathbf{c}_k} E = -\sum_{i=1}^n \left\{ \left[\mathbf{y}_{0i} - \sum_{j=1}^m \mathbf{w}_j \phi_{ij}(\Delta \mathbf{c}_k) \right]^T \cdot \mathbf{w}_k \cdot \phi_{ij}(\Delta \mathbf{c}_k) \cdot \frac{(\mathbf{x}_i - \mathbf{c}_j + \Delta \mathbf{c}_k \cdot \nabla_{\mathbf{c}_k} E)^T \cdot \nabla_{\mathbf{c}_k} E}{2\sigma_k^2} \right\} \quad (4.8)$$

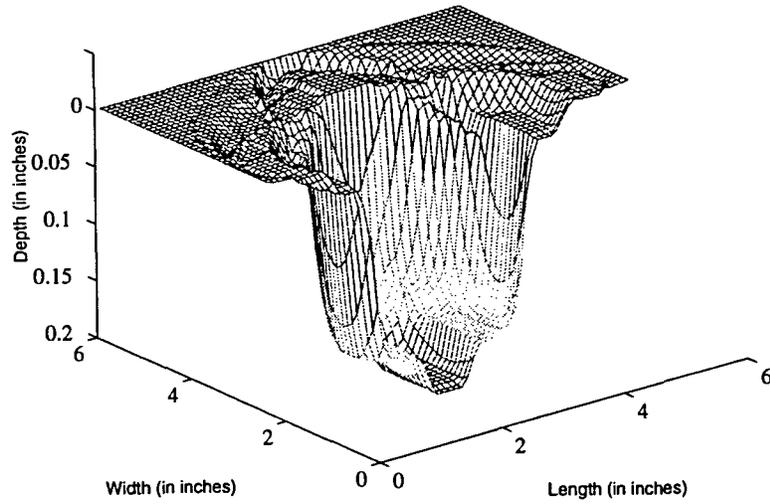
$$\nabla_{\Delta \sigma_k} E = \sum_{i=1}^n \left\{ \left[\mathbf{y}_{0i} - \sum_{j=1}^m \mathbf{w}_j \phi_{ij}(\Delta \sigma_k) \right]^T \cdot \mathbf{w}_k \cdot \phi_{ij}(\Delta \sigma_k) \cdot \frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2 \cdot \nabla_{\sigma_k} E}{2(\sigma_k - \Delta \sigma_k \cdot \nabla_{\sigma_k} E)^3} \right\} \quad (4.9)$$

where

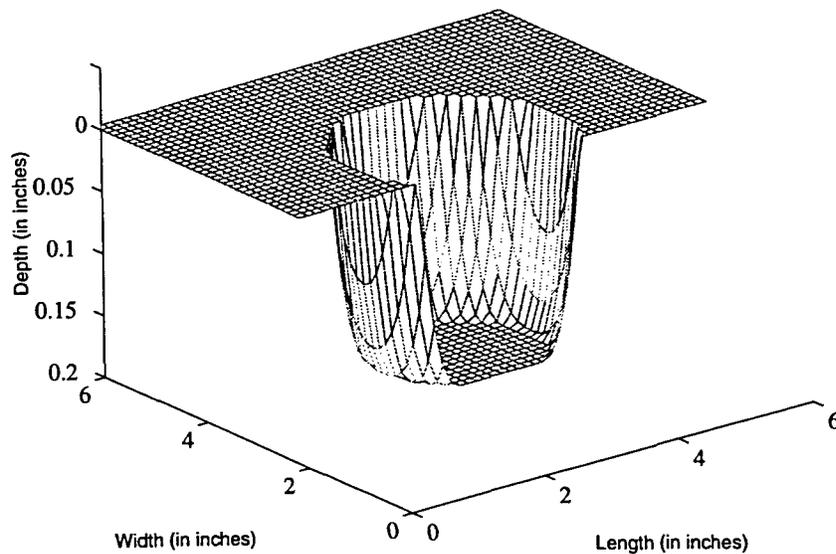
$$\phi_{ij}(\Delta \mathbf{c}_k) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{c}_j + \nabla_{\mathbf{c}_k} E \cdot \Delta \mathbf{c}_k\|^2}{2\sigma_j^2}\right) \quad (4.10)$$

$$\phi_{ij}(\Delta \sigma_k) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2(\sigma_j - \nabla_{\sigma_k} E \cdot \Delta \sigma_k)^3}\right) \quad (4.11)$$

The RBF network can be trained optimally employing the steepest descent algorithm discussed in the previous section. Typical predictions of defect profiles are shown in Figure 4.4(a). The corresponding true profile and 1-D scan along axial direction are shown in Figure 4.4(b) and 4.4(c), respectively. The error percentages of depth, length, and width prediction are about 47%, 32% and 100%, respectively. Figure 4.5 shows another defect profile prediction together with the true profile and a 1-D scan along the axial direction. The results show that the locations and widths of the radial basis function have significant influence on the performance of the network. The results obtained using this approach are still unacceptable, since the difference between the true and predicted defect profile is quite large. Approaches for minimizing the error are described in the following section.

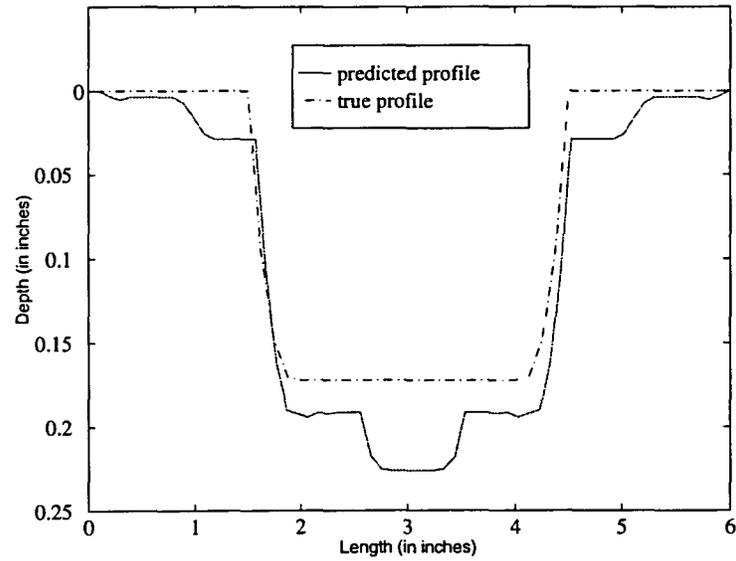


(a)



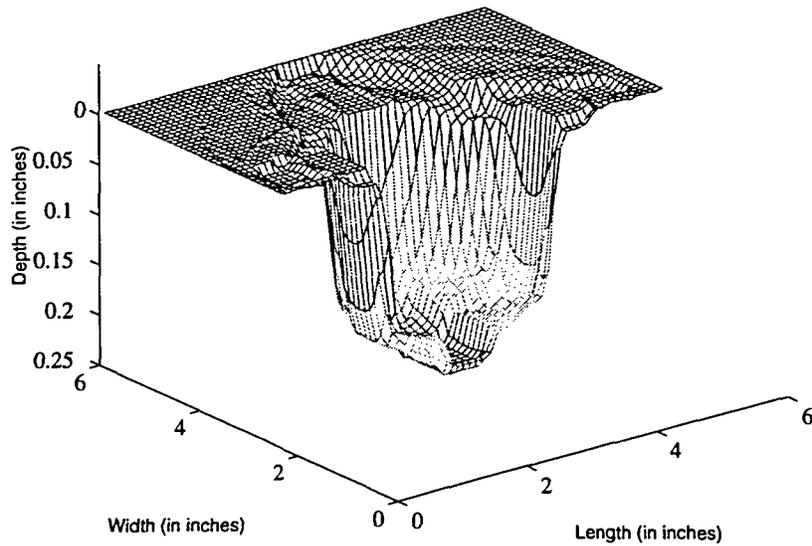
(b)

Figure 4.4 (a) Defect profile predicted using the steepest descent algorithm in the time domain. (b) Corresponding true defect profile.



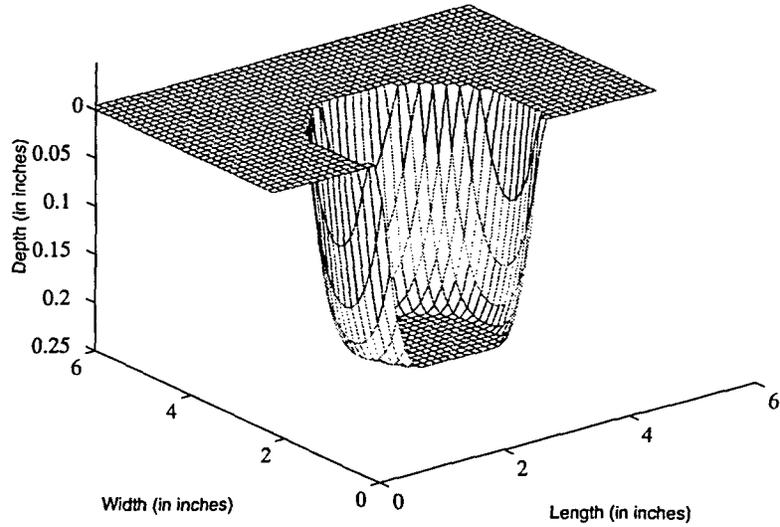
(c)

Figure 4.4 (continued) (c) 1-D scan of both predicted and true profiles along the axial direction.

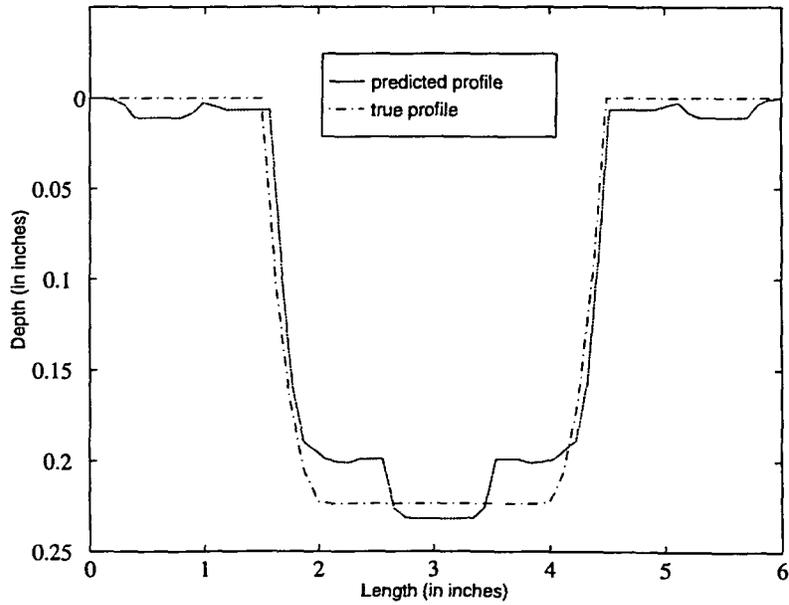


(a)

Figure 4.5 (a) Defect profile predicted using the steepest descent algorithm in the time domain.



