This is to certify that the
dissertation entitled

INVERSE PROBLEMS IN NON-DESTRUCTIVE EVALUATION OF GAS TRANSMISSION PIPELINES USING MAGNETIC FLUX LEAKAGE

presented by

Ameet Vijay Joshi

has been accepted towards fulfillment
of the requirements for the

Doctoral degree in Philosophy

Lalita Udpa
Major Professor's Signature

November 9, 2006
Date

MSU is an Affirmative Action/Equal Opportunity Institution

INVERSE PROBLEMS IN NON-DESTRUCTIVE EVALUATION OF GAS TRANSMISSION PIPELINES USING MAGNETIC FLUX LEAKAGE

By

Ameet Vijay Joshi

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Electrical and Computer Engineering Department

2006

ABSTRACT

INVERSE PROBLEMS IN NON-DESTRUCTIVE EVALUATION OF GAS TRANSMISSION PIPELINES USING MAGNETIC FLUX LEAKAGE

By

Ameet Vijay Joshi

The critical issue in non-destructive evaluation (NDE) is the characterization of materials on the basis of information derived from the response of energy-material interactions without affecting the quality or utility of the test piece. This is commonly referred to as an "inverse problem". Inverse problems are in general ill-posed and full analytical solutions to these problems are seldom tractable. Practical solutions however employ constrained search techniques to find the optimal solution from the set of possible solutions. NDE signals recorded by the sensors are influenced not only by internal defects and degradations, but they also tend to be marred by the noise originating from various sources. Hence inverse problems in NDE need to deal with filtering techniques to obtain a noise-free signal. In this dissertation, NDE of natural gas transmission pipelines is considered.

Natural gas is transported to consumer sites through a vast network of pipelines. In order to ensure integrity of the system, the pipelines are periodically examined using Magnetic flux leakage (MFL) technique. The MFL inspection assembly is commonly referred to as "pig". In the MFL inspection tool, permanent or electromagnets are used to magnetize the pipe-wall in an axial direction and an array of Hall-effect sensors is usually installed around the circumference of the pig to sense the leakage flux caused by anomalies in the pipe-wall. The signal picked up by the sensor array is recorded and subsequently analyzed

by trained analysts. Traditional methods involving manual analysis of this signal are time consuming. The performance of these methods cannot be standardized and is subject to change depending on the levels of skill and training of the analyst. The gas pipeline industry is therefore interested in automated methods for analyzing the MFL inspection signal in order to improve accuracy and decrease the turn around time between the actual pigging and receipt of inspection results.

MFL signals obtained from the pig are contaminated with noise from various sources. The variation in magnetic properties of seamless pipes introduces a quasi-periodic noise called as seamless pipe noise (SPN). Lift-off variations in sensors due to motion of pig inside the pipe and noise from electronic system hardware contribute to additional noise in the signal. Hence signal interpretation is carried out in two steps:

1. Noise removal and identification of regions of interest (ROIs) enclosing potential defects.
2. Inversion of the signal in the ROI to predict full 3-dimensional depth profile.

This dissertation discusses the conventional methods of noise removal and their limitations when applied to the MFL signal. A new method based on higher order statistics (HOS) is introduced and described; its advantages over conventional methods are discussed.

Two approaches, namely direct and iterative inversion methods are presented in the second step of inverting the MFL signal to predict the defect depth profile. Both methods are based on the use of radial basis function neural network (RBFNN).

The challenge of high dimensionality (of the order of few thousands) is addressed by modifying traditional approaches described in literature. These modifications in both direct inversion as well as iterative inversion are new contributions to the field.

To,

My father, ***Mr. Vijay Joshi***
and my mother, ***Mrs. Madhuri Joshi***
whose encouragement and support paved the way to this success !!!

ACKNOWLEDGMENTS

I would like to express my sincere gratitude towards my advisor, Dr. Lalita Udpa for her expert guidance and encouragement throughout the duration of my PhD. Together with Dr. Satish Udpa, they have created a wonderful atmosphere for research in the Nondestructive Evaluation (NDE) Lab at Michigan State University, that accelerated my research.

I would like to thank Dr. Satish Udpa for his insightful suggestions during our weekly lab meetings and also being part of my thesis committee. I would also like to thank Dr. Pradeep Ramuhalli for being available for help any time I needed and being part of my committee. Thanks are also due to the fourth committee member Dr. Gary Cloud for his comments and suggestions regarding the dissertation.

I would also like to thank my fellow lab mates: Ravi Dayana, Vikram Melapudi, Kavitha Arunachalam and Zhiwey Zeng for helping me in our overlapping areas of research. Many interesting discussions that we had helped in strengthening the learned concepts.

I would like to thank Electric Power Research Institute (EPRI) for funding my research as I started with my PhD and Microline Technology Corporation for funding my research that led to the completion my PhD. I would like to extend my regards towards Doug Spencer (President, Microline), Bruce Girrell (Vice President, Microline) and Jabin Reinhold (Software Engineer, Microline) for helping me with the resources that I needed from time to time during the research. I would like to thank Microline for further encouraging and endorsing my research by offering me a job opportunity at Microline even before my graduation.

My parents and brother were always there for me whenever I needed the moral support or any other difficulty and without them it would have been impossible for me complete the PhD. Last, but not the least, I would like to express my deep regards towards my wife, Meghana, who was always right beside me with continuous encouragement and praise. Her company made the whole monotonous journey towards PhD an exhilarating experience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 About NDE

Non-destructive evaluation (NDE) deals with the inspection of the interiors of materials without compromising their usefulness. Figure 1.1 shows the block diagram of a generic NDE system.

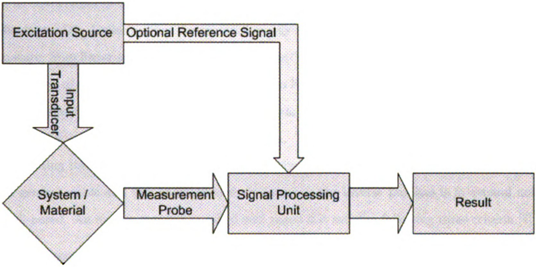


Figure 1.1. Basic block diagram of an NDE system.

An input transducer or excitation source is used to inject energy into the test sample.

The energy can be in the form of microwave, ultrasound, electric current or magnetic flux as is used in this thesis. The signal applied by the excitation source is modulated by the material being inspected. The response of the material-energy interaction is sensed by the measurement probe and the captured signal is analyzed to detect any existent flaws and estimate their size.

1.1.1 Inverse Problem in NDE

The signal flow in a typical NDE system is schematically shown in Figure 1.1.

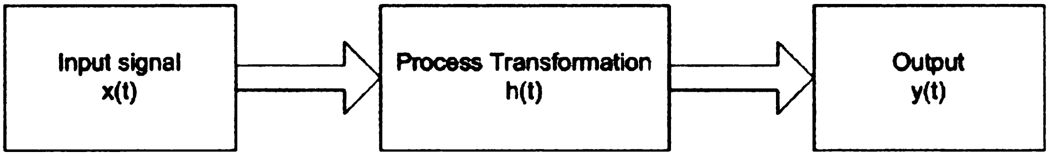


Figure 1.2. Systems approach for the signal processing block in NDE.

The excitation signal denoted as $x(t)$ is applied to the material sample to be inspected. The characteristics of the material are represented mathematically in the form of a transfer function $h(t)$. Assuming that the material energy interaction is linear and time invariant in nature, from linear systems theory, the output $y(t)$ is mathematically represented using the operation of convolution as, $y(t) = x(t) * h(t)$. In NDE the objective is to accurately predict the transfer function $h(t)$ by measuring the system output $y(t)$ in response to a known input $x(t)$. This is an example of an inverse problem. The forward problem, in general aims at estimating the response $y(t)$ corresponding to given $x(t)$ and $h(t)$. The forward problem is in general stable and well characterized. In contrast, the inverse problem is in general not well-posed. An inverse problem is called well-posed if it satisfies following three criteria [4].

1. *Existence*: For every finite input there exists a finite output.
2. *Uniqueness*: For every finite input there is a unique output. This criterion guaranties one-to-one mapping.

3. *Continuity*: When input signal is continuous the output is also continuous. This assures existence of finite gradient of the output.

If the inverse problem is not well-posed it is called as ill-posed. Most often, the solution of an ill-posed inverse problem is difficult to obtain and different types of regularization techniques are employed to improve the accuracy of the solution. Conceptually, regularization can be defined as: construction of approximate solution of an ill-posed inverse problem that is stable with respect to small perturbations of input data. The process of regularization imposes constraints on the solution space to keep the solution within the empirically selected bounds. The solution of inverse problem in NDE is therefore an area of significant research interest. The following section describes the details of NDE of gas transmission pipelines.

1.2 NDE of Gas Pipeline Inspection

Thirty percent of the energy produced in the country is conveyed to the end user through a vast network of pipelines [31]. There are 90,000 miles of natural gas gathering and field pipelines, 280,000 miles of gas transmission pipelines and over 835,000 miles of gas distribution mains and service lines [31]. The main component of the pipeline system is the transmission and distribution pipelines which transport huge amounts of high pressure gas (pressures measuring up to 1400 psi [31]) over long distances. The reliable operation of these pipelines is extremely important from safety and economic perspectives. A typical cost for the replacement of a 16 inch diameter pipe can range from \$250,000 to \$500,000 per mile, while that of a 30 inch diameter pipe can range from \$600,000 to \$3,000,000 per mile. Preventive maintenance can reduce these costs to a great extent and also reduce accidental deaths.

Instrumented probes commonly referred to as 'pigs' [31] are used for surveying and monitoring pipeline conditions. The first inspection of any pipeline is called as baseline inspection. Subsequent inspections can be compared with baseline inspection to determine the growth of the defects in the pipeline. This method is used for preventive maintenance of the pipes.

Gas transmission pipelines are typically inspected using the magnetic flux leakage (MFL) technique because it is robust and can detect anomalies encountered in ferromagnetic materials. The capabilities of MFL technique with respect to some of the commonly encountered anomalies are listed in Table 1.1.

Table 1.1. Anomalies detected with MFL.

Type of Anomaly	MFL Capability
Metal Loss (corrosion, gouges)	Generally well suited.
Metallurgical	Occasionally found.
Cracks (axial)	Not well suited.
Cracks (circumferential)	Occasionally found when inside.
Dents and Buckles	Detectable.

A schematic of the in-line inspection (ILI) pig mounted with MFL detection assembly including “excitation source” and “measurement probe” is shown in Figure 1.3.

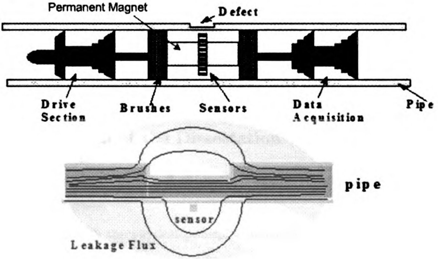


Figure 1.3. NDE of gas pipeline. (This image is taken from web site www.battelle.org)

MFL pigs are designed to detect and characterize internal and external metal loss due to various causes. To inspect a pipe, ferromagnetic material of the pipe-wall is magnetized in the axial direction using permanent magnets as shown in Figure 1.3. The level of magnetization is maintained such that the pipe-wall in contact is completely saturated. When the pig passes through a defective region (i.e. region with metal loss) some of the flux leaks into the surrounding air as shown in Figure 1.3. The axial component of the leakage flux is measured using a circumferential array of Hall-effect sensors. The measured Hall voltage is a measure of the leakage flux density and constitutes the MFL signal. In a defect free region, there is no leakage and the a low uniform amplitude signal is measured. In the presence of metal loss corrosion, the flux 'leaks' into the air producing a local change in the measured MFL signal. The information in this signal can be utilized to detect and characterize anomalies in the pipe wall.

MFL signals measured as the pig moves through the pipe are often corrupted with noise from various sources. Seamless pipe noise (SPN) due to variation in the magnetic properties of the pipe material, sensor lift-off variations and system noise due to on-board electronics are prominent sources of noise and artifacts. Signal interpretation proposed and implemented here is a two step process: (1) Segmentation of the regions in MFL signal having potential defects, (2) Solution of the inverse problem to predict the defect profile from the information contained in the segmented MFL data.

1.3 Organization of the Dissertation

The dissertation is organized into 6 chapters. Chapter-1 begins with introduction and statement of the problem.

Chapter-2 deals with the physical principles underlying the MFL inspection method. The physics of magnetic flux leakage and its detection is explained briefly along with the instruments that are used for this purpose by the industry. This chapter also describes the challenges in MFL inspection in detail and states the precise scope of the dissertation.

Chapter-3 begins with a discussion regarding the post processing of raw MFL signals obtained from MFL inspection. This is followed by segmentation of signal, using various filtering techniques. The segmentation process identifies and segments the regions of interest (ROIs) enclosing potential defects. The compensation of MFL signal to make it invariant to changes in wall thickness, permeability and velocity are also discussed here.

Chapter-4 focusses on solution of the inverse problem using a direct inversion technique. Radial basis function neural network (RBFNN) is used for predicting the 3-dimensional depth profile of defect from the two dimensional MFL data. The traditional model of the RBFNN is first described and its limitations with respect to handling of large dimensional input and output vectors are discussed. Modifications of the traditional architecture and algorithms are then proposed and described. The traditional and modified approaches are implemented and tested on experimental data. The results are discussed.

Chapter-5 focusses on the solution of the inverse problem using an iterative inversion technique. The RBFNN is used as an intermediate tool to model the forward problem of predicting the MFL signal for a given defect geometry. The simple iterative inversion approach is described first. This approach is found to converge very slowly due to high dimensionality of the data. A new approach involving adaptive use of two-dimensional wavelet transform is proposed. Both the approaches are tested on experimental data and the results are discussed.

Chapter-6 presents the conclusions and also indicates the scope of the future work in this field of research.