(b)



(c)

Figure 4.5 (continued) (b) Corresponding true defect profile. (c) One dimensional scan of both predicted and true profiles along the axial direction.

4.3 Defect Characterization Using Defect Parameters

From the discussion in the previous section, it is apparent that an RBF network whose parameters are derived using the steepest descent algorithm does not offer a desirable level of performance. The defect profile predictions shown in Figure 4.4(a) and Figure 4.5(a) appear to deviate from the true profiles considerably. The principle factor contributing to error is the relatively large number of weights that need to be determined using only a limited amount of training data.

As mentioned in section 2.3, the defects used in this research work have certain regular shapes. They can be defined using five geometric parameters: length, width, depth, surface angle, and roundness. Consequently, the output nodes can be reduced if these five parameters are used as the output of the optimal RBF network. Figure 4.6 illustrates a block diagram for this defect characterization procedure.

A typical defect profile predicted using this technique is shown in Figure 4.7(a). The corresponding true profile and a 1-D slice along the axial direction are shown in Figure 4.1(b) and Figure 4.7(b). The percentage errors in depth, length, and width prediction are about 3%, 14%, and 10%, respectively. Figure 4.8 shows another defect profile predicted using this technique, together with the true profile, and a 1-D section along the axial direction. The corresponding error percentages for depth, length, and width prediction are about 0.2%, 10%, and 3%, respectively. It can be seen from these results that the improvement obtained using this network is significant. However, the approach has limited applicability since defects occurring naturally seldom have regular shape and therefore cannot be characterized by five geometric parameters. It is difficult to obtain any *a priori* knowledge about defect profiles, and so from a practical view point, this approach is not useful even though it offers superior performance for characterizing defects with regular profiles. However, the results indicate that by decreasing the number of nodes in the RBF network, one could improve the performance of the network.

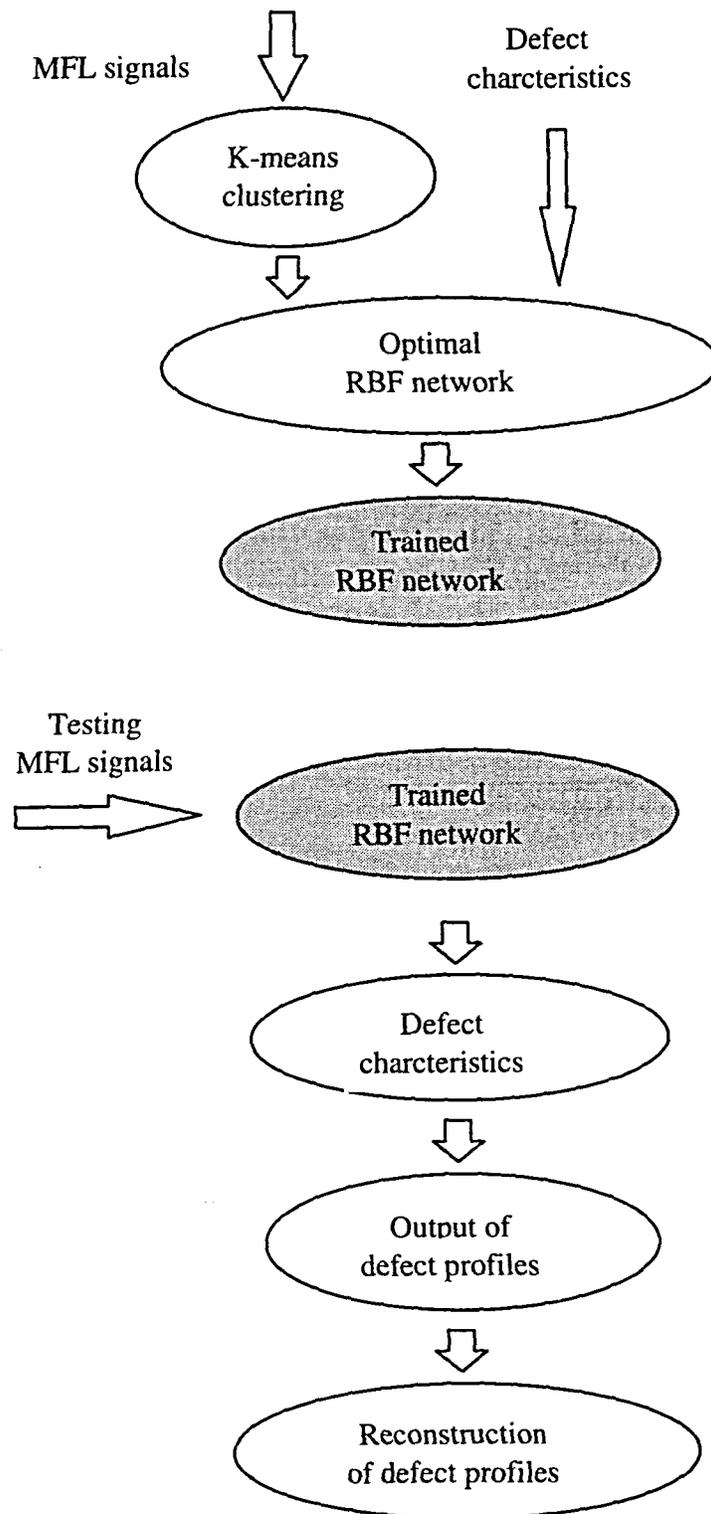


Figure 4.6 A parametric procedure for defect characterization.

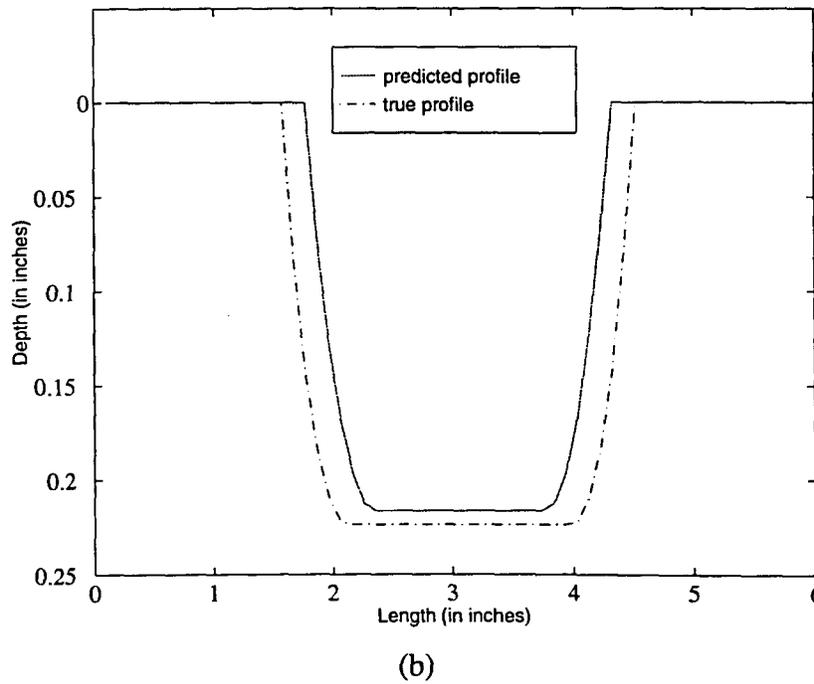
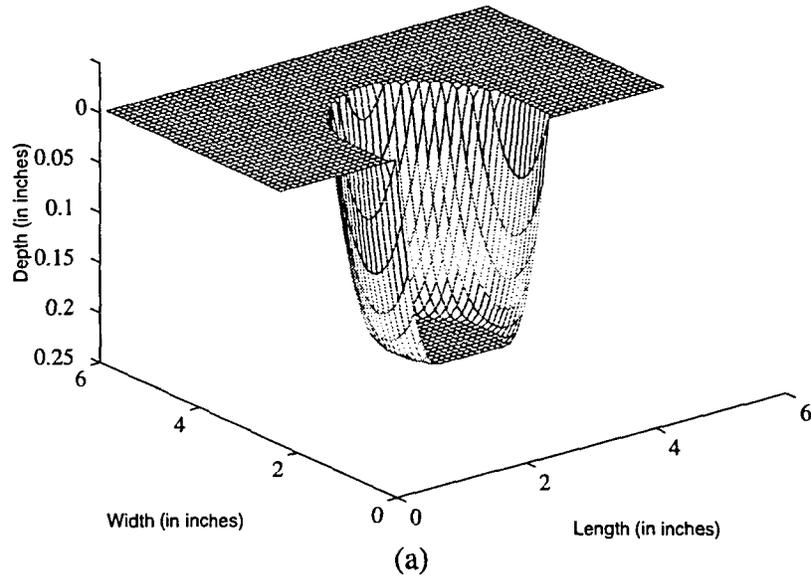
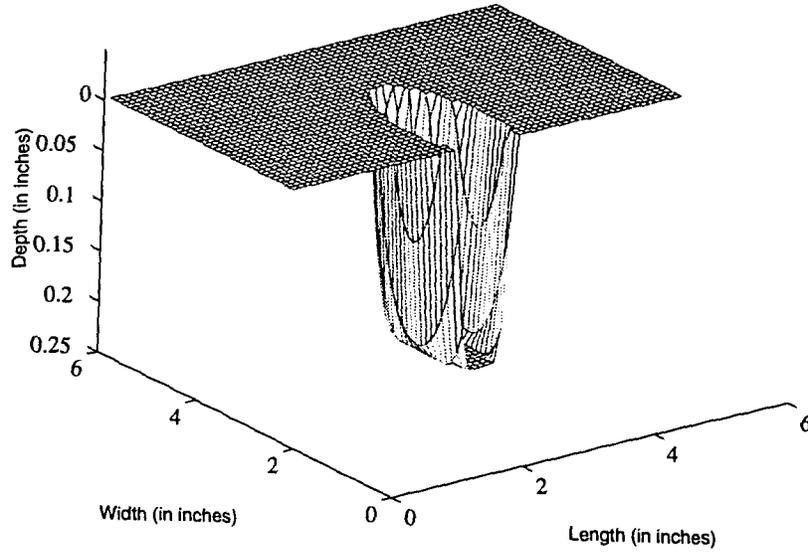
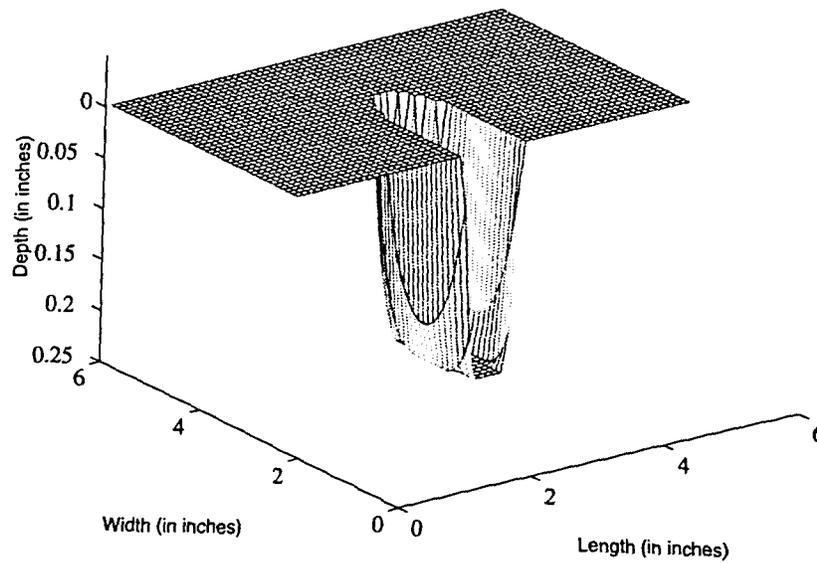


Figure 4.7(a) Defect profile predicted using a limited number of parameters. (b) 1-D section of both predicted and true profile along the axial direction.

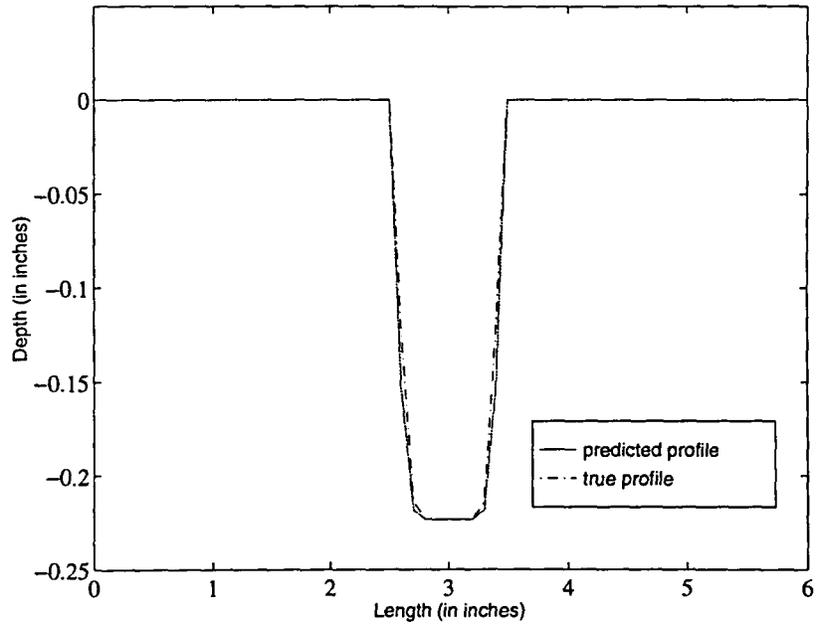


(a)



(b)

Figure 4.8 (a) Defect profile synthesized from parameters predicted by the RBF network. (b) Corresponding true profile.



(c)

Figure 4.8 (continued) (c) One dimensional slice along the axial direction.

CHAPTER 5. OPTIMAL FREQUENCY DOMAIN RBF NETWORKS

Results from using the approach involving the prediction of the defect parameters instead of the full profile indicated that a reduction in the number of output nodes (and hence the degrees of freedom) can reduce “noise” in the predicted profile. In this chapter, we will extend this concept further by using a method that seeks to minimize the number of input nodes as well. Specifically, the method proposed in this chapter attempts to map the signal spectrum to the defect spectrum.

5.1 Spectra of MFL Signals and Defect Profiles

So far, the strategy involved mapping the MFL signal directly to the defect profile. An alternate approach is to map the spectral information. This exploits the fact that the low frequency components of MFL signals contain most of the information. Similarly, the spectrum of the defect profile has most of its energy confined to the low frequencies.

It is therefore possible to solve the inverse problem in the frequency domain, where we map the spectrum of an MFL signal to the spectrum of a defect profile. Figure 5.1 and Figure 5.2 show typical spectra of an MFL signal and a defect profile, respectively. Both spectra indicate that most of the energy of the signals is concentrated at the low frequency end. Most high frequency components have limited energy. Therefore, the low frequency parts of the spectra are sufficient to characterize an MFL signal or a defect profile.

Figure 5.3(a) illustrates a reconstructed MFL signal, in which only 13×25 points out of a total of 42×58 points in the MFL spectrum are used. Similarly Figure 5.4 shows a reconstructed defect profile, in which only 20×20 out of a total of 50×50

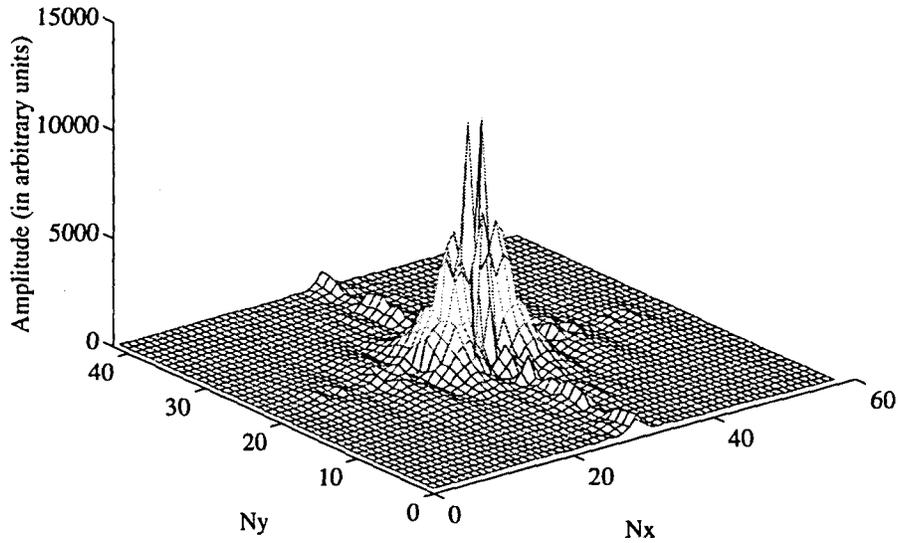


Figure 5.1 The spectrum of an MFL signal.

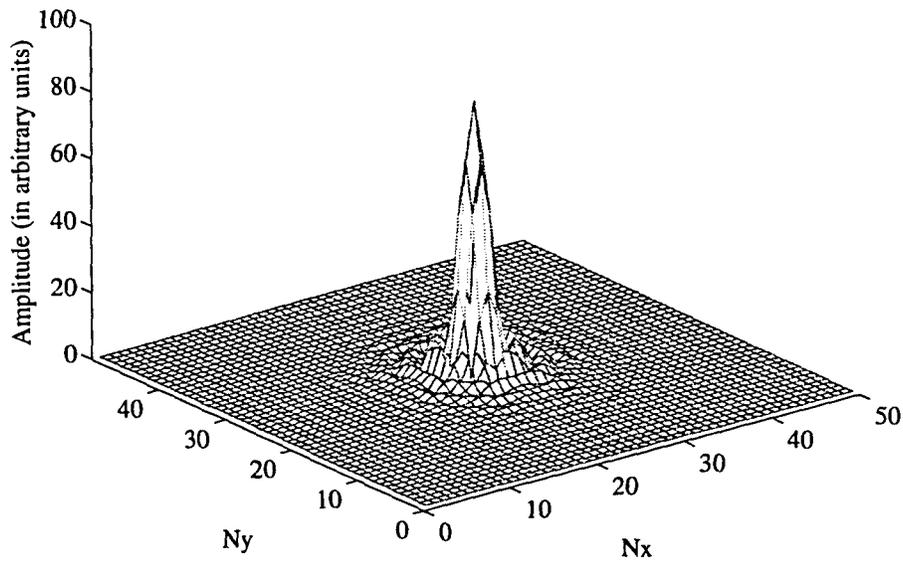


Figure 5.2 The spectrum of a defect profile.

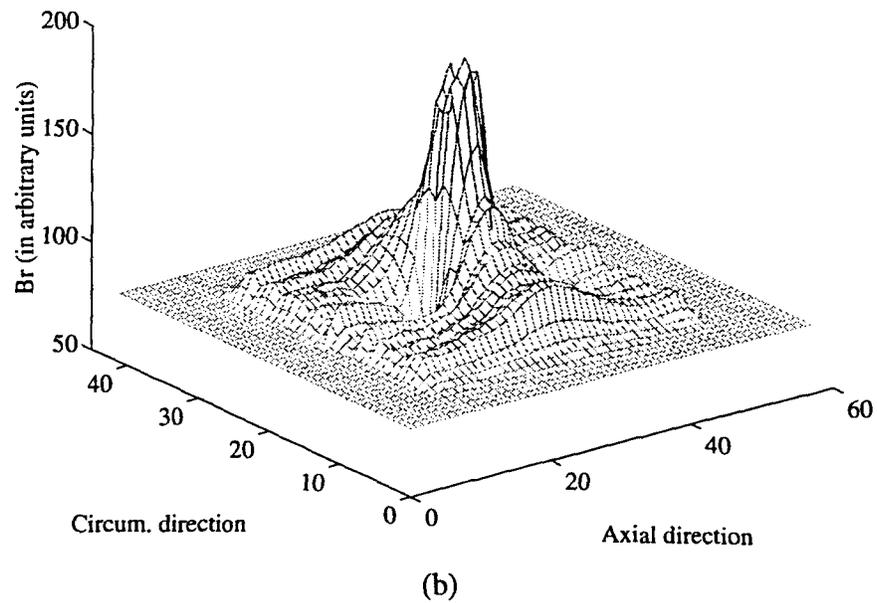
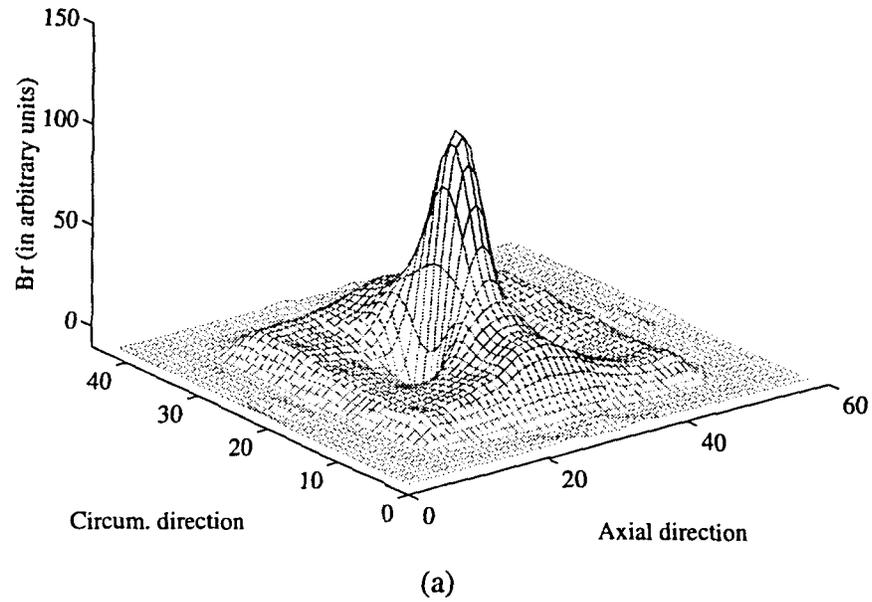


Figure 5.3 (a) MFL signal reconstructed using low frequency components of the signal. (b) The original MFL signal.