CHAPTER 2

Magnetic Flux Leakage Inspection

2.1 Brief History of Magnetic Flux Leakage Inspection

Magnetic flux leakage techniques were used as early as 1868 by the Institute of Naval Architects in England, where defects in magnetized cannon tubes were found with a compass [10]. In 1918, magnetic particle inspection was accidentally discovered [19]. Magnetic particles (iron shavings) were held in place by a local change in magnetic flux at the surface near a defect. The development and commercialization of magnetic particle inspection followed soon after [8], [9]. MFL inspection systems are based on the same principles as magnetic particle inspection. The main difference between MFL and magnetic particle inspection is: MFL methods use sensors for quantitative measurements of the magnetic leakage field while the magnetic particle inspection methods do not. MFL sensors, which were developed in the 1920's and 30's, [8] measure the leakage magnetic field around a defect. Sensors allow quantitative measurement, as opposed to the qualitative information provided by particles. The first MFL in-line inspection (ILI) tool for pipelines was introduced in 1965 by Tuboscope [31]. Since 1965, MFL tools have gained wide acceptance by the gas transmission industry and since then there has been a consistent growth in their performance.

2.2 Theoretical Aspects of Magnetic Flux Leakage (MFL) Inspection

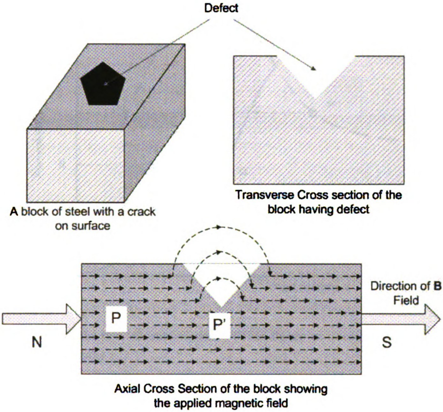


Figure 2.1. A block of steel with a defect on surface.

The theoretical aspects of Magnetic Flux Leakage (MFL) were first investigated by Lord et. al. in 1972 [25]. A ferromagnetic sample to be inspected is first magnetized either using a yoke, current carrying conductor or by a permanent magnet. Consider a block of steel, having a crack on its surface as shown in Figure 2.1. Transverse and axial cross sections of the block are also shown for clarity. Magnetic flux density B is induced axially using the permanent magnets placed at the ends as shown. In the defect region, the cross sectional area available for the passage of flux reduces, thereby increasing the flux density from B to

B_1 as shown in B-H curve in the Figure 2.2.

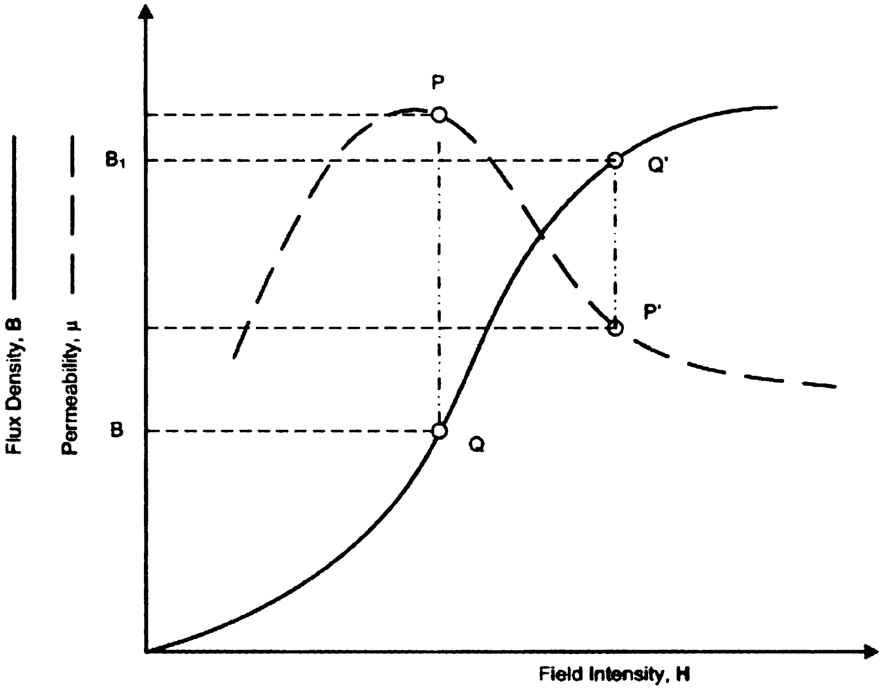


Figure 2.2. Magnetization characteristics. (This figure is obtained from the website http://www.battelle.org)

Consequently, permeability of the material reduces from P to P'. Increased induced flux in the defect region, reduces permeability of the material in the vicinity. Since flux lines follow the path of least reluctance, some of the flux leaks into the surrounding medium. This leakage flux is measured with the use of Hall-effect sensors (The details of Hall-effect are discussed in Section 2.2.1.) as described in the following subsection. The measured leakage flux signal is called as MFL signal. As described in [25], the leakage flux distribution is a complex function of the defect geometry. The inverse problem associated with prediction of defect geometry from knowledge of the MFL signal is often ill-posed. Figure 2.3 shows variation in the leakage flux distribution with increase in applied magnetic field. The more the applied field strength the more is the leakage flux, and more is the accuracy of the measurement. In most practical cases, in order to get maximum leakage flux, a strong magnetic field is applied to keep the material in saturation. In saturation region the leakage flux is insensitive to the magnetizations of the sample.

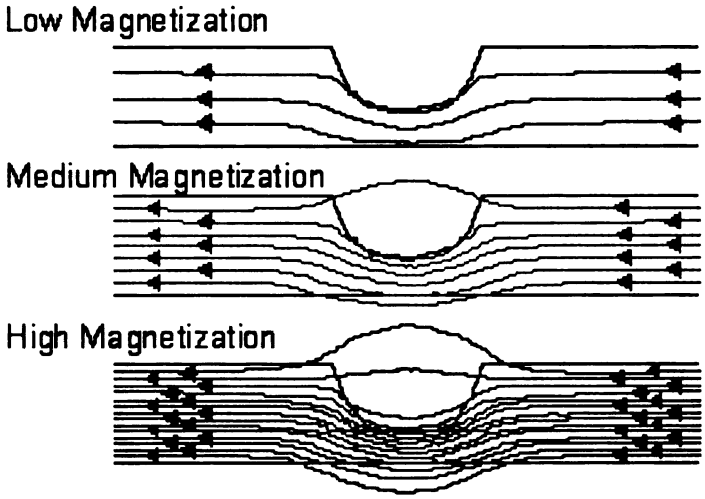


Figure 2.3. Flux leakage at various magnetization levels.

2.2.1 Hall-Effect

Hall effect is named after an American physicist Edwin H. Hall. In 1879 Hall observed that when a current carrying conductor is placed in a magnetic field, an electric voltage (now called as Hall Voltage) is developed at the ends of the conductor perpendicular to the direction of current and the magnetic field. Details of the quantitative relationship between Hall Voltage and other factors involved is described below.

Consider a rectangular conducting plate of thickness d as shown in Figure 2.4. Let current I be passed along the length of the material. The direction of the magnetic field B is perpendicular to the direction of current as shown. The magnetic field exerts a transverse force F_m on the moving charge carriers in the conductor. Separation of the charge carriers towards the sides of the conductor produces an electric force F_e that balances the magnetic force F_m. Voltage developed at the ends of the conductor due to separation of the charge carriers can be measured. This voltage is called as *Hall Voltage* V_H. Let n be density

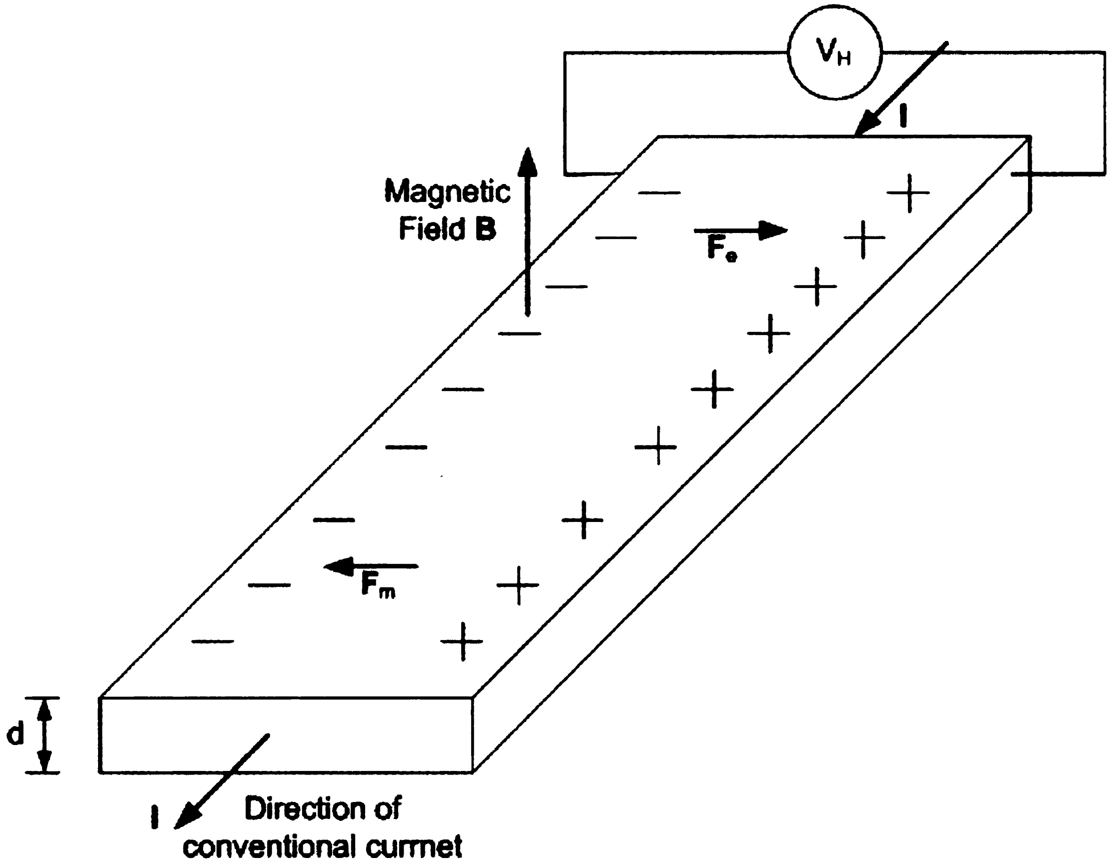


Figure 2.4. Theory of Hall-effect.

of the mobile charges on the conductor and e be the electric charge on an electron. The quantitative expression for the Hall voltage is given as,

$$V_H = \frac{IB}{ned} \tag{2.1}$$

When all the variables in Equation 2.1 except B are constant, the Hall voltage is directly proportional to the applied magnetic field. Using this principle a probe can be developed to measure the magnetic field in terms of Hall voltage. As current must be passed through the conductor in order to observe the Hall voltage, the Hall-effect sensor is an active sensor. Hall effect sensors are widely used in the MFL inspection assembly.

2.3 MFL Inspection of Gas Pipelines

The MFL tool used for inspection of gas transmission pipelines carries a permanent magnet as an excitation source for the axial magnetization of pipe-wall. Leakage flux is measured using a circumferential array of Hall-effect sensors. The inspection assembly is mounted on a commercial tool called a pig. Figure 2.5 shows the schematic of MFL inspection assembly on a pig and Figure 2.6 shows full schematic of the pig.

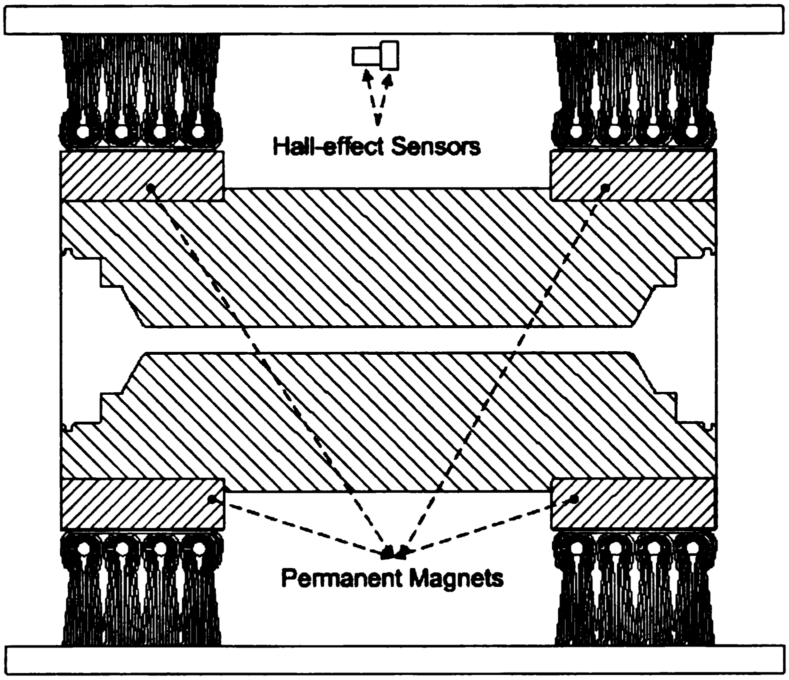


Figure 2.5. MFL inspection schematic.

As a general practice, pipe walls are magnetized to saturation using permanent magnets. When there is a crack or metal loss in the pipe wall, magnetic flux leaks into the air surrounding it. This leakage flux is captured by Hall-effect sensors. Hall-effect sensors convert the flux into electric voltage signal as explained before. This signal, called as MFL signal is stored for further analysis. All three components (axial, radial and circumferential)

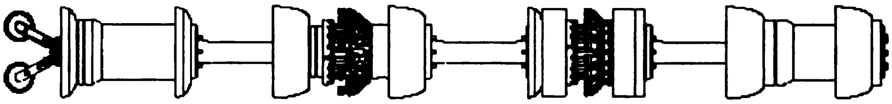


Figure 2.6. A schematic of the pig.

of the magnetic flux density can be measured using appropriate orientation of the Hall element. In order to reduce the data to be analyzed, most of the commercial tools measure only the axial component. The analysis performed in this thesis also deals with only the axial component of the MFL signal. The analysis and interpretation of MFL signal to assess integrity of the pipe involves following steps:

1. *Preprocessing of MFL signal*: The MFL signal is corrupted with noise due to various sources; seamless pipe noise (SPN) being the dominant source. This step involves denoising and compensation of MFL signal. MFL signal obtained from pipes with different wall thickness or different materials with different permeabilities can vary significantly. In order to have a uniform analysis system the MFL signal is compensated for variation in wall thickness and permeability of the pipe. Variation in velocity of the pig introduces some artifacts in the measured signal. These artifacts are removed using velocity compensation techniques.