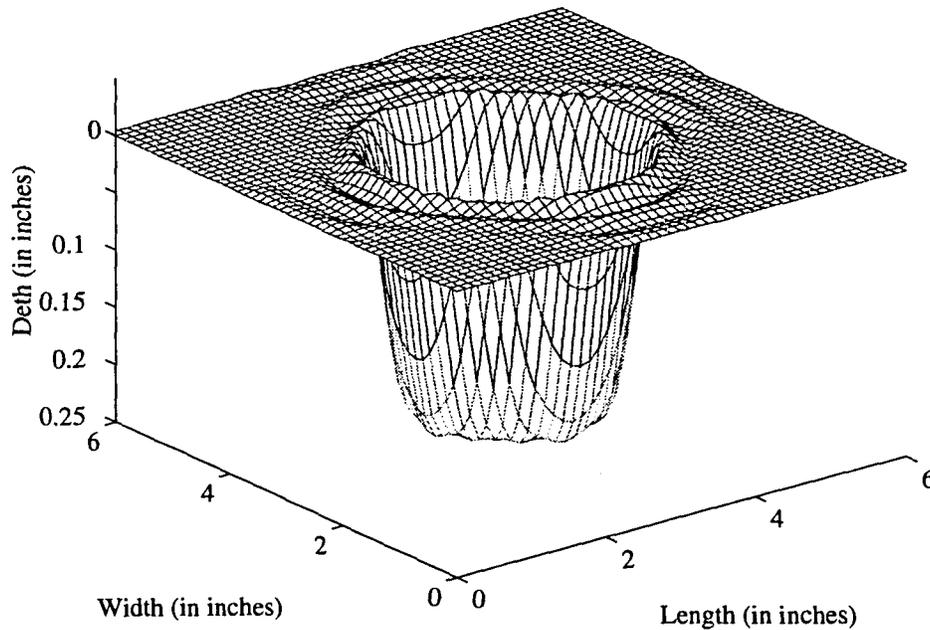


Figure 5.4 Defect profile reconstructed using the low frequency components.

point spectrum of a defect profile are used. The original MFL signal is shown in Figure 5.3(b), while the original defect profile is shown in Figure 4.1(b).

The spectra of MFL signals and defect profiles contain not only the magnitude information, but also phase information. In order to make use of all the information, some modifications to the optimal RBF network are required. The simplest way to accomplish this is to use the complex spectra. After this is done, the RBF network can be trained and optimized properly using the steepest descent algorithm in the frequency domain. Figure 5.5 illustrates the performance of an RBF network trained in this manner. The corresponding true defect profile is shown in Figure 4.4(b). Figure 5.5 demonstrates that the side lobes of the predicted profile have decreased significantly, and the prediction of the defect length is better than those obtained with previous methods. However, the general performance is not very satisfactory. There is some difference between the predicted depth and true depth of the defect profile. The maximum error percentage of

depth prediction is more than 20%. This may be due to the limitation of the steepest descent method whose solution sometimes trapped at a local minimum, rather than a global minimum, since the cost function is not quadratic. Consequently, a better optimization algorithm has to be employed to improve the RBF network performance.

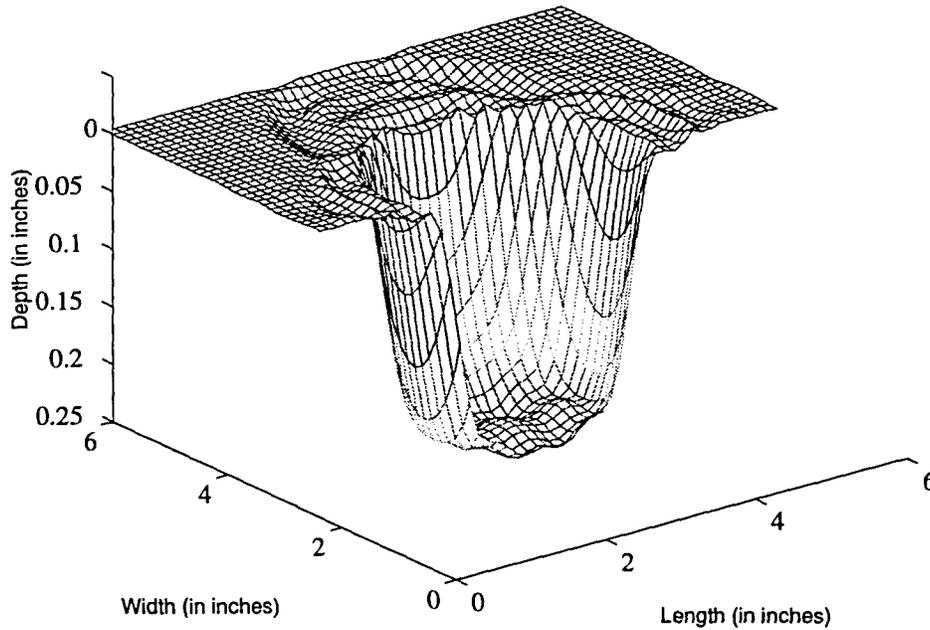


Figure 5.5 Defect profile predicted using steepest descent method in frequency domain.

One of the candidates is the conjugate gradient algorithm [24]. However, the behavior of this method is dependent on the cost function, and the algorithm is sensitive to the choice of the initial starting point. Even if a good initial starting point is chosen, it is not guaranteed that a global minimum will be reached. Another choice is the simulated annealing algorithm. The following section describes the use of simulated annealing algorithm to arrive at the solution.

5.2 Simulated Annealing Algorithm

The simulated annealing algorithm has attracted significant attention in recent years as a tool for solving global optimization problems, particularly in cases where the desired global minimum is hidden among many local minima. It was first introduced by Kirkpatrick in the early 1980's [37]. The algorithm is based on a strong analogy between the physical annealing process of condensed materials and the problem of solving global optimization problems. In this section, the theory underlying the simulated annealing algorithm is presented.

5.2.1 The Metropolis Algorithm

In condensed matter physics, annealing is known as a thermal process for obtaining the lowest energy state (called ground state) of a condensed matter system by using a heat bath . The process consists of the following steps [38]:

Step 1. Increase the temperature of the heat bath to a value at which the solid state changes to liquid state.

Step 2. Decrease the temperature of the heat bath very slowly so that at each temperature the system is in the equilibrium state. Continue the process until the system reaches the ground state.

The physical annealing process can be modeled successfully by using the Monte Carlo technique developed by Metropolis in 1953 [39]. The Metropolis algorithm can be stated as follows: Given a current state i of a system with energy E_i , then the next state of the system is generated by applying a small distortion, for instance by changing the spatial configuration of the system. The energy of the next state is denoted as E_j . If E_j is smaller than E_i , then this new state is accepted. If E_j is larger than E_i , this state j is accepted with the following probability:

$$P_{i \rightarrow j} = \exp\left(-\frac{E_j - E_i}{k_B T}\right) \quad (5.1)$$

where k_B is the Boltzmann constant, and T is the system temperature. Equation (5.1) is actually the well known Boltzmann distribution for classical physical systems.

If the lowering of the system temperature is a sufficiently slow process, then the system can reach thermal equilibrium at each temperature. In the Metropolis algorithm, this can be achieved using a great number of transition states at each given temperature.

5.2.2 The Simulated Annealing Algorithm

The simulated annealing algorithm is based on the above theory and can be applied for solving global optimization problems. In order to develop the simulated annealing algorithm, an analogy between a many-body system in condensed matter physics and a combinatorial optimization problem can be drawn based on: (i) The solution of a global optimization problem is equivalent to a state of a many-body system in condensed matter physics, (ii) The cost function is equivalent to the energy of that state. Also, we introduce a parameter T that functions as temperature in thermal physics. The simulated annealing algorithm can be viewed as a variant of the Metropolis algorithm, evaluated at decreasing “temperatures”. The pseudo-code for the simulated annealing algorithm can be stated as shown in Figure 5.6 [40].

In Figure 5.6 $P(i \rightarrow j)$ refers to the acceptance probability, and is defined as:

$$P(i \rightarrow j) = \begin{cases} 1 & \text{if } E_j \leq E_i \\ \exp\left(-\frac{E_j - E_i}{T}\right) & \text{if } E_j > E_i \end{cases} \quad (5.2)$$

N_k is the number of iterations to reach the equilibrium state at “temperature” T_k .

A typical feature of the simulated annealing algorithm is that, besides accepting improvements in cost, it also, to a limited extent, accepts deterioration in cost. Initially, at

```

SimulatedAnnealing( );
{
  Initialize( $i_{start}, T_0, N_0$ );
   $k = 0$ ;
   $i = i_{start}$ ;
  do{
    for( $l = 0; l < N_k; l ++$ ) {
      GenerateState( $j$  from  $i$ );
      if ( $E_j < E_i$ )
         $i = j$ ;
      elseif  $P(i \rightarrow j) > \text{random}[0,1]$ 
         $i = j$ ;
    }
     $k ++$ ;
    AdajustIterNum( $N_k$ );
    AdajustTemp( $T_k$ );
  } while(StopCriterion)
}

```

Figure 5.6 The pseudo-code for the simulated annealing algorithm [40].

high “temperature”, large levels of deterioration may be accepted. As the “temperature” decreases, only a smaller level of deterioration will be accepted. Therefore, the simulated annealing algorithm, in contrast to the steepest descent algorithm, can allow the solution to escape from the local minima without adding greatly to the complexity of the method.

The probability of acceptance is implemented by comparing the transition probability $P(i \rightarrow j)$ with a random number uniformly distributed in the interval $[0, 1]$.