2. *Segmentation of the defect areas*: This step identifies the regions having potential defects in MFL signal. These regions are called regions of interest or ROIs. Restricting all the further analysis to ROIs greatly reduces the amount of data to be analyzed.

3. *Solving the inverse problem*: Processing the MFL signal in the ROIs to solve the inverse problem of predicting the 3-dimensional depth profile of the defect in the pipe-wall.

4. *Prediction of maximum allowable operating pressure (MAOP)*: Using depth profile information along with finite element modeling (FEM) technique to find the MAOP

of gas that can be transmitted through the given pipe. The details of FE modeling are excluded from the scope of this dissertation.

2.4 Summary

The fundamental physical concepts underlying the magnetic flux leakage technique have been discussed in this chapter. The subsequent chapters continue with the discussion of the signal processing aspects in the interpretation of signal measured using the MFL assembly.

CHAPTER 3

Preprocessing and Defect Identification

3.1 Introduction

The raw MFL signal obtained from the pig is often corrupted with noise from various sources. As a first step in the processing of MFL signal, the regions of interest (ROIs) that contain the potential defects are identified. A ROI is defined as a rectangular region in two-dimensional MFL signal space that encloses one or more potential defects. Noise in the MFL signal makes identification of the ROIs difficult. In addition to noise the MFL signal is also affected by other factors such as:

1. pipe grade or wall-thickness
2. permeability of the pipe material
3. velocity of the pig during the inspection process.

It is essential to compensate the MFL signal in the ROIs to make it independent of these variations. A uniform analysis procedure can then be applied to the signal in the ROI.

The first half of this chapter discusses different approaches used to identify the ROIs in presence of noise, and the second half of the chapter discusses the compensation techniques used for making the signal in ROI invariant to the above mentioned factors.

3.2 Effect of Noise on MFL Signal

The raw MFL signal can be corrupted with noise from various sources. These include seamless pipe noise (SPN), lift-off variations and system noise due to on-board electronics and electromagnetic interference (EMI). The effect of SPN is generally stronger than the other two types of noise.

3.2.1 Seamless Pipe Noise (SPN)

Seamless pipes do not have welds along the axial direction and are manufactured using a special process involving a series of piercing, rolling and milling operations. The helical nature of these operations induce mechanical stresses in the pipe material. These stresses modify the magnetic properties of the material in such a way that a quasi-periodic signal, called SPN is superimposed on the raw MFL signal during the inspection of these pipes. Depending on the manufacturing of each pipe, the SPN level can vary. MFL signals with different levels of SPN are shown in Figures 3.1 and 3.2.

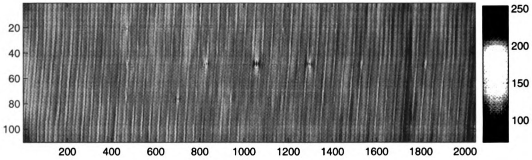


Figure 3.1. Sample raw MFL measurements with relatively high level of SPN.

The MFL signal shown in these figures is 2-dimensional. The horizontal direction is along the axis of the pipe and the vertical direction is along the circumference of the pipe. The signal in a row represents the signal collected by a single Hall-effect sensor as the pig

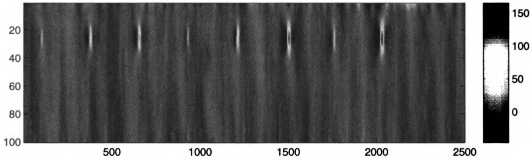


Figure 3.2. Sample raw MFL measurements with relatively low level of SPN.

moves along the axial direction and the signal in a column represents the signal obtained from all the sensors located along the circumference of the pipe when the pig is at same axial location. Thus the number of rows is equal to the number of sensors present on the instrument. As can be seen from Figures 3.1 and 3.2, the relative magnitude of SPN with respect to the MFL signal can vary substantially. This variation poses a challenge to develop a uniform systematic procedure to eliminate all types of SPN from given MFL data.

3.2.2 Lift-off variations

The internal surface of the pipe along which the MFL inspection assembly moves is not uniform. The wear plates, internal deposits and liners create non-uniformities; as a result, the distance between the Hall-Effect sensors and actual pipe surface fluctuates randomly. The presence of such lift-off has two effects:

1. *Base Level Shift*: Amount of air-coupled field increases due to lift-off. A higher base level is measured for a sensor with larger lift-off than for a sensor with smaller lift-off.
2. *Signal Level Shift*: There is a change in signal level at defects. The additional separation due to increasing liftoff decreases the amplitude and increases the length or duration of the signal.

Effect of lift-off can be reduced by making simple changes in the magnetizer assembly. Introduction of the brush system to reduce magnetizer stand-off is one such change. Brush system magnetically couples the flux to the pipe. Brushes also serve to soften impact of the magnetizer and sensors on pipeline intrusions such as welds, valves, branch connections, etc. At low inspection speeds, the brush length could be reduced by at least the lift-off to reduce the air-coupled field and increase the magnetization level in the pipe.

3.2.3 Proposed Schemes for Defect Detection

The two noise sources mentioned above produce non-stationary and time varying signals that corrupt the MFL signal. The objective is to detect and analyze the ROIs in this noisy signal. The first approach for detection of ROIs uses an adaptive filter to remove the noise. Adaptive Noise Cancelation (ANC) filter as proposed in [40] is used for this purpose. The tap weights of the filter are updated by means of least mean square (LMS) algorithm [1].

In the second approach, instead of filtering the MFL signal it is transformed into higher order statistics domain using higher order statistics transformation (HOST) to detect the ROIs in presence of noise. The statistics based approach is a new contribution presented in this dissertation [21] and it provides distinct advantages over the approach based on adaptive filtering. Following sections describe each algorithm in detail followed by a comparison of results on experimental data.

3.3 Adaptive Filtering for removal of SPN

3.3.1 LMS and NLMS Algorithms

The adaptive noise cancelation (ANC) filter was first developed by Widrow [40], [33] as an application of the fundamental adaptive filter. The ANC filter assumes the availability of a reference noise source that is highly correlated with the noise corrupting the information bearing signal. The architecture of an ANC filter is shown in Figure 3.3. $\mathbf{b}(k)$ is the reference noise source, $\mathbf{s}(k)$ is the information bearing signal that is corrupted with noise $\mathbf{n}(k)$. $\mathbf{d}(k) = \mathbf{s}(k) + \mathbf{n}(k)$ is the MFL signal to be filtered. Thus we have,

$$\mathbf{b}(k) = \mathbf{n}_0(k) \tag{3.1}$$

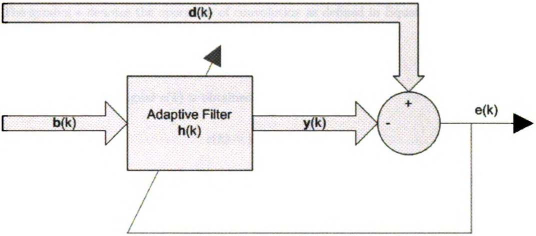


Figure 3.3. Architecture of adaptive noise cancelation (ANC) filter.

and

$$\mathbf{d}(k) = \mathrm{s}(k) + \mathbf{n}(k) \tag{3.2}$$

All the signals are vectors of length l; l being length of the filter. Parameter k represents the iteration number. It is assumed that the noise in $\mathbf{d}(k)$ (denoted as $\mathbf{n}(k)$) is correlated with reference noise source $\mathbf{n}_0(k)$.

$$\rho\{\mathbf{n}(k), \mathbf{n}_0(k)\} \approx 1 \tag{3.3}$$

The symbol ρ denotes correlation coefficient between $\mathbf{n}(k)$ and $\mathbf{n}_0(k)$. It is also assumed that the noise signals and the information bearing signals are uncorrelated.

$$\rho\{\mathbf{s}(k), \mathbf{n}_0(k)\} \approx 0 \tag{3.4}$$

$$\rho\{\mathrm{s}(k), \mathbf{n}(k)\} \approx 0 \tag{3.5}$$

The adaptive filter $\mathbf{h}(k)$ produces output $\mathbf{y}(k)$ using convolution operation in the time domain.

$$\mathbf{y}(k) = \mathbf{b}(k) * \mathbf{h}(k) \tag{3.6}$$

The symbol * denotes the operation of convolution as defined in Equation 3.7.

$$\mathbf{y}(k) = \sum_{i=0}^{i=l-1} \mathbf{b}(i)\mathbf{h}(k-i) \tag{3.7}$$

The scalar error signal $e(k)$ is obtained as,

$$e(k) = \|\mathbf{d}(k) - \mathbf{y}(k)\| \tag{3.8}$$

The weights or coefficients of the filter $\mathbf{h}(k)$ are updated by means of the least mean square (LMS) algorithm. The objective is to minimize the mean square error, $E[|e(k)|^2]$. Using Equation 3.8, this translates to the minimization of $E[\|\mathbf{n}(k) - \mathbf{y}(k)\|^2]$; see [1]. The filter $\mathbf{h}(k)$ is adapted to model $\mathbf{n}(k)$ using the reference noise source $\mathbf{n}_0(k)$. The adaptive LMS filter is preferred over an optimal Wiener filter, where the statistics of the signal is not known *a priori*. In the long run the LMS filter produces near optimal solutions and resembles a Wiener filter [1,9] if the given signal is wide sense stationary. It is asserted that in most practical situations the LMS filter can outperform the Wiener filter [17] due to its flexibility in adapting to the local changes in the data. However, to ensure the stability of the LMS filter the input data must be at least wide sense stationarity (WSS). The update equation for LMS filter is given in Equation 3.9.

$$\mathbf{h}(k+1) = \mathbf{h}(k) + \mu\mathbf{b}(k)e(k) \tag{3.9}$$

The step size, also called as learning rate, μ, plays a crucial role in the stability of the filter. Theoretically, the learning rate should be chosen based on the highest eigenvalue ($\lambda_{\max}$) of the covariance matrix of the input data [17], [18] as,

$$\mu \leq \frac{1}{2\lambda_{\max}} \tag{3.10}$$

The LMS filter is sensitive to rapid changes in the input signal $\mathbf{b}(k)$ and can become unstable if the data loses wide sense stationarity. In order to improve the performance of LMS filter, a normalized input signal is used for updating the weights. The resulting filter is called a normalized LMS filter or NLMS filter. The update equation of NLMS filter is given in Equation 3.11.

$$\mathbf{h}(k+1) = \mathbf{h}(k) + \frac{\mu}{(l+1)\sigma^2}\mathbf{b}(k)e(k) \tag{3.11}$$

The parameter σ^2 denotes the signal power of $\mathbf{d}(k)$ as defined in Equation 3.12.

$$\sigma^2 = \sum_{i=1}^{i=l}[d(i)]^2 \tag{3.12}$$

A NLMS filter has been experimentally verified to be more stable than the simple LMS filter. As all adaptive filters are highly data dependent, their stability cannot be guaranteed in general. It is always possible to encounter a signal where the given adaptive filter becomes unstable. In the reference [18] various approaches are presented to model the long term behavior and stability of these filters. In most practical situations, the statistics of the input data is not known beforehand and learning rate is determined empirically. With higher rate of learning, the filter converges faster but it becomes more unstable. Hence there is a trade-off between convergence rate and stability. The length of the filter also plays a role in determining the stability of the filter. Longer length filter are typically slow to converge but are more stable and vice versa.

3.3.2 PNLMS Algorithm

Recently a new method of updating the weights using LMS architecture has been proposed; proportionate NLMS or PNLMS [18]. In LMS or NLMS methods, the increment in the weights is made proportional to the input (without or with normalization) and the scalar error. However the previous values of the weights do not participate in quantifying their updates. In such cases, conceptually, energy contained in the error $e(k)$ is distributed to different weights of the filter regardless of their previous values. Ideally, weights having larger values must be incremented with a larger step size than weights having lower values in order to have the optimal distribution of energy. The PNLMS paradigm addresses this issue. The update equations for PNLMS are given in Equations 3.13, 3.14 and 3.15.

$$g_i = \max\{\rho \max\{\delta, |h_1(k)|, ..., |h_l(k)|\}, |h_i(k)|\} \tag{3.13}$$

$$\mu(k) = \frac{\mu\mathbf{G}(k)}{\mathbf{b}'(k)\mathbf{G}(k)\mathbf{b}(k) + \alpha} \tag{3.14}$$

$$\mathbf{h}(k+1) = \mathbf{h}(k) + \mu(k)\mathbf{b}(k)e(k) \tag{3.15}$$

The matrix $\mathbf{G}(k)$ is a diagonal matrix and the diagonal elements are given in Equation 3.13. The parameters ρ and δ are small positive real numbers employed to ensure that the diagonal elements of G do not become zero. The parameter α is also a small real positive number added to denominator to avoid division by zero. The parameters are chosen so that $0 < \alpha, \rho, \delta < 1$ and $h_1(k), ..., h_l(k)$ denote the coefficients of the filter in iteration number k. The further details about PNLMS algorithm and its variations can be found in [18]. Basic implementation of PNLMS algorithm is used in this thesis. Due to the matrix computation required in each step of the update, the PNLMS algorithm is extremely slow in execution compared to the other LMS algorithms.

3.3.3 Applying ANC for SPN removal

As can be seen from Figures 3.1 and 3.2, SPN shows some sort of periodicity along each row. As this periodicity is not well defined, SPN is said to be quasi-periodic. In order to apply LMS adaptive filters for removal of SPN [1], the input signals $\mathbf{d}(k)$ and $\mathbf{b}(k)$ need to be identified. Each row in the two dimensional image represents the MFL signal reported by a sensor as pig moves along axial direction. It is observed that SPN in the neighboring rows of MFL signal show a large correlation. For canceling the SPN in a given row ($\mathbf{d}_k$) one of the neighboring rows can be used as reference noise signal ($\mathbf{b}_k$). As the MFL signal is obtained from a cylindrical pipe, the first and last rows in the MFL signal are actually neighbors of each other. Thus each row of the signal can be associated with a neighboring row for obtaining the reference noise source in a uniform cyclic manner. If the current row of the MFL signal is n, then the reference noise signal is obtained from row $n \pm 2$. The signal in each row is processed sequentially to obtain the filtered signal.