A concrete demonstration of validity of this approach was presented by Arts and Korst who discussed the traveling salesman problem [40]. However the potentially burdensome amount of time required to converge to a near-optimal solution, limits its application. A faster algorithm has to be employed in order for the simulated annealing algorithm to be more practical.

5.3 Optimal RBF Network Using Simulated Annealing

In order to improve the performance of Gaussian RBF network further, the simulated annealing algorithm is used. The flow chart of this technique is shown in Figure 5.7. The principle difference between the simulated annealing algorithm and the steepest descent algorithm lies in the procedure for updating the c and σ , and hence the network weights w . In steepest descent, c and σ are updated using equations (4.5) and (4.6), which causes the cost function E to always decrease. Therefore, the solution has a very good chance to be trapped at a local minimum, while in simulated annealing, c and σ are updated using the following equations:

$$\mathbf{c}_{i+1} = \mathbf{c}_i + \mathbf{rand}(\Delta\mathbf{c}_i) \quad (5.3)$$

$$\sigma_{i+1} = \sigma_i + \mathbf{rand}(\Delta\sigma_i) \quad (5.4)$$

where $\mathbf{rand}(\Delta\mathbf{c}_i)$ and $\mathbf{rand}(\Delta\sigma_i)$ are random vectors with the same dimensions as c and σ . Consequently, the value of the cost function will increase at some steps. This makes “climbing over a hill” possible if the deterioration of the cost function is accepted with a certain probability. Eventually the solution will move towards the near-global minimum.

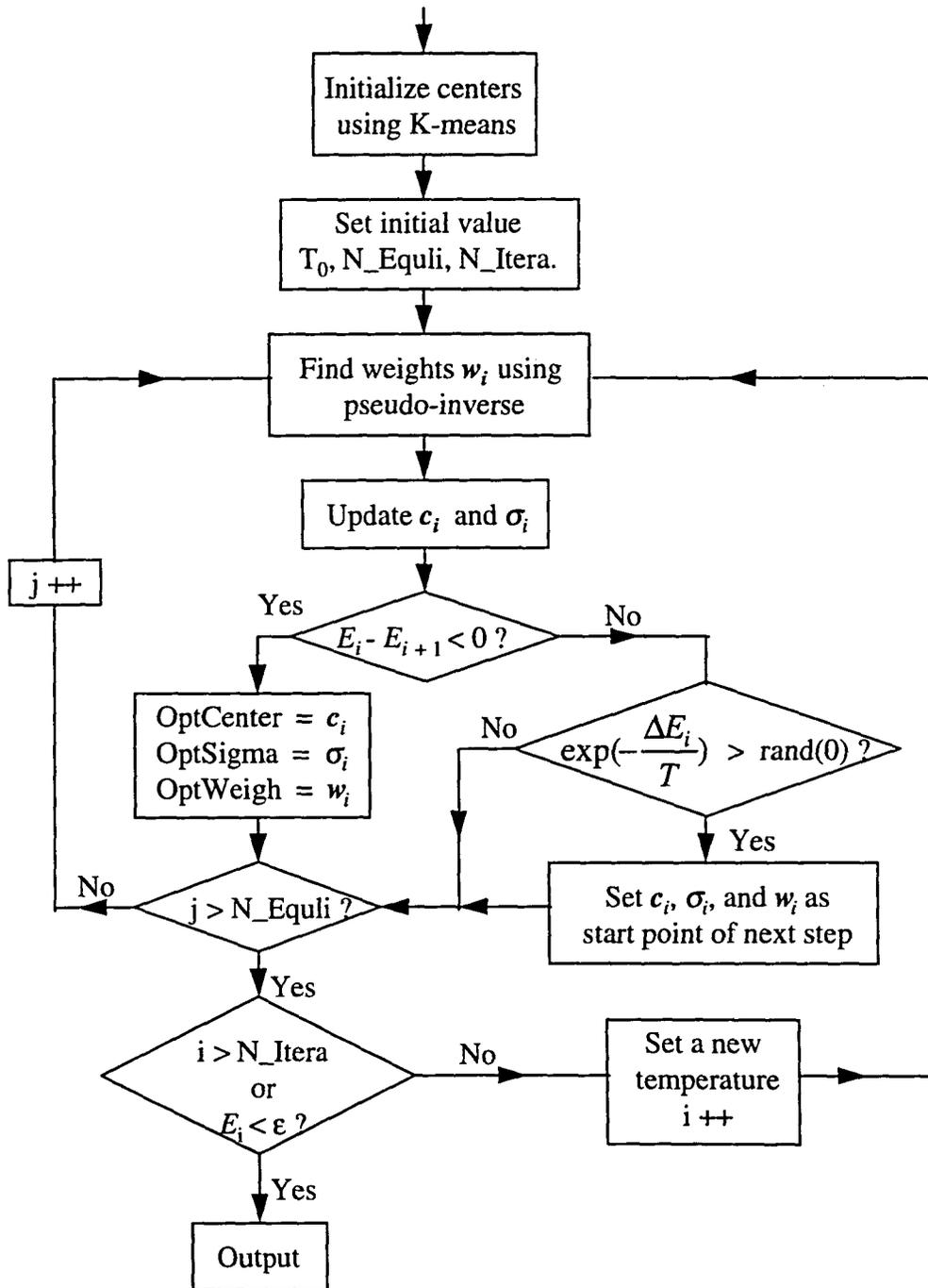


Figure 5.7 Flow chart of the simulated annealing algorithm for training an RBF network.

However, as mentioned earlier, the pure random walk updating method (equations (5.3) and (5.4)) is not practical due to the large computational effort. The updating algorithm must be modified to expedite the whole algorithm. Several methods can be used to minimize the computation time, such as a choosing bigger “rate of temperature change”, or a “low initial temperature”. The major challenge lies in choosing an update algorithm for c and σ . The steepest descent and simulated annealing require contrasting levels of convergence rates. The former converges too fast, while the other is slow to converge. A good compromise is to employ a method that offers a near-global minimum with shorter computational time. In order to accomplish this, the updating algorithm for c and σ in the simulated algorithm can be modified to

$$c_{i+1} = c_i + \text{sign}[\text{rand}(0) - 0.5] \cdot \nabla_c E \cdot \Delta c_i \quad (5.5)$$

$$\sigma_{i+1} = \sigma_i + \text{sign}[\text{rand}(0) - 0.5] \cdot \nabla_\sigma E \cdot \Delta \sigma_i \quad (5.6)$$

where $\text{sign}(\)$ is defined as

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases} \quad (5.7)$$

and $\text{rand}(0)$ is a uniformly distributed random number in the interval [0, 1].

The first step in this method is to find the gradient of the cost function with respect to c and σ , just as it is done in the steepest descent approach. The second step is to choose *both* the down hill direction *and* the up hill direction as the search directions. Therefore, eventually, it is possible to move out of a local minimum trap. In order to take advantage of the simulate annealing algorithm, both search directions and the updating step sizes are randomized.

Figure 5.8(a) shows one of the defect profiles predicted using this algorithm. The improvement in the RBF network performance compared to the previous methods is

remarkable. Figure 5.8(b) shows a 1-D section of the resulting profile. The corresponding true profile is shown in Figure 4.4(b). The error percentages of depth, length, and width prediction are about 16%, 13%, and 15%, respectively. The input data set contained 48 sample signals. 46 out of the 48 samples were used as training data, and the remaining two were used as testing data. Using the same 48 sample signal set, another 46 signal sample set was used as training data, and the remaining two were used as test data. Continuing this way, a total of 6 defect profiles was predicted. The error percentage of the predicted depths turned out to be about 15%, except for one profile while had an error of about 45%. Figure 5.9 shows one of these predicted defect profiles, the corresponding true profile and 1-D section along the axial direction.

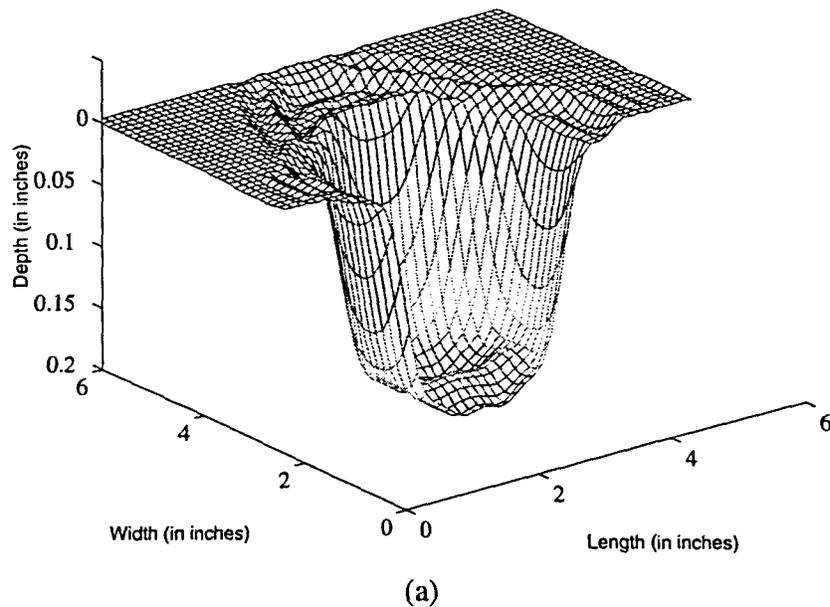
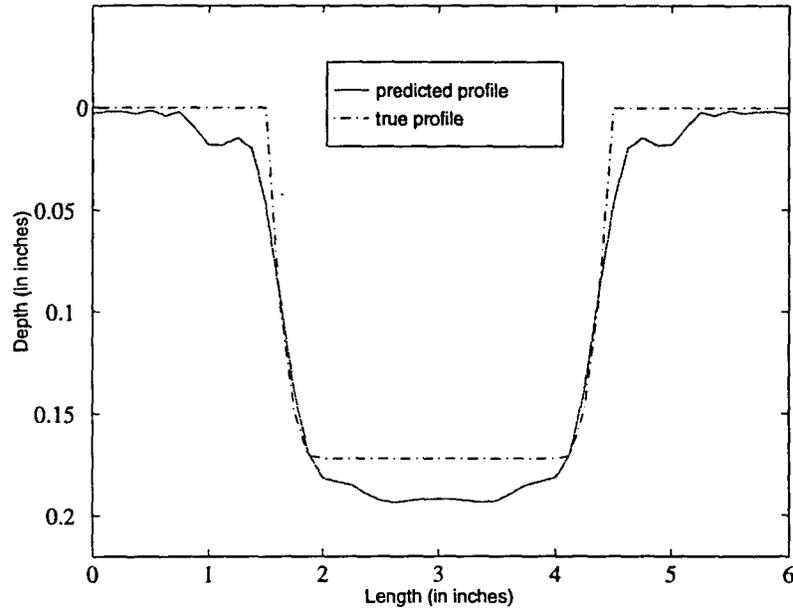
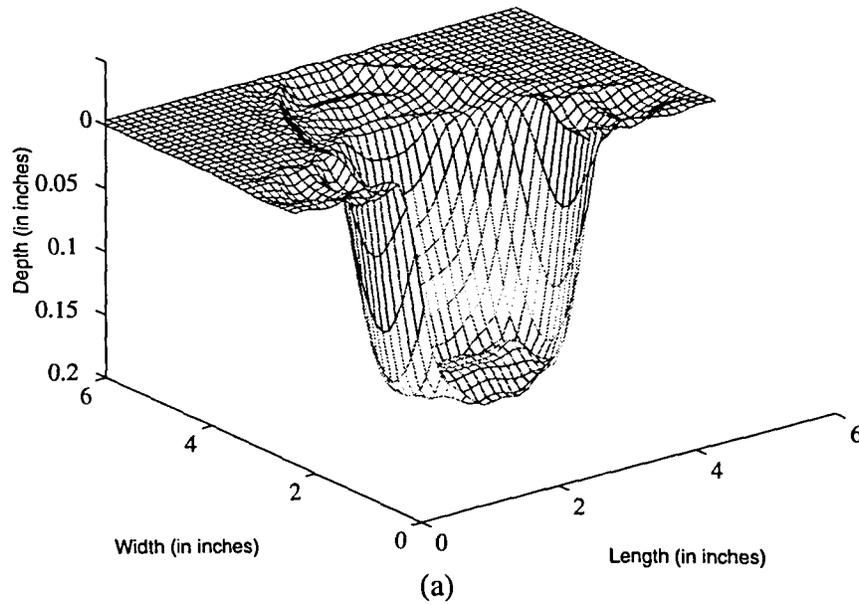


Figure 5.8 (a) Defect profile predicted using the simulated annealing algorithm.



5.8 (b)

Figure 5.8 (continued) (b) 1-D section of both the predicted and true profile along the axial direction.



(a)

Figure 5.9 (a) Defect profile predicted using the simulated annealing algorithm.

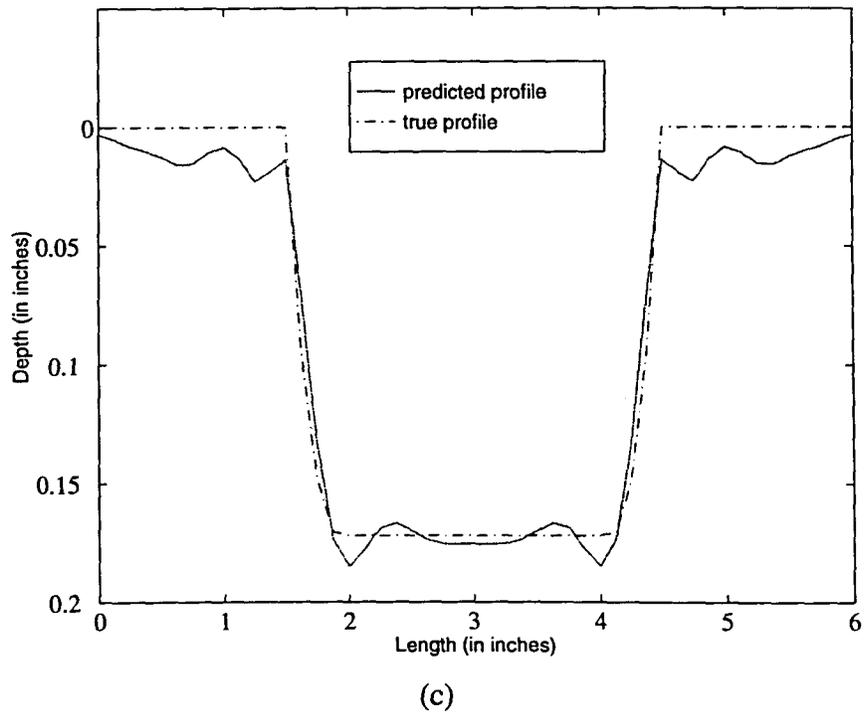
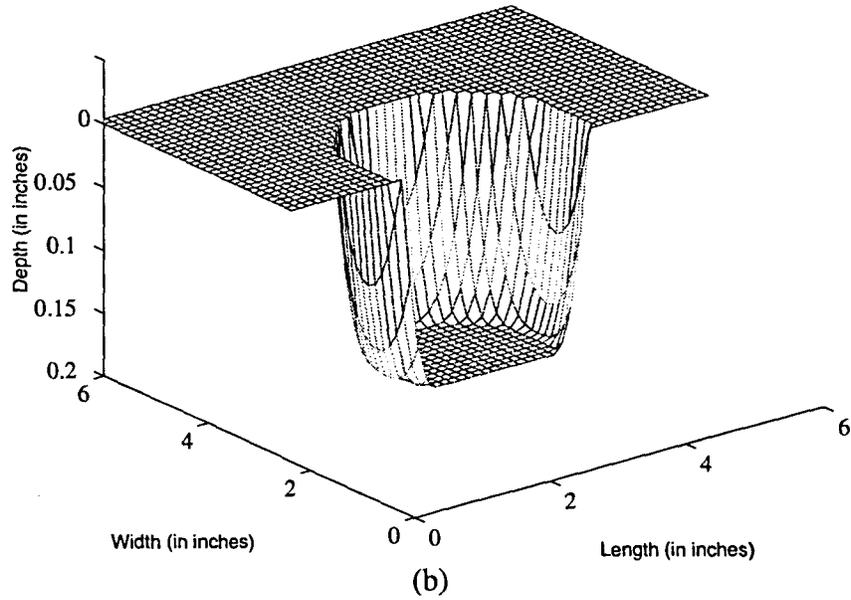


Figure 5.9 (continued) (b) The true profile. (c) 1-D section of both the predicted and true profile along the axial direction.

5.4 Modified Optimal RBF Network.

So far, not all the information of the MFL signals has been used to predict defect profiles. It is well known that the length, width and depth have significant effect on the corresponding MFL signal [21]. Therefore, an MFL signal could be used to predict the maximum length, width, and depth of the corresponding defect profile. This method is not similar to the approach using a parametric representation of the profile discussed earlier in which *a priori* knowledge about the defect profile is needed.

The modified optimal RBF network combines two networks together. One is the optimal RBF network, using simulation annealing in the frequency domain, which predicts the general shape of a defect. The other is the optimal RBF network, using the steepest descent algorithm in the time domain, which predicts the maximum depth, width and length. The predicted depth is used to modify the general shape of a defect. A block diagram for this RBF network architecture is shown in Figure 5.10. The predicted defect profile obtained using the MFL signal shown in Figure 5.8 is depicted in Figure 5.11(a). The results show that additional improvement in performance can be obtained. Figure 5.11(b) shows the 1-D section of both the predicted and true profiles along the axial direction. The error percentages of depth, length, and width prediction are about 10%, 13%, and 15%, respectively. Figure 5.12 illustrates another defect profile predicted using this network. The corresponding error percentages relating to depth, length, and width predictions are about 6%, 1%, and 16%, respectively.

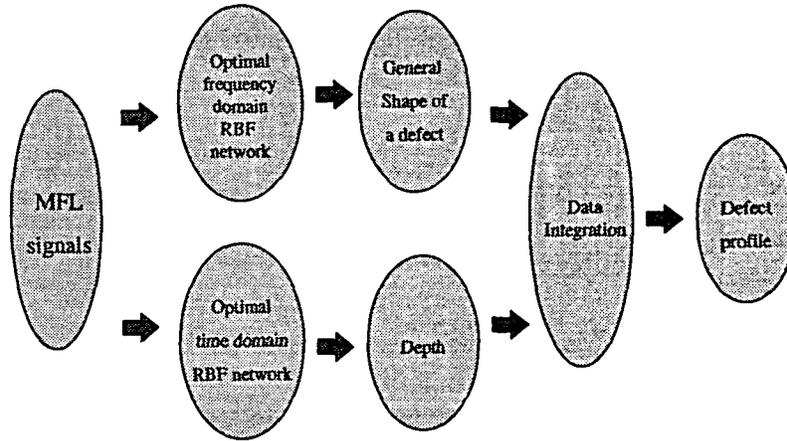


Figure 5.10 Block diagram for the modified RBF network .

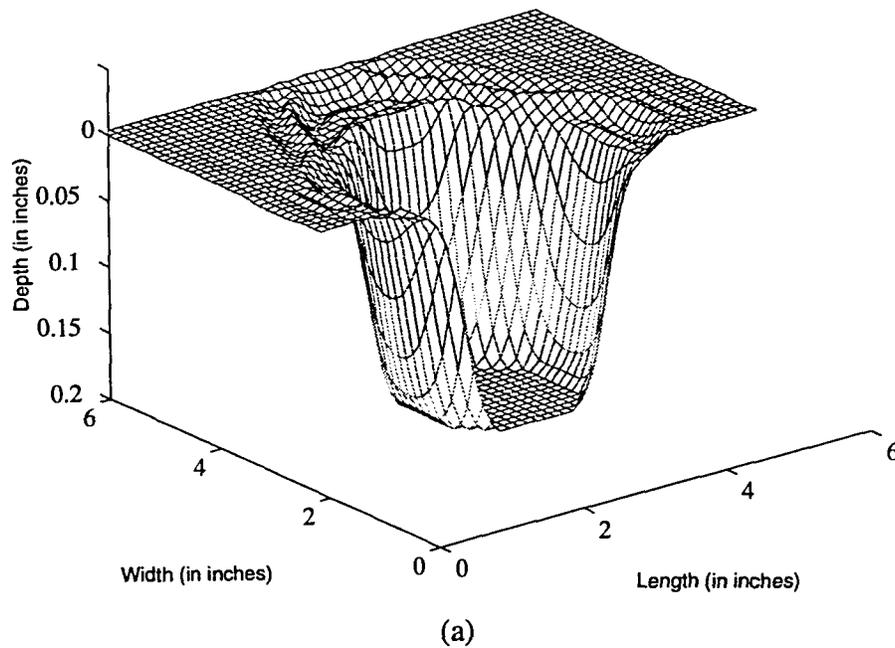


Figure 5.11 (a) Defect profile predicted using the modified RBF network.