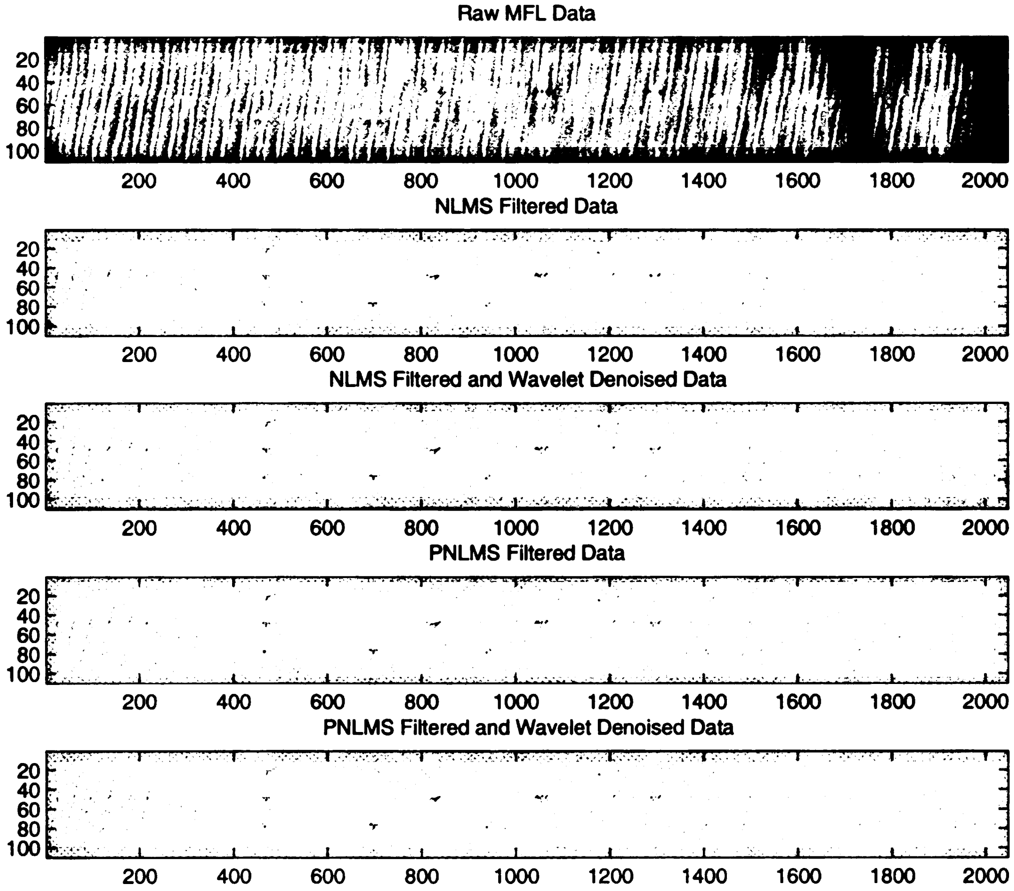


Figure 3.4. Results obtained by application of NLMS and PNLMS filters followed by wavelet shrinkage denoising on raw MFL signal with high level of SPN.

It is interesting to note that filtered signal corresponding to the defect region is substantially different than the original signal; see Figure 3.5.

Unfiltered MFL signal corresponding to defect **Filtered MFL signal corresponding to defect**

Figure 3.5. Difference between the PNLMS filtered and unfiltered signal in the defect region.

This distortion makes the filtered signal useless for the further processing. However, the filtered signal has relatively higher signal to noise ratio (SNR) compared to the original signal and hence thresholds applied to filtered signal generally give better segmentation results.

The wavelet shrinkage denoising technique using a soft threshold is used to clean the filtered signal [1]. The denoised signal is then subjected to a simple magnitude based threshold to obtain the segmented MFL signal. The segmented MFL signal contains the locations of the ROIs. The unfiltered MFL signal corresponding to the ROIs is used for further processing. Thus it is important to note that although ANC filtering is used here, the signal in the ROI is not filtered and is still corrupted with SPN. This noise is taken into consideration while developing the regularization schemes used in the inversion algorithms described in the following chapters.

The NLMS and PNLMS algorithms are applied to a raw MFL signal from Figure 3.1 and the results are shown in Figure 3.4. The performance of PNLMS filter appears be marginally better than NLMS filter at the cost of higher computation time. Also a PNLMS filter is more stable than a NLMS filter in most cases and is preferred over a NLMS filter. An example demonstrating superior stability of a PNLMS filter compared to a NLMS filter is shown in Figure 3.6.

3.4 Filter using Higher Order Statistics Transformation (HOST)

A novel approach is proposed that uses higher order statistics of the MFL signal for segmentation. The assumption of wide sense stationarity is applicable to the signal due to SPN. However, the MFL signal at the defect region is highly non-stationary and has higher amplitude than the defect free neighborhood. This non-stationarity is reflected as sudden change in the higher order statistics of the signals in an ROI. This change although visible in the third and forth order statistics (skewness and kurtosis respectively), is not easy to

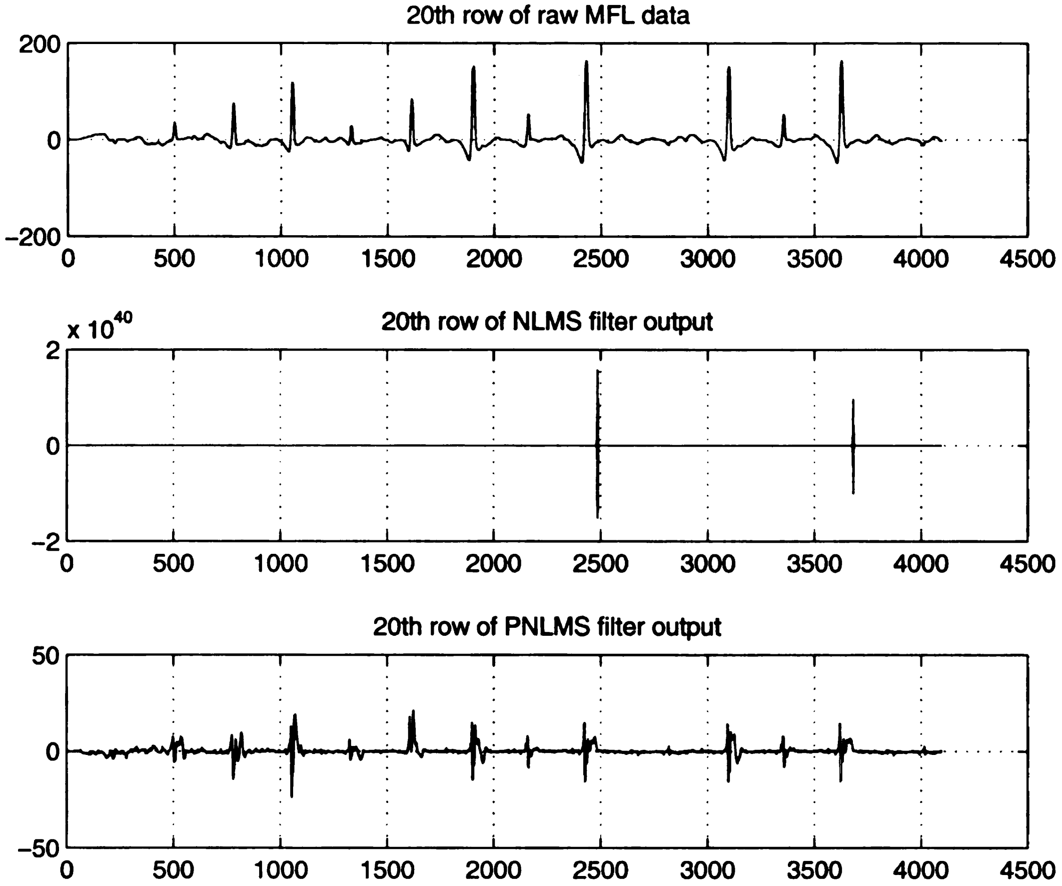


Figure 3.6. An example where NLMS filter becomes unstable while a PNLMS filter remains stable. 20^{th} row of the signal in Figure 3.2 is shown.

differentiate using first and second order statistic (mean and variance respectively) that is typically used for thresholding. It is observed that the skewness (depends of the left or right peaked-ness of the signal) and kurtosis (depends on overall peaked-ness of the signal as opposed to flatness of the signal) of the signal change rapidly in the local neighborhood of the defect region. These statistics are not affected by the relatively stationary SPN. The proposed algorithm makes use of this observation. The higher order statistics based filtering procedure involves three steps: (1) Higher order statistics transformation, (2) Filtering in the transformed domain and (3) Recovery of the filtered Signal.

3.4.1 Higher Order Statistics Transformation

The matrix of original MFL signal $\mathbf{D}$ is expressed as $\mathbf{D} = \mathbf{F} + \mathbf{N}$, where $\mathbf{F}$ is the non-stationary MFL signal corresponding to the defects and $\mathbf{N}$ is the noise component due to SPN. $\mathbf{N}$ is observed to be wide sense stationary. The skewness ($\varpi(\mathbf{x})$) and kurtosis ($\kappa(\mathbf{x})$) operations are specified using a one dimensional moving window $\mathbf{w}_{ij}$ of length l. Length of $\mathbf{w}_{ij}$ is chosen to be greater than the approximate period of the SPN. Elements of the moving window $\mathbf{w}_{ij}$ are defined as,

$$\mathbf{w}_{ij} = \mathbf{D}(i, j + k),, \text{ where } \alpha = 1, ..., l \text{ and } i = 1, ..., m \text{ and } j = 1, ..., n - l + 1 \tag{3.16}$$

The skewness and kurtosis of the data in window $\mathbf{w}_{ij}$ are defined as,

$$\varpi(\mathbf{x}) = \frac{\sum_{i=1}^{i=l}(x_i - \bar{x})^3}{(l-1)\sigma^3} \tag{3.17}$$

$$\kappa(\mathbf{x}) = \frac{\sum_{i=1}^{i=l}(x_i - \bar{x})^4}{(l-1)\sigma^4} - 3 \tag{3.18}$$

where σ is the variance of $\mathbf{x}$ and $\bar{x}$ is the mean of $\mathbf{x}$.

The signal $\mathbf{D}$ of dimensions m x n in the spatial domain is transformed to the HOST domain as matrix $\mathbf{\Omega}$ of dimensions m x $(n - l + 1)$ using the moving window $\mathbf{w}_{ij}$ as given by,

$$\mathbf{\Omega}(i, j) = \varpi(\mathbf{w}_{ij})\kappa(\mathbf{w}_{ij}), \text{ where } i = 1, ..., m \text{ and } j = 1, ..., n - l + 1 \tag{3.19}$$

3.4.2 Filtering in the Transformed Domain

The transformed domain signal is denoted as $\mathbf{\Omega}$. A threshold ρ is applied to $\mathbf{\Omega}$ to obtain $\mathbf{\Omega}'$ according to Equation 3.21. The threshold ρ is obtained using global mean γ and global variance ν of $\mathbf{\Omega}$ as,

$$\rho = \gamma + \beta\nu \tag{3.20}$$

Value of β is empirical and depends on noise. Typical values are 0.2 for low noise levels and 2 for high noise levels.

$$\boldsymbol{\Omega}'(i,j) = \begin{cases} 1, & \boldsymbol{\Omega}(i,j) > \rho; \\ 0, & \boldsymbol{\Omega}(i,j) \leq \rho; \end{cases}, \quad \{i = 1, ..., m; j = 1, ..., (n - l + 1)\} \tag{3.21}$$

The regions of interest (ROIs) are identified from $\boldsymbol{\Omega}'$ using a voting process as given below:

1. Initialize a template matrix $\mathbf{M}$, $\mathbf{M}(i,j) = 0; i = 1, ..., m$ and $j = 1, ..., n$ $r = 1$ and $c = 1$.

2. Update the template matrix as follows:
   If $\boldsymbol{\Omega}' = 1$, $\mathbf{M}(r,c) = \mathbf{M}(r,c) + 1$ and $\mathbf{M}(r, c + l - 1) = \mathbf{M}(r, c + l - 1) + 1$. In this process, each signal point in the HOST domain that lies above the threshold votes for two points in spatial domain. Each of the two points lie in the same row at a distance equal to the length of the filter. As each point in spatial domain can contribute a maximum of two points in the HOST domain, the maximum value of any point in the template matrix is 2.

3. Increment $c = c + 1$.

4. If $c > n - l + 1$, $c = 1$ and $r = r + 1$, else go to step 2.

5. If $r > m$ go to step 6, else go to step 2.

6. Stop.

3.4.3 Recovering the Filtered Signal

The template matrix $\mathbf{M}$ contains pixels with values in $\{0, 1, 2\}$. The cells in matrix $\mathbf{M}$ having value 2 are the pivotal points in the MFL signal where the statistics of the signal changes rapidly. These points represent the centers of the defects. The cells in $\mathbf{D}$ corresponding to the cells in $\mathbf{M}$ having the value 2 are retained to get the final filtered signal $\hat{\mathbf{F}}$, which is an estimate of **F**. Figure 3.7 shows output at individual steps of the HOST based filtering on a sample MFL signal. The threshold value is selected as described in the algorithm and is kept constant for all the available data.

As seen from Figure 3.7, the final filtered signal directly provides the segmented defect regions. The regions are narrow and concentrated at the center of the defect and do not include the entire defect signature. These boxed regions are then expanded to capture entire defect signal. The expanded regions are standardized to a box with constant dimensions using zero padding from all the four sides.

3.4.4 Post-processing Filtered Signal

The filtered signal shown in Figure 3.7 shows some spurious indications that do not enclose a defect. These indications are removed using variance based thresholding. Figure 3.8 shows the result of this post processing.