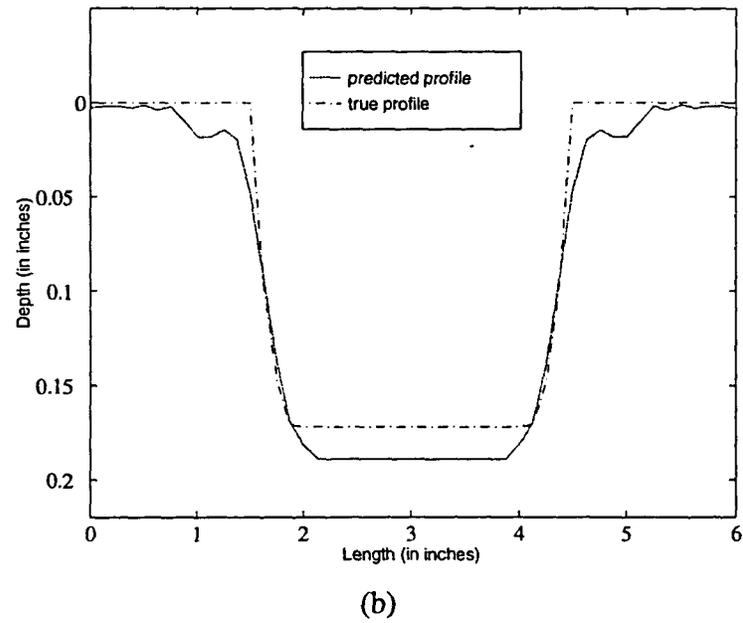


Figure 5.11 (continued) (b) 1-D section of both the predicted and true profiles along the axial direction.

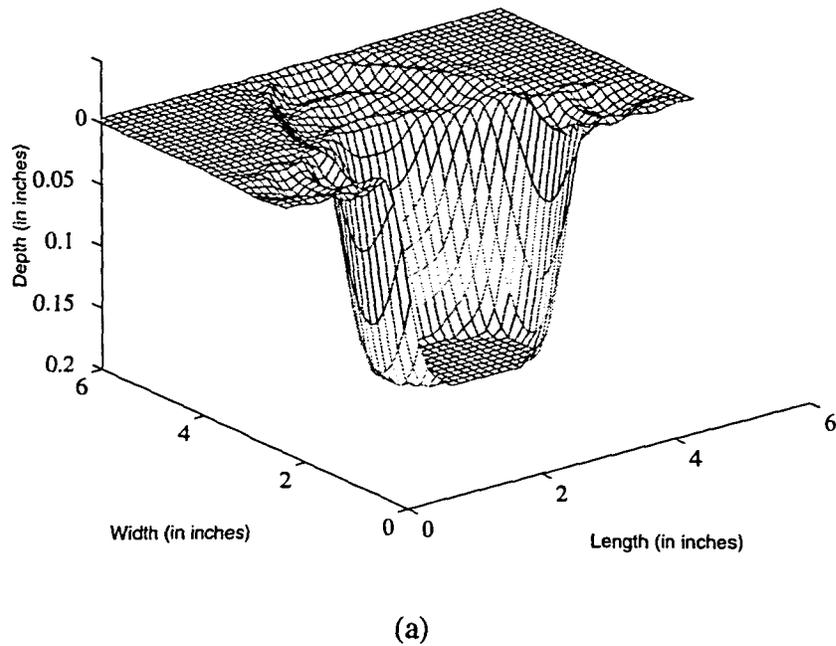
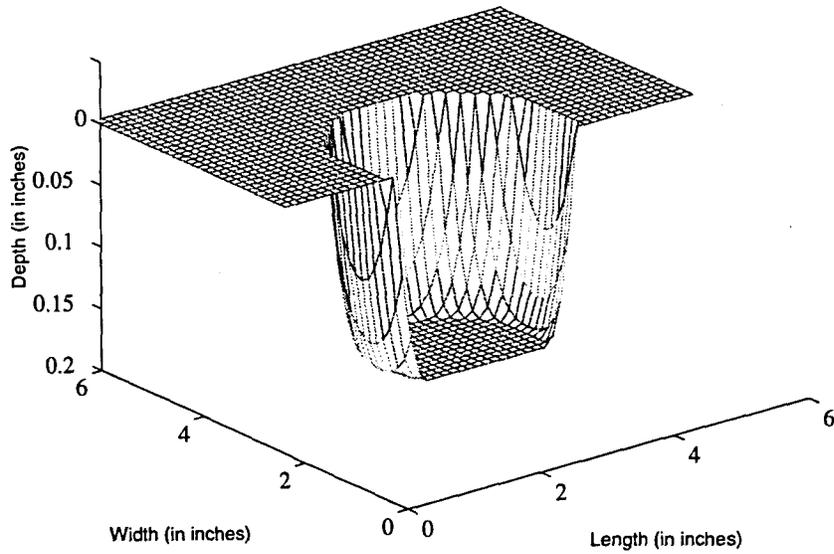
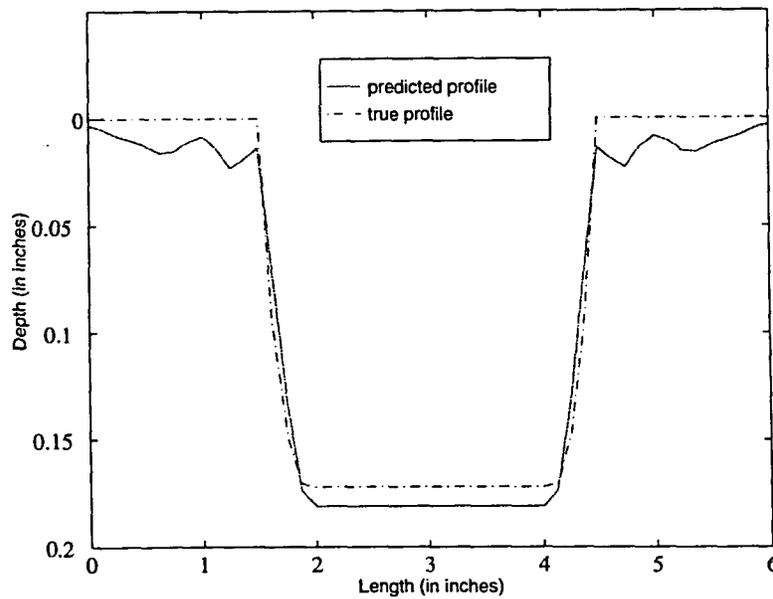


Figure 5.12 (a) Defect profile predicted using the modified RBF network.



(b)



(c)

Figure 5. (continued) (b) Corresponding true profile. (c) 1-D section of both the predicted and true profiles along the axial direction.

CHAPTER 6. DISCUSSION AND CONCLUSION

6.1 Summary

The work described in this thesis was motivated by a desire to design and developed a neural network which is able to generate defect profiles from MFL signals. To solve the inverse problem of mapping MFL signals to defect profiles, an optimal RBF neural network was developed using different optimization techniques, such as the steepest descent and simulated annealing methods.

Two general categories of optimal RBF networks are presented in this thesis. The first type belongs to the class of time domain networks. In this class, both non-optimal as well as optimal versions of the network were developed. As a first step towards characterizing MFL signals, a non-optimal network was designed using the matrix pseudo-inverse method. In order to improve the performance of the network, the steepest descent algorithm was used to optimize the location of cluster centers and the widths of radial basis functions, as part of the procedure for optimizing the weights of the output layer of the network. The initial values of network parameters were obtained using either the K-Means or an adaptive K-Means clustering algorithm. The latter method is a variant of the conventional K-Means algorithm and is able to optimize the number of cluster centers.

The time domain optimal RBF network performs relatively well in predicting defect profiles. Its performance can be improved by restricting attention to defects that have regular shapes, and have the network predict only the geometric parameters describing the defect instead of the full profile. A network, capable of mapping the MFL signal to five geometric parameters of the corresponding defect was implemented and evaluated.

The second category of optimal networks presented in this thesis maps signals in the frequency domain. The spectra of both MFL signals and defect profiles show that most of the energy is concentrated at low frequencies. This characteristic was exploited by designing an RBF network to map the MFL spectrum to the defect profile. The defect profile was then reconstructed from its predicted spectrum. Simulated annealing techniques were used to optimize the frequency domain RBF neural network. The initial parameters of the optimal RBF network were obtained using the K-Means algorithm in the frequency domain.

Table 6.1 lists a defect profile predicted using different RBF networks. From this table, we notice considerable improvement in the RBF network performance when optimal networks are used, especially, when we operate in the frequency domain.

Table 6.1 Comparison of various defect characterization strategies

	Direct Mapping		Geometric Mapping		Spectral Mapping		Actual Profile
	Prediction (in inches)	Error (%)	Prediction (in inches)	Error (%)	Prediction (in inches)	Error (%)	
Length	6.00	100.0	2.60	13.3	3.03	1.0	3.00"
Width	6.00	100.0	3.30	10.0	3.02	0.7	3.00"
Depth	0.25	11.6	0.216	3.6	0.218	3.0	0.244"

6.2 Discussion and Future Work

Future work in this research project should focus on obtaining additional improvement in the neural network performance. The optimal RBF network technique used in this research project is much better than other techniques described in Chapter 1. However, there are several areas where improvements could be made.

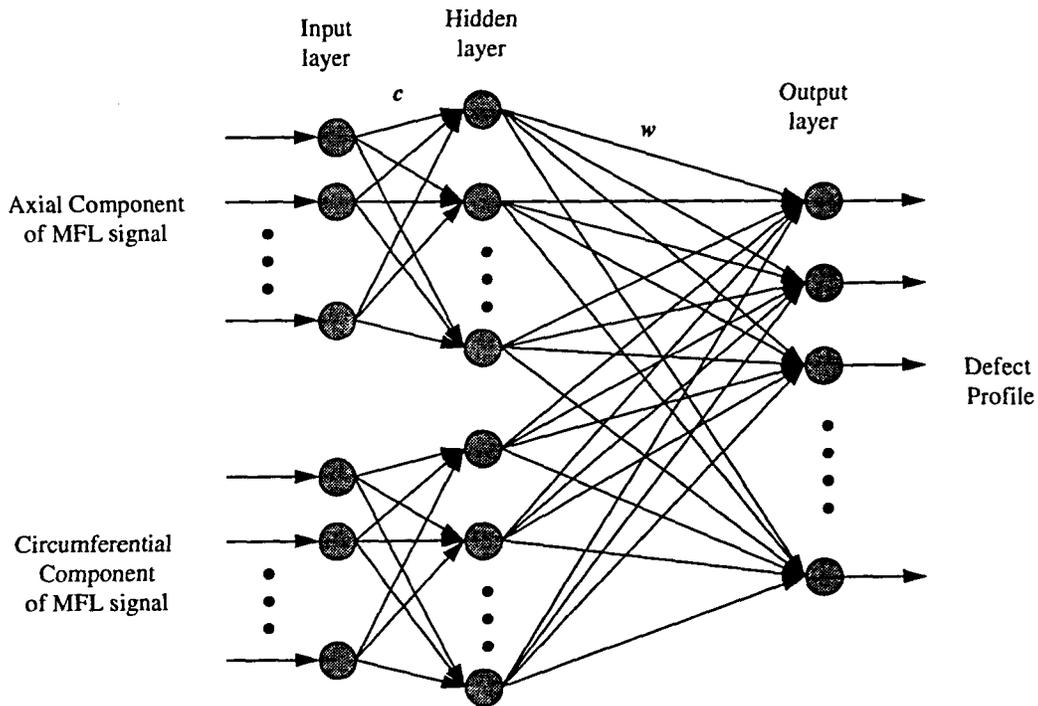


Figure 6.1 Proposed neural network for using both axial and circumferential components of MFL signals.