3.4.5 Generalization of HOST Filter

HOST filtering algorithm described above is applied along the rows of the MFL signal. For reference, denote this procedure as HOST-r. The same algorithm can also be applied along the columns. The filtered signal is post processed as before. This procedure is labeled as HOST-c. The results of using HOST-c are shown in Figures 3.9 and 3.10. Figure 3.9 shows the result before post processing and Figure 3.10 shows the result after post processing. The HOST-r process identifies more defects than HOST-c. However, the last defect in the second row of defects is detected with HOST-c that is missed with HOST-r, see Figures 3.8 and 3.10. Thus the HOST-r process does not perform better than HOST-c in all the situations. In order to achieve optimal detection performance, both the processes are independently applied and the results are merged using simple OR operation.

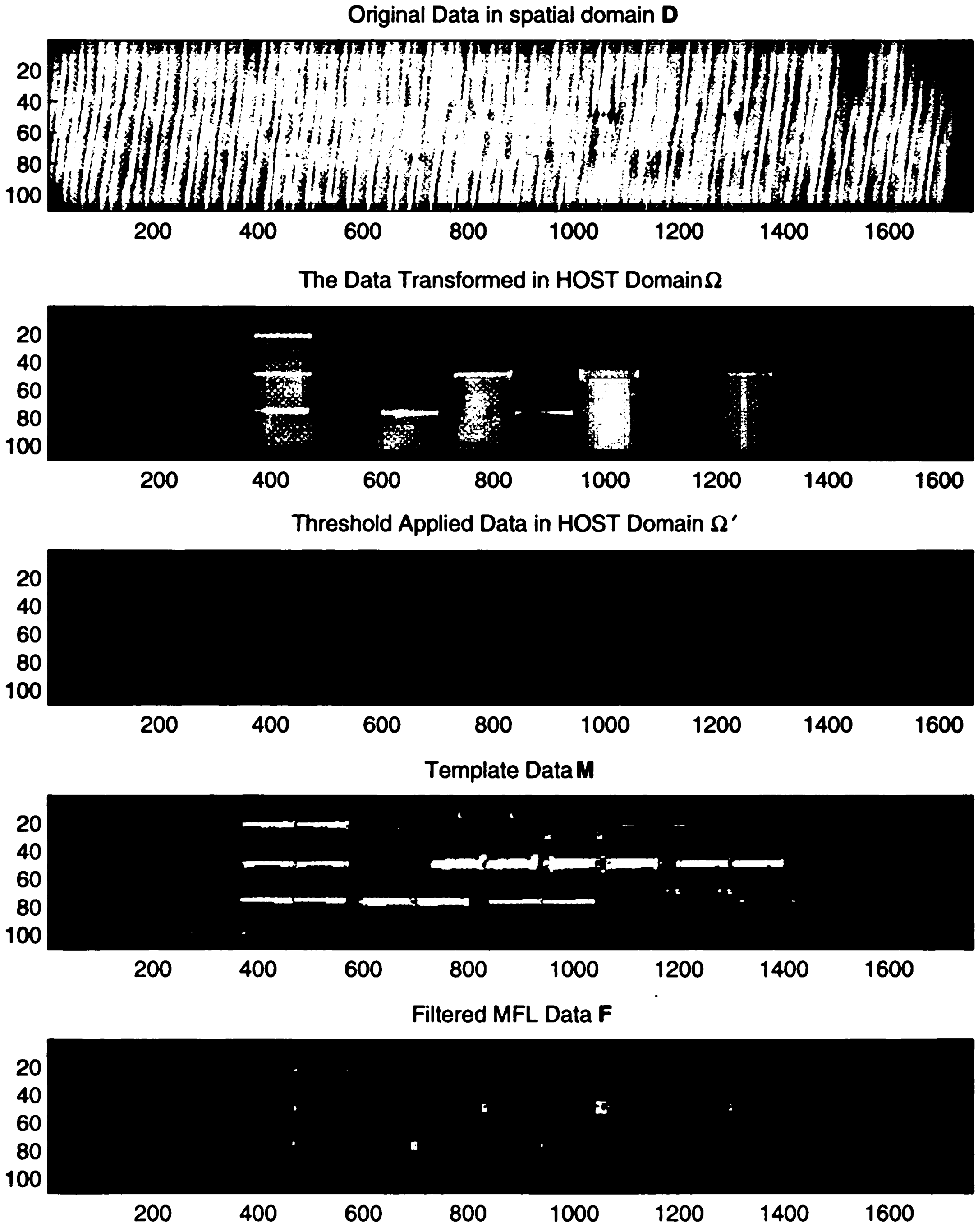


Figure 3.7. Intermediate steps in the processing of HOST based filtering.

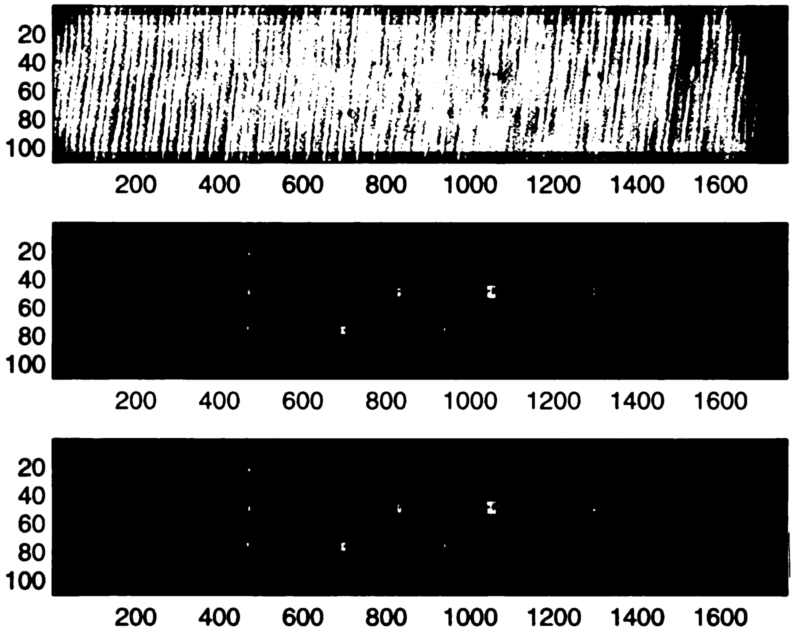
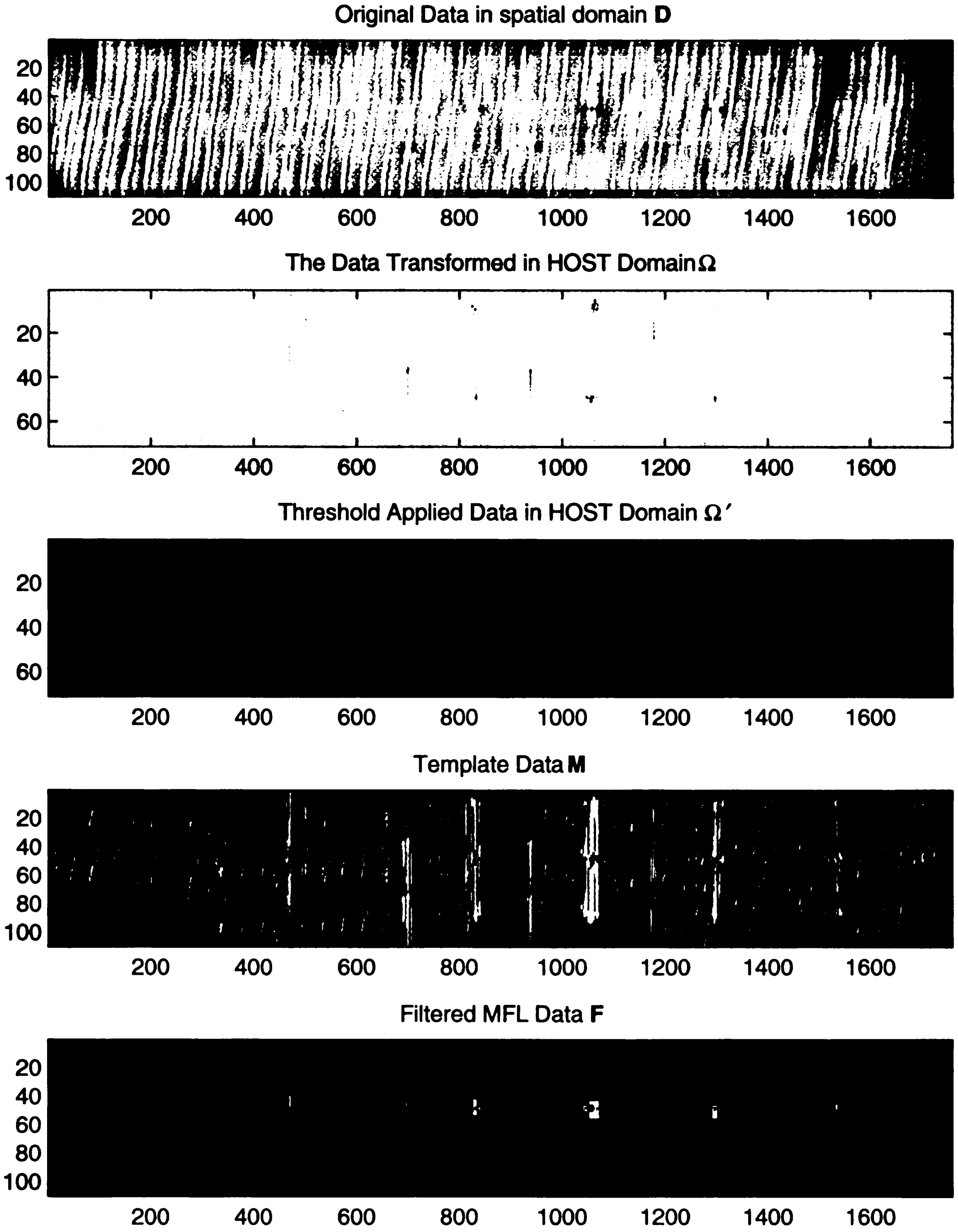


Figure 3.8. Post processing of the HOST filtered signal.

3.5 Comparison of ANC and HOST filters

The ANC filters with learning algorithm of LMS, NLMS or PNLMS are based on the assumption that the input signal is wide sense stationary. Although this assumption is satisfied by the noise signal, the signal corresponding to a defect region is non-stationary and shows rapid changes in the amplitude. Thus strictly speaking, the stability of an ANC filter cannot be guaranteed. Consequently they cannot be used in developing fully automated analysis system. Adaptive filtering is essentially a sequential process and cannot exploit the advantages of parallel processing architecture (typically used in high-speed computation).

HOST filtering approaches (HOST-r and HOST-c) are based on the non-stationarity of MFL signal in the defect region. Stability of the algorithm can be assured due to its non-adaptive and data independent procedure. The algorithm can be made arbitrarily fast by executing it on a parallel processing machine; the process of higher order statistic trans-



Original Data in spatial domain **D**

The Data Transformed in HOST Domain Ω

Threshold Applied Data in HOST Domain Ω'

Template Data **M**

Filtered MFL Data **F**

Figure 3.9. Intermediate steps in the processing of HOST based filtering along columns.

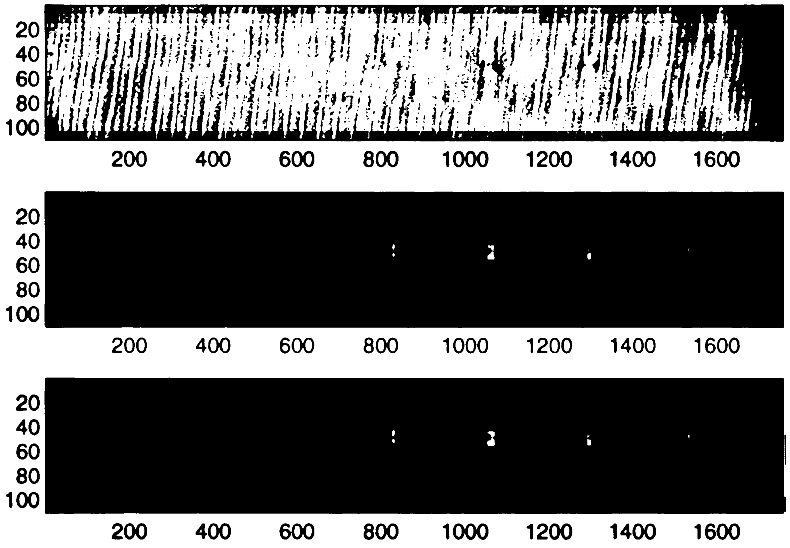


Figure 3.10. Post processing of the HOST filtered signal applied along columns.

formation can be executed in parallel for different rows in HOST-r and different columns in HOST-c.

The performance criterion used for comparing ANC filters and HOST filter is the signal to noise ratio before applying threshold. With this criterion the HOST filters outperform ANC filter by a considerable margin as seen in Table 3.1. The *max signal level* represents the maximum signal level of the least detectable defect and *max noise level* represents the maximum signal level outside all the defect regions.

Table 3.1. Quantitative comparison of the performance of ANC and HOST filters.

	Max Signal Level	Max Noise Level	SNR
NLMS	39.76	30.34	1.3103
PNLMS	34.49	19.75	1.74
HOST	39.70	3.5134	11.3018

The results described in Table 3.1 for HOST filter refer to the use of HOST-r filter alone.

All the methods of defect identification discussed above suffer from a common drawback. None of the methods is capable of removing the SPN content in the MFL signal corresponding to the defect region. As the defect region has higher magnitude than the average magnitude of SPN in the local neighborhood, the signal still possesses the discriminating capabilities for further processing. This noise is accounted for during the inversion algorithms by using regularization scheme.

Typically in an inspection the volume of the data collected over the hundreds of miles of pipeline can be very large (hundreds of Gigabytes). The segmentation step vastly reduces the amount of data that is analyzed subsequently. Typical reduction in the data is of the order of 1000.

3.6 MFL Signal Compensation Techniques

Preceding section described the defect segmentation techniques using various methods. The segmented MFL signal needs to be processed further before applying the inversion algorithms for predicting the 3-dimensional depth profile of the defects. This processing is commonly referred to as signal compensation. The compensation is required to make the MFL signal invariant to changes in following parameters: (1) Velocity of the pig and (2) Change in the permeability or wall thickness of the pipe material. Changes in permeability of the material and wall thickness affect the signal in the same way and hence a uniform compensation technique can be used for eliminating the effects of both parameters. The following sections describe the compensation techniques in detail.

3.6.1 Velocity Compensation

Basic Theory of Velocity Effects on MFL signal

In the theory of electromagnetics, induced currents in a conductor are governed by two laws: Lenz' law and Lorentz' law [20]. Lenz' law explains the generation of opposing magnetic fields with a macro-perspective view while Lorentz' law provides a micro-perspective view of the same by examining electron interactions in a magnetic field.

Lenz's law states that velocity of a moving magnet induces a magnetic field that opposes the direction of the inducing magnetic field. In the context of MFL inspection, the moving tool generates currents in the pipe that produce a magnetic field that reduces the strength of the applied magnetic field.

Lorentz' Law provides information about the coupling between mechanical motion and the magnetic field. It states that a charge moving perpendicular to a static magnetic field experiences a force, called as Lorentz force, perpendicular to both the velocity and the field. The direction of the Lorentz force is given using the right hand thumb rule. Although this law is more commonly used to describe the deflection of an electron beam in a cathode ray tube, Lorentz' law also applies to conducting material moving relative to a perpendicular static magnetic field. The magnetizing tool moving down the pipeline is such a case. Here, the Lorentz force induces a current density within the conducting material. The velocity induced current density, $\mathbf{J}$, the velocity of the conductor relative to the field $\mathbf{v}_c$, and the magnetic flux density $\mathbf{B}$, are related as,

$$\mathbf{J} = \sigma(\mathbf{v}_c \times \mathbf{B}) \tag{3.22}$$

where σ is the conductivity of the material and "$\times$" is the cross product operator. Due to the cross product operation only the orthogonal components of $\mathbf{v}_c$ and $\mathbf{B}$ are responsible for inducing the current density, $\mathbf{J}$.

When $\mathbf{v}_c$ and $\mathbf{B}$ are in the same direction, there is no induced current due to velocity, and when $\mathbf{v}_c$ and $\mathbf{B}$ are perpendicular to each other the maximum current is induced. This theory was formulated to explain generation of electricity in the moving conducting wire in a stationary field. According to the convention, $\mathbf{v}_c$ is the velocity of the conducting material relative to the field.

Applying Electromagnetic Theory to Pipelines

The direction of motion of pig and the field direction must be orthogonal to each other in order to generate current that will affect the inspection results. Using a cylindrical coordi-

nate system to describe the magnetizer moving through a pipe, the direction of magnetizer motion $-\mathbf{v}_c$ is always in the axial direction. So, a magnetic field in radial direction $\mathbf{B}_r$ or circumferential direction $\mathbf{B}_\phi$ is needed to induce currents. The magnetic field in axial direction is denoted as $\mathbf{B}_z$.

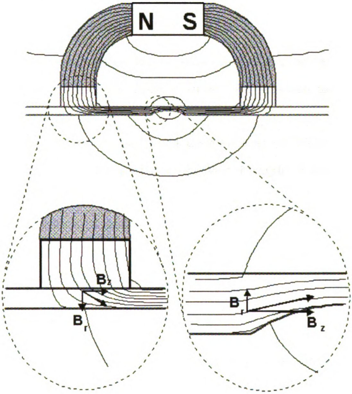


Figure 3.11. Velocity effects on MFL signal.

Magnetic field in radial direction, $\mathbf{B}_r$, produces an induced current in the circumferential direction, $\mathbf{J}_\phi$ as shown in Figure 3.11. The pipe acts as a loop to complete the current path for circumferentially flowing current. Magnetic field in circumferential direction, $\mathbf{B}_\phi$, produces a radial current, $\mathbf{J}_r$. Enclosed current path does not exist for current flowing in radial direction.

Radial and circumferential magnetic flux densities occur at two places: near defects and near the brushes, where the magnetic energy is coupled into the pipe. Near the point of contact, commonly referred to as the pole pieces of magnetizer, majority of the flux is radial and the induced current flows circumferentially around the pipe. Near defects, the flux leaks out both radially and circumferentially, and the induced current flows close to the defect.

Compensation of Velocity Effects

The compensation of velocity effects can be done using non-linear systems such as neural networks, or using the theory of linear systems. An approach using a Wiener filter deconvolution is presented here.

The compensation operation is performed along each row of 2D MFL signal as shown in 3.1 and 3.2. In each row a moving window operator of length l is applied for performing the deconvolution. Let the true MFL signal vector be denoted as $\mathbf{d}(n)$ and modified MFL signal due to velocity effects be denoted as $\mathbf{x}(n)$. The signal $\mathbf{x}(n)$ is the measured signal and $\mathbf{d}(n)$ is the true MFL signal that needs to be recovered. Let the velocity effects be represented in the form of a filter $\mathbf{g}(n)$. Hence we have,

$$\mathbf{x}(n) = \mathbf{d}(n) * \mathbf{g}(n) \tag{3.23}$$

where the operator $*$ represents convolution. The deconvolution problem can be formally stated as recovery of the signal $\mathbf{d}(n)$ by estimating an inverse filter $\mathbf{g}^{-1}(n)$ that restores the true MFL signal as,

$$\mathbf{d}(n) = \mathbf{x}(n) * \mathbf{g}^{-1}(n) \tag{3.24}$$

The convolution operation becomes multiplication and the deconvolution operation becomes division in the frequency domain. Thus, the frequency domain representation of the deconvolution can be written as,

$$\mathbf{D}(e^{j\omega}) = \frac{\mathbf{X}(e^{j\omega})}{\mathbf{G}(e^{j\omega})} \tag{3.25}$$

where $\mathbf{D}(e^{j\omega})$, $\mathbf{X}(e^{j\omega})$ and $\mathbf{G}(e^{j\omega})$ denote the frequency domain representations of $\mathbf{d}(n)$, $\mathbf{x}(n)$ and $\mathbf{g}(n)$ respectively. In practice, $\mathbf{x}(n)$ is also corrupted with some additive noise

$\mathbf{w}(n)$, which is not correlated with signal $\mathbf{d}(n)$, and the expression for $\mathbf{x}(n)$ can be written as,

$$\mathbf{x}(n) = \mathbf{d}(n) * \mathbf{g}(n) + \mathbf{w}(n) \tag{3.26}$$

The Wiener filter produces minimum mean-square error estimate (MSE) of $\mathbf{d}(n)$ from $\mathbf{x}(n)$. Let $\mathbf{h}(n)$ be IIR linear shift-invariant filter and let $\hat{\mathbf{d}}(n)$ be the estimate of $\mathbf{d}(n)$ that is produced by filtering $\mathbf{x}(n)$ with $\mathbf{h}(n)$.

$$\hat{\mathbf{d}}(n) = \mathbf{x}(n) * \mathbf{h}(n) = \sum_{l=-\infty}^{\infty} \mathbf{h}(l)\mathbf{x}(n-l) \tag{3.27}$$

The filter coefficients $\mathbf{h}(n)$ minimize the mean-square error defined as,

$$\varepsilon = E\{\|\mathbf{d}(n) - \hat{\mathbf{d}}(n)\|^2\} \tag{3.28}$$

$\mathbf{h}(n)$ provide the solution to the Wiener-Hopf equation, $\mathbf{R}_x\mathbf{w} = \mathbf{r}_{dx}$, where $\mathbf{R}_x$ is a Hermitian Toeplitz matrix of autocorrelations. $\mathbf{w}$ represents the filter coefficients and $\mathbf{r}_{dx}$ represents cross-correlations between the desired signal $\mathbf{d}(n)$ and the observed signal $\mathbf{x}(n)$.

This method is used throughout this dissertation for compensating the MFL signal for changes in the velocity of the pig. Figure 3.12 is an illustration of where the MFL signal measured with velocities 3, 5 and 7 miles per hour are compensated to match the MFL signal measured at a reference velocity of 3 miles per hour.

3.6.2 Permeability and Wall-Thickness Compensation

Wall-thickness (WT) determines the amount of magnetic flux that can flow inside a pipe. Greater WT leads to a weaker leakage signal. This effect is similar to the increase in permeability of the pipe. In order to have a uniform analysis algorithm for different pipes (with different WTs and permeabilities), the MFL signal needs to be compensated for variations in WT and permeability. The idea is to transform the thicker and thinner WTs to a reference value of WT. As the effect of changes in permeability and WT are indistinguishable a uniform compensation scheme for WT is applied assuming all the pipes are of the same permeability. It is found that although the relation between the peak-to-peak (P2P) value

of MFL signal and WT is highly nonlinear, there exists a near linear relation between P2P and 1/WT as shown in Figure 3.13. The WTs are expressed in $\frac{1}{1000}^{\text{th}}$ of an inch. The variation of P2P versus 1/WT is plotted for defects with variable depths viz. 10%, 25%, 50% and 80%.

In general, for all the defects at various depths, the slope of P2P versus 1/WT plot is not very different. If a common universal slope 'm_{uni}' for all the defects and depths can be found, then various WTs can be mapped onto one reference (or normalized) WT. This algorithm works equally efficiently for variations in permeability. For more than 3 WTs, piece-wise linear approximation can be used.

The WT and permeability compensation algorithm works in two steps: (1) Estimate the WT of pipe for the given MFL signal if it is unknown. (2) Apply WT compensation as explained above.

Results of using WT compensation applied to MFL signal obtained from three pipes with different WTs are shown in Figures 3.14 and 3.15.

This procedure is used throughout this dissertation for preprocessing any MFL signal whenever not mentioned explicitly.

3.7 Concluding Remarks

Preprocessing of MFL signal and identification of ROIs are discussed in this chapter. Traditional methods that make use of adaptive noise cancelation (ANC) filters are discussed first. These methods are found to be unstable and divergent when applied to some MFL signals. There exists no theoretical bound that can ascertain the stability of these filters. These methods therefore cannot be used for developing fully automated analysis systems.

A higher order statistics transformation (HOST) based approach towards identification of ROIs is proposed. The algorithms developed with this approach make use of changes in skewness and kurtosis of the MFL signal to identify the ROIs. It incorporates a deterministic and data independent procedure and is stable for any given data. These new procedures,

termed as HOST-r and HOST-c, are found to provide results with higher signal to noise ratio at faster speeds compared with the ANC filter based ROI identification method.

Standard compensation techniques are described that make the MFL signal invariant to factors such as wall-thickness, permeability of the pipe material and the velocity of the pig.

The next chapter discusses the processing of the data in the ROIs to predict the details of the defect geometry using direct inversion method.

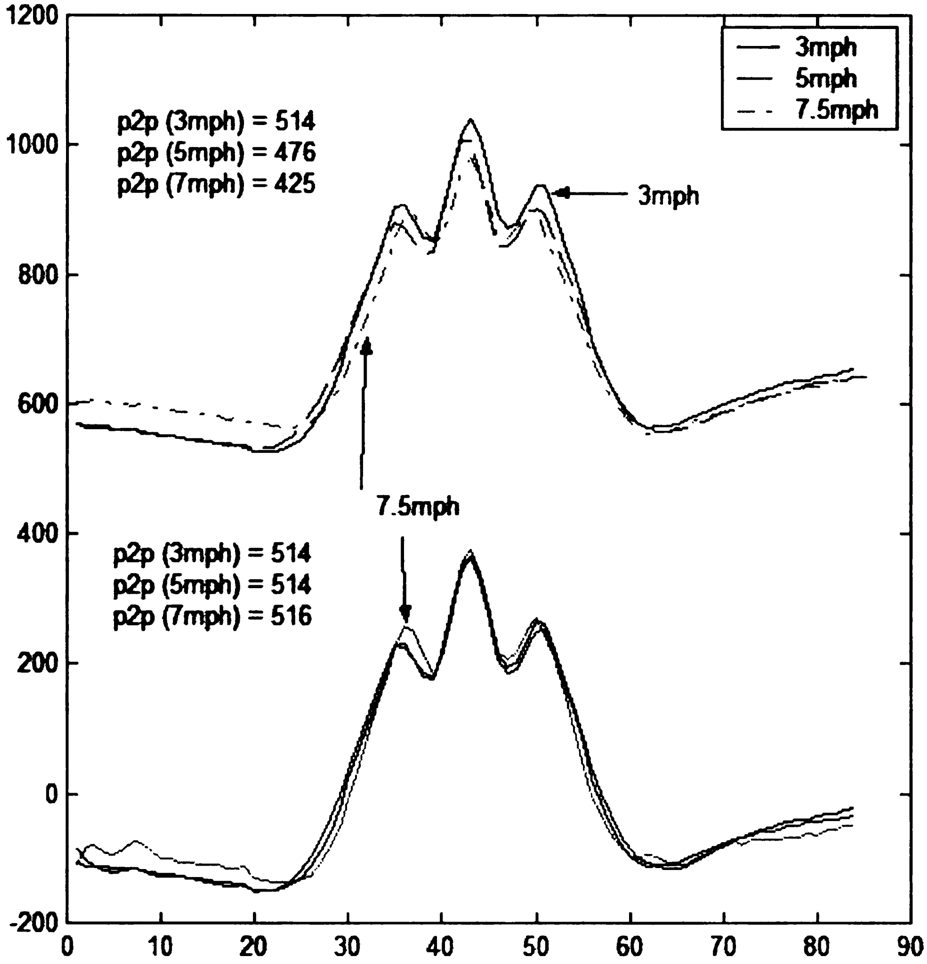


Figure 3.12. Illustration of velocity compensation.