The study of the predicted defect profiles shows that the length of a defect can be predicted fairly accurately, while the predictions of the width and the depth of the flaw are not as good. These results are due to the fact that only the axial components of the MFL signals are used in the analysis. Additional information relative to the width of a defect can be obtained from the circumferential component of the MFL signal. In order to improve the prediction of width and depth, this thesis suggests that both the radial components and the circumferential components of MFL signals be used to train the neural network in addition to the axial component. Figure 6.1 proposes a neural network architecture using both axial and circumferential components. The system consists of two networks, one of which mainly predicts the length and depth, while the other mainly predicts the width and depth of the defect. The depth predictions of both networks can be combined to improve the defect characterization.

As mentioned earlier, the width, σ , of a Gaussian radial basis function determines the localization of the basis function. Increasing σ leads to more overlap between neighboring clusters. Therefore the width plays a crucial role in the performance of Gaussian RBF networks. Several methods can be used to choose the width parameter. A straightforward method is to use half the distance between the two nearest centers as the width. Another method that was used in this work is to choose a set of random values as the initial values of widths of these cluster centers and use an algorithm to optimize these values. However, since the cost function E is not a quadratic function of c and σ , the result usually gets trapped in a local minimum. Consequently, the initial choice of values for the widths of the Gaussian radial basis functions is critical. The best performance is achieved by performing several trials and choosing the best result.

The width of each cluster is a scalar quantity, which implies that the shape of the Gaussian basis function looks like a multivariate normal whose covariance matrix is an identity matrix. Its contour is a hypersphere in a high-dimensional space. This is often not the best approach. In real situations, the contour should be a hyper-ellipsoid. The deviation along the principal axis reflects the spread of the basis function. If this concept is used, the radial basis function could be described in the most general form:

$$\phi(\mathbf{x}) = \exp\left[-\frac{(\mathbf{x} - \mathbf{c})^T \cdot \Sigma^{-1} \cdot (\mathbf{x} - \mathbf{c})}{2}\right] \quad (6.1)$$

where Σ is the covariance matrix of \mathbf{x} .

The use of the above radial basis functions will make the RBF network even more complicated, and the training time will increase significantly. This concept was actually evaluated. However the performance was not found to be as good as obtained with other approach. The concept needs to be investigated more rigorously.

The objective of this research project is to find the weights, w , of RBF networks. Considering equation (3.16), we find that the number of equations is less than the

number of unknowns. The problem is clearly under-specified. Some constraints must therefore be introduced to obtain a unique solution. One method is to use vector basis functions rather than Gaussian radial basis functions. This approach may yield more satisfactory results.

REFERENCES

1. H. H. Haines, T. A. Bubenik and J. B. Nestleroth, "Gas Research Institute's Research and Development Program on In-line Inspection of Nature Gas Pipelines," *Characterization of Magnetic Flux Leakage (MFL) Indication Found During In-line Inspection of Natural Gas Transmission Pipelines*, Monthly Report, July, 1995.
2. L. Jackson and R. Wilkins, Presented at the 55th Autumn Meeting, Kensington Town Hall, Nov. 1989.
3. G. A. Alers and L. R. Burns, "EMAT Design for Special Applications," *Materials Evaluation*, Vol. 45, p1184, 1987.
4. B. W. Maxfield, A. Kuramoto, and J. K. Hulbert, "Evaluating EMAT Designs for Selected Applications", *Material Evaluation*, Vol. 45, p1166, 1987.
5. W. Lord and D. J. Oswald, "Leakage Field Methods of Defect Detection," *International Journal of Nondestructive Testing*, Vol. 4, p249, 1972.
6. J. F. Hinsley, *Non-destructive Testing*, MacDonald and Evans Press Ltd., London, 1959.
7. J. B. Nestleroth, S. W. Rust, and D. A. Burgoon, "Determining Corrosion Defect Geometry from Magnetic Flux Leakage Pig Data," *Characterization of Magnetic Flux Leakage (MFL) Indication Found During In-line Inspection of Natural Gas Transmission Pipelines*, Monthly Report, Oct., 1995.
8. M. M. Chao, *Neural Networks for Characterizing Magnetic Flux Leakage Signals*, M.S. Thesis, Iowa State University, 1995.
9. R. P. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE Acoustics, Speech and Signal Processing Magazine*, Vol. 61, p4, 1987.
10. S. Haykin, *Neural Networks, A Comprehensive Foundation*, Macmillian College Publishing Company, Inc., Englewood Cliffs, New Jersey, 1994.

11. W. Lord, S. Nath, Y. K. Shin, and Z. You, "Electromagnetic Methods of Defect Detection," *IEEE Transaction on Magnetics*, Vol. 26, p2070, 1990.
12. H. L. Libby, *Introduction to Electromagnetic Nondestructive Test Methods*, Robert E. Krieger Publishing Company, Huntington, New York, 1979.
13. J. D. Jackson, *Classical Electrodynamics*, 2nd Ed., John Wiley & Sons, New York, 1975.
14. Lord, and J. H. Hwang, "Finite Element Modeling of Magnetic Field/Defect Interactions," *ASTM Journal of Testing and Evaluation*, Vol. 3, No. 1, p. 21, 1975.
15. N. Zatsepin, and V. E. Shcherbinin, "Calculation of the Magnetostatic Field of Surface Defects I," *Defektoskopiya*, No.5, p50, Sept. 1966.
16. N. Zatsepin, and V. E. Shcherbinin, "Calculation of the Magnetostatic Field of Surface Defects II," *Defektoskopiya*, No.5, p59, Sept. 1966.
17. Udpa, and W. Lord, "A Discussion of the Inverse Problem in Electromagnetic NDT," *Review of Progress in Quantum Nondestructive Evaluation*, edited by D. O. Thompson and D. E. Chimenti, Vol. 5A, p375, Plenum Press, New York, 1986.
18. Udpa, and W. Lord, "An AI Approach to the Eddy Current Defect Characterization Problem," *Review of Progress in Quantum Nondestructive Evaluation*, edited by D. O. Thompson and D. E. Chimenti, Vol. 6A, p899, Plenum Press, New York, 1987.
19. Murphy, H. A. Sabbagh, and J. C. Treece, "Some Inversion Problems in Nondestructive Evaluation," *Review of Progress in Quantum Nondestructive Evaluation*, edited by D. O. Thompson and D. E. Chimenti, Vol. 14A, p857, Plenum Press, New York, 1995.
20. E. Beisner, et al., "NDE Applications of Magnetic Leakage Field Methods, A State of the Art Survey," *Report from Southwest Research Institute*, Southwest Research Institute, San Antonis, Texas, 1980.
21. Lord and J. H. Hwang, "Defect Characterization from Magnetic Leakage Fields," *British Journal of NDT*, Vol. 19, No. 1, p14, 1977.
22. M. J. Koenig, T. A. Bubenik, S. W. Rust and J. B. Nestleroth, *Topical Report, GRI Pipeline Simulation Facility Metal Loss Defect Set*, Gas Research Institute, Chicago, 1995.

23. G. J. Posakony, *Topical Report, Assuring the Integrity of Natural Gas Transmission Pipelines*, Gas Research Institute, Chicago, 1992.
24. S. Nair, S. Udpa, and L. Udpa, "Radial Basis Functions Network for Defect Sizing," *Proceedings of the Review of Progress in Quantitative Nondestructive Evaluation*, San Diego, 1992.
25. D. S. Broomhead, and D. Lowe, "Multivariate Functional Interpolation and Adaptive Networks" *Complex Systems*, Vol. 2, p321, 1988.
26. M. J. D. Powell, "Radial Basis Function Approximations to Polynomials," *Numerical Analysis 1987 Proceedings*, p223, Dundee, UK, 1988.
27. A. Ben-Israel, *Generalized Inverses: Theory and Applications*, Robert E. Krieger Publishing Company, Huntington, New York, 1980.
28. S. Shekhar, M. B. Amin, and P. Khandelwal, "Generalization Performance of Feed-Forward Neural Networks," *Neural Networks: Advances and Applications 2*, E. Gelenbe, Ed., Elsevier Science Publishers B. V., Amsterdam, The Netherlands, p13, 1992.
29. E. W. Forgy, "Cluster Analysis of Multivariate Data: Efficient Versus Interpretability of Classification," Paper Presented at Biometric Society Meeting, Riverside, CA, 1976
30. J. B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, p281, 1967.
31. H. Spath, *Cluster Analysis Algorithm for Data Reduction and Classification of Objects*, Elis Horwood Publishers, New York, 1980.
32. J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1974.
33. L. Udpa, "Class Notes of EE529G," Fall Semester, Iowa State University, 1995.
34. C. Chinrungrueng and C. H. Sequin, "Optimal Adaptive K-means Algorithm with Dynamic Adjustment of Learning Rate," *IEEE Transactions on Neural Networks*, Vol. 6, No. 1, p157, 1995.
35. S. S. Rao, *Optimization Theory and Applications*, Wiley Eastern Limited, New Delhi, India, 1978.

36. L. Cooper and D. Steinberg, *Introduction to Methods of Optimization*, W. B. Saunders Company, Philadelphia, 1970.
37. S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, Vol. 220, No. 4598, p671, 1983.
38. S. K. Ma, *Modern Theory of Critical Phenomena*, Benjamin, New York, 1982.
39. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, *Journal of Chemical Physics*, Vol. 21, p1087, 1953.
40. E. Arts and J. Korst, *Simulated Annealing and Boltzman Machines, A Stochastic Approach to Combinatorial Optimization and Neural Computing*, John Wiley & Sons, New York, 1989.
41. B. A. Auld and S. Jefferies, "Semi-elliptical Surface Flaw EC Interaction and Inversion: Theory," *Review of Progress in Quantitative Nondestructive Evaluation*, Vol. 5A, p383, Plenum Press, New York, 1986.
42. H. A. Sabbagh, L. D. Sabbagh and T. M. Roberts, "A Multifrequency Algorithm for Inverting Eddy-Current Data and Reconstructing Three-Dimensional Anomalies in Graphite-Epoxy", *Review of Progress in Quantitative Nondestructive Evaluation*, Vol. 5A, p409, Plenum Press, New York, 1986.