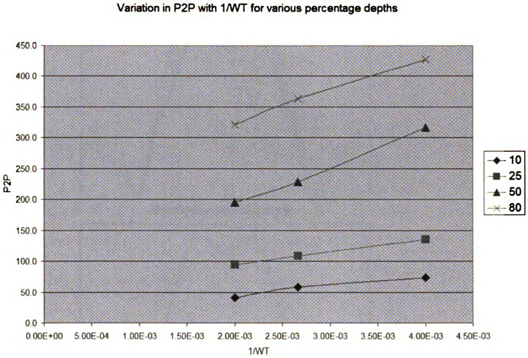
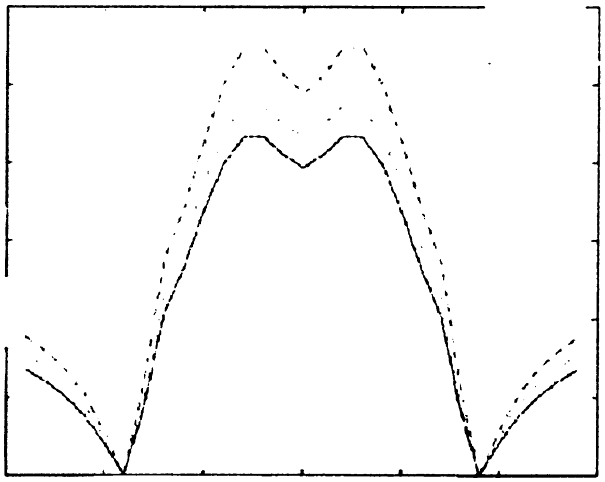
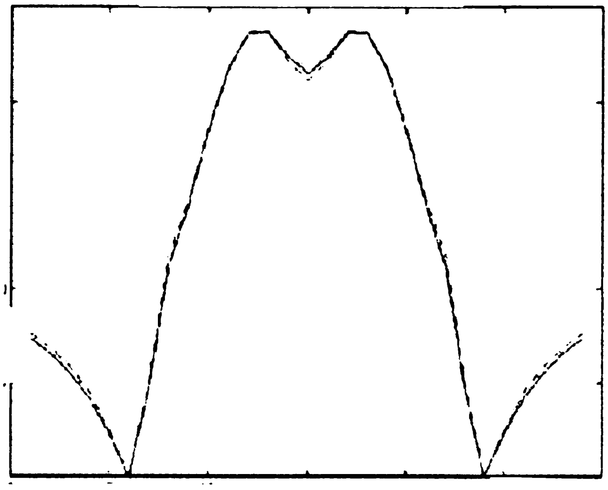


Figure 3.13. Variation of the P2P with 1/WT for different depth defects.



Before Compensation, Depth = 60



After Compensation, Depth = 60

Figure 3.14. Result of wall thickness compensation.

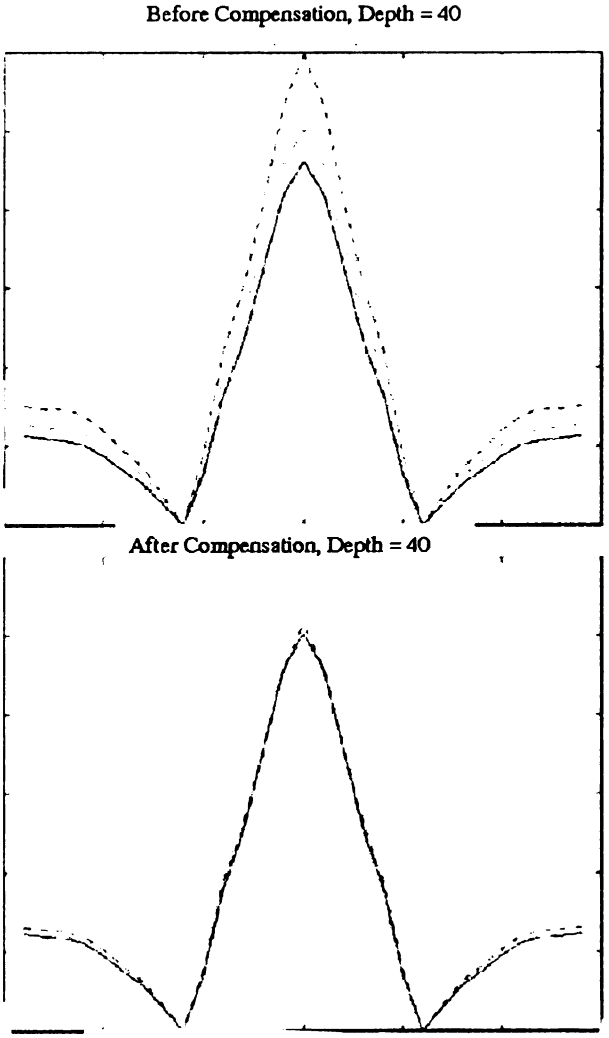


Figure 3.15. Another result of wall thickness compensation.

CHAPTER 4

Defect Profiling: Direct Inversion Techniques

4.1 Introduction

The segmentation techniques discussed in the previous chapter provide a 2-dimensional block of MFL signal corresponding to each region of interest (ROI) enclosing potential defects. Each ROI is of a different size depending on the corresponding defect size. Large and deep defects produce strong MFL signals that spread over a large area while the small or narrow defects produce weak MFL signals that spread over a relatively small area. These segmented ROIs are then processed to predict the depth profile.

A neural network based approach to predict the depth profile is discussed in this chapter. Examples of such an approach can be found in [32] and [7]. Before discussing the details of the inversion approach for predicting the depth profile from MFL data, the theory of neural networks is discussed in brief.

4.2 Background of Neural Networks

The theory of neural networks finds its roots in two different and apparently unconnected domains of science, viz. neurobiology and mathematics of function approximation. Neural networks have been used in the scientific community for over 50 years for solving a variety of problems ranging from multidimensional function approximation and signal classification to process control applications. Their structure is inspired from the structure of neurons in the human brain. A fully developed human brain possesses over 10-100 billion $(1000000000 - 10000000000)$ neurons [World Book 2001]. Each neuron is connected to about 10000 other neurons on an average, forming over 60 trillion (60000000000000) connections (also called synapses) among them [16]. No computer currently in production can match the complexity of this nature. The response time of silicon ship can be of the order of nanoseconds (10^{-9} seconds), while that of human neuron is about a millisecond (10^{-3} seconds). The development of artificial neural network (ANN) is an attempt to understand and exploit the functioning of the biological nervous system and combine this knowledge with higher speeds of silicon chips to obtain a machine capable of doing intelligent tasks.

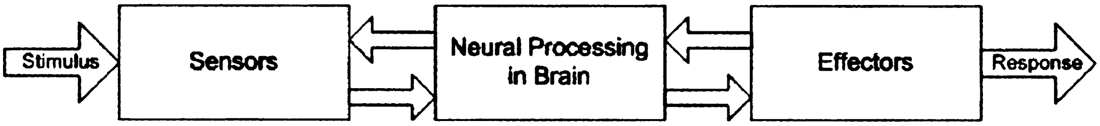


Figure 4.1. Block diagram representation of functioning of brain.

Figure 4.1 shows a block diagram of the functionality of a biological neural network. The sensory information (called stimuli) acquired by the sensors is transmitted to the brain in the form of electric pulses. The brain processes the information to produce an output in the form of electric pulses that is sent out to the effectors to produce the response. The neural processing done in the brain is based on the experiences in the past and innate behavior. Artificial neural networks are provided with memory to store past experiences and some basic processing rules that can process sensory information along with the stored experiences.

In order to understand the details of information processing in artificial neural networks, one needs to use the language of mathematics. The roots of neural networks from the mathematical domain are discussed in the following subsection.

4.2.1 Theory of Function Approximation

The theory of multivariate function approximation using univariate functions was pioneered by Kolmogorov [5] in 1957. According to Kolmogorov's theorem, any function of multiple variables can be represented in terms of multiple functions of single variable. Mathematically, for a function of n variables, the approximation using basis function expansion is given as,

$$f(x_1, x_2, ..., x_n) = \sum_{1}^{2n+1} h_j(\sum_{i}^{n} g_{ji}(x_i)) \tag{4.1}$$

The functions h_j and g_{ji} are continuous functions of single variable. The functions g_{ji} are also fixed monotonically increasing functions that are independent of $f(x_1, x_2, ..., x_n)$. Further simplifications to Equation 4.1 by Sprecher [35], resulted in the following equation,

$$f(x_1, x_2, ..., x_n) = \sum_{1}^{2n+1} h(\sum_{i}^{n} \lambda_i g_j(x_i)) \tag{4.2}$$

where all the different h_j are replaced with a single h and all the different g_{ji} are replaced with constants λ_i and functions g_j. With the help of Equation 4.2, one can develop a network with exactly 2 hidden layers for mapping any n-variate function using $2n + 2$ number of univariate functions (g_j, where $j = 1, ..., 2n + 1$ and h) and n number of constants (λ_i, where $i = 1, ..., n$). As a result of this proof a network exists with up to two hidden layers that can correctly map any scalar continuous multivariate function using multiple scalar univariate continuous functions and appropriate weight constants. This network can be easily extended for mapping vector functions.

Although these theorems prove the existence of functions capable of approximating any multivariate function $f : \Re^N \rightarrow \Re^M$, using multiple univariate functions, they do not provide any information on how these univariate functions can be found for a given multivariate function. It is the task of an engineer to find and optimize these functions for precise and compact representation of the given multivariate function. Neural networks are essentially parallel algorithms for estimating the parameters of the multi-dimensional approximation functions.

4.2.2 Various Realizations of Neural Networks

Based on this background Artificial Neural Networks (ANNs) are realized using various approaches. The two most commonly used approaches are [5]: (1) Multi-layer Perceptron (MLP) having one or more hidden layers and (2) Radial Basis Function Neural Network (RBFNN) with a single hidden layer.

The Multilayer Perceptron (MLP

The MLP typically uses sigmoid function as an activation function for all the nodes, while RBFNN typically uses a gaussian or other radially symmetric activation function. A typical ANN is shown in Figure 4.2. There is an input layer of dimensionality N followed by h hidden layers ($h \geq 1$). The dimensionality of each hidden layer can be arbitrarily defined. This dimensionality can even be higher than input or output dimensionality. The hidden layers are followed by a single output layer of dimensionality M. The number of nodes in the i^{th} hidden layer is P_i, where $1 \leq i \leq h$. The P_i are chosen using some empirical or *a priory* information about the problem at hand.

Radial Basis Function Neural Network

A RBFNN has a single hidden layer in addition to the input and output layers. The dimensionality of the hidden layer is less than or equal to the number of input-output samples and is independent of the input or output dimensionality. In spite of having only one hidden layer RBFNNs are capable of solving all the problems solvable with MLPs having multiple hidden layers [16]. MLPs develop a *global mapping* from input space to output space with the set of hidden and output layers. RBFNN, on the other hand, generates a

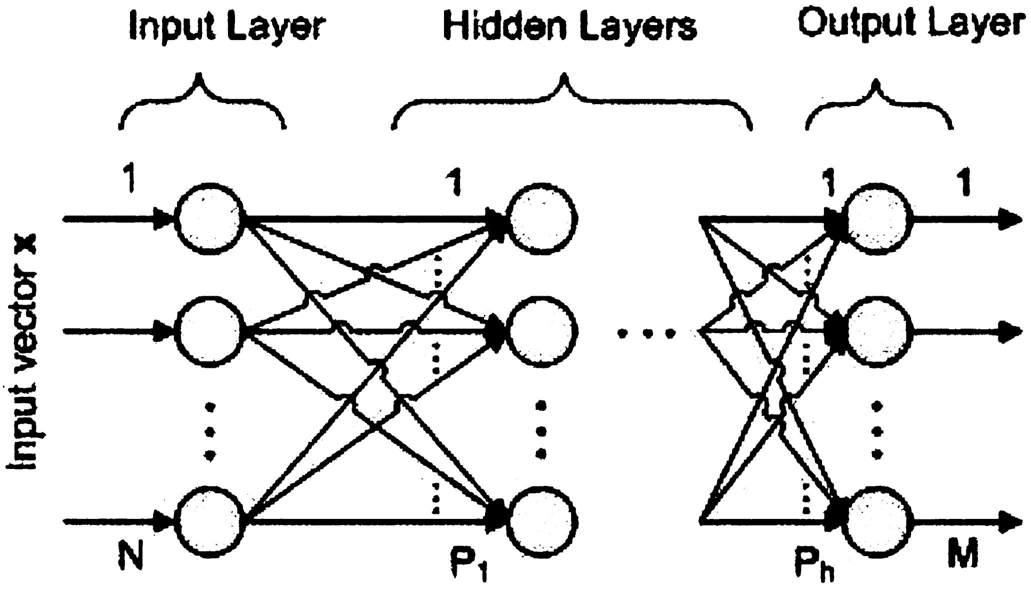


Figure 4.2. Architecture of Artificial Neural Network.

local mapping from input space to output space, with the help of nodes in the single hidden layer that represent the clusters in the input space. RBFNN offers a simpler representation for the ANN and can be better represented analytically. Due to these advantages RBFNN is chosen as the function approximator in this dissertation.

4.3 Architecture of RBFNN

The architecture of a radial basis function neural network (RBFNN) is shown in Figure 4.3. Mathematically, input-output relationship of a RBFNN is given in Equation 4.3.

$$\mathbf{y} = \sum_{i=1}^{i=P} \mathbf{w}_i f(\|\mathbf{x} - \mathbf{t}_i\|, \sigma_i) + \mathbf{w}_0 \tag{4.3}$$

where $\mathbf{x}$ is the input vector in $\Re^N$ and $\mathbf{y}$ is the output vector in $\Re^M$. The hidden layer of RBFNN consists of P centers of radial basis functions denoted as $\mathbf{t}_i, i = 1, ..., P$ and $\mathbf{t}_i \in \Re^N$. $f : \Re^N \rightarrow \Re$ is a scalar valued radial basis function and the scalar quantity $\sigma_i \in \Re$ is the spread or radius of the i^{th} center, $\mathbf{t}_i$. $\mathbf{w}_i$ is the weight vector corresponding to the i^{th}

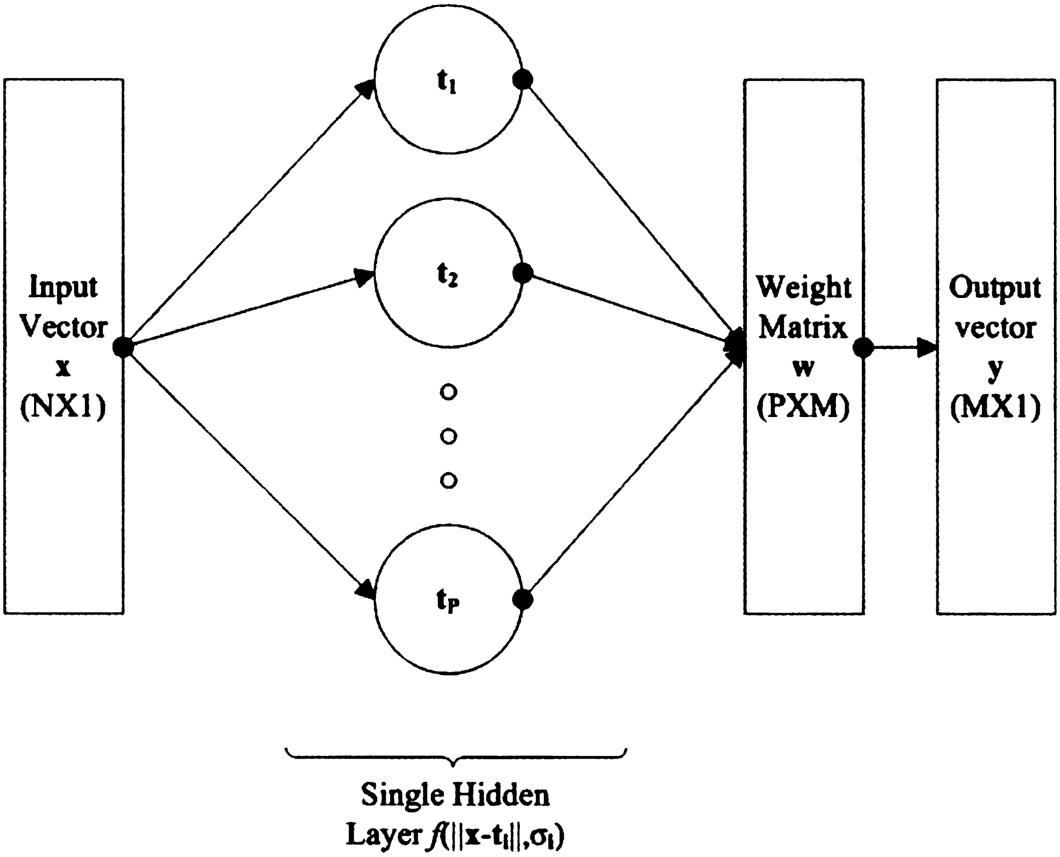


Figure 4.3. The basic architecture of RBFNN.

center and $\mathbf{w}_i \in \Re^M$. The same radial basis function is generally used for all the centers. Networks using different basis functions for different nodes are called hyper basis function neural networks as proposed in [30]. However, they are not widely used as they increase the complexity of the network without significant gain in the performance. More details about the RBFNN can be found in [16].

4.4 Training Parameters of RBFNN

During the training phase of RBFNN, the training data set $S_TR = \{(\mathbf{x}_i, \mathbf{y}_i)|i = 1, ..., n\}$ is used to estimate the parameters of the RBFNN collectively denoted as Θ. Θ is defined as, $\Theta = \{(\mathbf{w}_i, \mathbf{t}_i, \sigma_i) | i = 1, ..., n\}$. The mapping of RBFNN can be compactly represented using the notation, $\tilde{\mathbf{y}}_i = \tilde{f}(\mathbf{x}_i, \Theta)$. The objective of training is to minimize the squared error between the predicted values and true values of $\mathbf{y}_i$ and can be denoted as,

$$\Theta : \min_{\Theta}\{\|\mathbf{y}_i - \tilde{f}(\mathbf{x}_i, \Theta)\|^2\} \tag{4.4}$$

However, as the minimization problem is highly ill conditioned in most practical situations, the solution is regularized for each parameter separately using suitable constraints [30], [16]. The training of each parameter in Θ ($\mathbf{w}_i$, $\mathbf{t}_i$ and σ_i) can be achieved using different methods. Following sections review the training algorithms for estimating each parameter.

4.4.1 Selection of Centers of Basis Functions ($\mathbf{t}_i$)

The objective of this step is to compute the basis centers to optimally represent the training data. Basis centers refer to cluster centers. Many different clustering methods are proposed in theory and one of them can be chosen to compute the clusters in the training data. The clustering methods can be classified into two types based on knowledge of the class names beforehand: (1) supervised clustering (class names are known) and (2) unsupervised clustering (no information about the classes beforehand).

The supervised clustering approaches use the input, as well as output, data for deriving the clusters. Instead of representing the training data compactly, these approaches try to find the relation between the distribution of the data in input space and output space. In the context of RBFNN, the class information is not known beforehand and, hence, supervised clustering cannot be used.

The unsupervised clustering can be performed using a number of approaches. An example of the use of orthogonal least squares method for center selection is described in [14]. This method is based on Gram-Schmidt orthogonalization procedure and tries to model the input training data ($\mathbf{x}_i$) with a linear mapping, as given in following equation.

$$\mathbf{x}_i = \sum_{i}^{p} \mathbf{d}_i a_i + e \tag{4.5}$$

where $\mathbf{d}_i$ are the orthogonal vectors in $\Re^N$, obtained using Gram-Schmidt orthogonalization and a_i are the coefficients and e is the residual error and $p < N$. Another approach based on Fisher ratio is described in [27]. The most basic approach that is commonly used for center selection is called K-means clustering [11]. K-means clustering is extremely intuitive and a simple algorithm, and produces near optimal results in most cases [11]. However, it is sensitive to the choice of starting point of the operation and an incorrect selection of the starting point may result in poor clustering result. The ISODATA clustering process, as described in [2], is an enhanced version of K-means clustering. The important steps in ISODATA are:

1. The user defined thresholds are defined as:
   n_{max} = maximum number of samples per cluster.
   n_{min} = minimum number of samples per cluster.
   d_{max} = maximum spread of a cluster.
   d_{min} = minimum spread of a cluster.
2. Apply standard K-means clustering to get initial m number of clusters C_i, where $i = 1, ..., m$. Let n_i and d_i denote the number of samples and spread of the i^{th} cluster.
3. The following operations are performed on all clusters:
   (a) If $n_i < n_{min}$ or $d_i < d_{min}$, merge the cluster to the neighboring cluster.
   (b) If $n_i > n_{max}$ or $d_i > d_{max}$, split the cluster into two clusters.
4. Let n' be the number of clusters after applying thresholds in step-3.
5. If $n \neq n'$, go to step 2. Else compare C_i and C_i' for $i = 1, ..., n$. If $C_i \neq C_i'$ for any i, then go to step 2. Else go to Step 6.
6. Stop.

ISODATA provides a sophisticated control over the cluster formation and is less sensitive to the initial center selection than K-means clustering. This algorithm is used for all the implementations of RBFNN in this dissertation.

4.4.2 Selection of Spreads or Radii of the Centers(σ_i)

Traditionally all the centers are assumed to have the same spread. This rule restricts the RBFNN to a constant resolution and limits its performance. In this dissertation, each center of the RBFNN is designed to have a different spread, making it a variable resolution RBFNN. When the input data has uniform distribution in the input space, single resolution RBFNN performs well. However, when the data is sparse and not uniformly distributed, the variable resolution RBFNN adapts to the training data and improves the performance. Determination of the spread of each center can be done using features of the well established ISODATA algorithm. There are two main properties that are useful in determining the spread: (1) Inter-cluster distances (γ_i) and (2) Intra-cluster distances (s_i).

The γ_i and s_i are defined as,

$$\gamma_i = \min\{\|\mathbf{t}_i - \mathbf{t}_j\|, j = 1, ..., P \text{ and } j \neq i\}, i = 1, ..., P \tag{4.6}$$

$$s_i = \max\{\|\mathbf{x}_j - \mathbf{t}_i\|, j = 1, ..., P\}, i = 1, ..., P \tag{4.7}$$

γ_i give information about the distance of the nearest neighboring cluster from a given cluster. When the spread of a center is based on this parameter, the basis function covers the entire space between all the neighboring clusters, thereby providing a complete mapping of the input space covered by the training data. However, if the inter-cluster distances are very large compared to the intra-cluster distances (as is the typical case for high dimensional data), the mapping obtained using γ_i becomes too flat and almost constant over all the input space. In other words, the nodes of hidden layer lose their discrimination capabilities over the input space.

s_i give information about the variation inside each cluster. This choice provides good discrimination inside all the clusters; however, when the test sample lies outside the cluster, the mapping may not be optimal.

RBFNN is implemented using both the metrics for performance comparison.

4.4.3 Computation of Weights ($\mathbf{w}_i$)

This is final step in the training of RBFNN and it requires input as well as output signals in the training data. Let the number of training samples be n. The input training data can be represented in the form of a matrix $\mathbf{X}$ of dimensions N x n and output data can be represented by a matrix $\mathbf{Y}$ of dimensions M x n. Using the input data, radial basis function, centers and their spreads the matrix of basis function can be computed as,

$$\mathbf{F}(i,j) = (\|\mathbf{x}_i - \mathbf{t}_j\|^2 + 1)^{-1/\sigma(j)^2}, i = 1, ..., n; j = 1, ..., P \tag{4.8}$$

$\mathbf{F}$ has dimensions n x P. The RBFNN equation in matrix form can then be written as,

$$\mathbf{FW} = \mathbf{Y} \tag{4.9}$$

The weight matrix $\mathbf{W}$ of dimensions P x M can be computed using the pseudo inverse of matrix $\mathbf{F}$ and output matrix $\mathbf{Y}$. However, direct use of pseudo inverse in an ill conditioned problem results in an impractical solution. Hence, there is need for using a regularization procedure. The solution of Equation 4.9 can be regularized using Landweber-Fridman [4] type iterative algorithm. Although this method works well for data of smaller dimensions, this method can take a very long time for convergence in the case of data with dimensionality as high as 10000, and also produces a large computation error. In most situations, a direct regularization scheme can be used. A regularizing matrix $\mathbf{F}_0$ is computed using the same power basis function as,

$$\mathbf{F}_0(i,j) = (\|\mathbf{t}_i - \mathbf{t}_j\|^2 + 1)^{-1/\sigma(j)^2}, i = 1, ..., P; j = 1, ..., P \tag{4.10}$$

The pseudo inverse matrix $\mathbf{F}^+$ is computed using the regularizing matrix $\mathbf{F}_0$ as,

$$\mathbf{F}^+ = (\mathbf{FF}' + \lambda\mathbf{F}_0)^{-1}\mathbf{F} \tag{4.11}$$

The parameter λ is a positive constant such that $0 < \lambda < 1$. The exact value of λ is chosen empirically. Herein λ is computed using the matrix norm as $\lambda = \frac{\|\mathbf{F}\|}{100000}$. The details of this regularization scheme can be found in [16]. This method is computationally quite efficient and also provides good regularization performance. This scheme is used throughout this dissertation.

4.5 Experimental Setup for Testing RBFNN

The RBFNN architecture described in the previous two sections is implemented using Matlab and tested with experimental MFL signals obtained from the inspection of 6 inch diameter pipes with specific defects milled on the outer surface. The total data contains 363 defects obtained from 7 pipes. Each pipe is inspected three times (run#0, run#1 and run#2) and data from each run is collected. One pipe has wall thickness of 0.432 inches while the other 6 pipes have wall thickness of 0.280 inches. The data from the pipe with higher wall thickness possesses a lower base value and hence lower amplitude. The base value is removed from the raw MFL data and is then processed with the HOST filter to segment the regions of interest (ROIs) containing potential defects. The ROIs are then standardized to a size of 42x220. The 3-dimensional defect geometry corresponding to each defect is obtained from the actual measurements of the defects using bridging bar assembly [31]. (The defect profiles used in this dissertation are provided by Microline$^{\mathrm{TM}}$.) The depth profile data is standardized to a size of 54x200. The first dimension is the length (l) of the defect and the latter dimension is taken as the width (w) of the defect. The standard sized 2D data is then converted to 1D vector for further processing. Hence, the input MFL signal vector is of dimension 9240 and the output depth profile vector is of dimension 10800. The defects to be analyzed are divided into four categories: pinholes, axial grooves, circumferential grooves and general corrosion. These definitions are only specific to this dissertation, and slightly different definitions are used in the pipeline industry which has more categories. Table 4.1 gives the defect classification. All the length and width limits are in inches and the depths are in terms of percentages of wall-thickness.

The labeling of the defect class, based on their length and width, is illustrated in Figure 4.4. The defects corresponding to the region ($0 < w < 1$) and ($1 < l < 6$) produce extremely

Table 4.1. Defect classes used for training.

Defect Category	Defect Classification	Depths of Defects	#of defects
Pinhole	$l \leq 1$ and $w \leq 1$	20% to 80%	186
Axial Grooves	$1 \leq l \leq 6$ and $2 \leq w \leq 2$	20% to 80%	45
Circumferential Groove	$l \leq 1$ and $1 \leq w \leq 6$	20% to80%	111
General Corrosion	$1 \leq l \leq 6$ and $2 \leq w \leq 6$	20% to 80%	21

feeble MFL signals and are difficult to analyze with reasonable accuracy. Therefore these defects are not analyzed.

Signals obtained as a result of MFL inspection are divided into two sets: training set and test set. The training set is used to estimate parameters of the RBFNN as described above. The inversion performance of the trained RBFNN is tested using the test set.

A pair of input and output signals for the training of RBFNN is shown in Figure 4.5.

4.6 Preliminary Results

The traditional RBFNN is applied to the data collected as described in the previous section. Centers of the basis function are computed using ISODATA algorithm, radii are computed using intra-cluster distance and weights are computed using the method of regularization described in the previous section. Gaussian function is used as basis function. Data in each category is randomly divided into a training set and a test set. Roughly 70% of the data is used for training and the remaining 30% is used for testing. The numerical error $\bar{\epsilon}$ is computed using the formula given below.

$$\bar{\epsilon} = \frac{1}{n} \sum_{i=1}^{n} \frac{\|\mathbf{y}_i - \hat{\mathbf{y}}_i\|}{\|\mathbf{y}_i\|} \tag{4.12}$$

where $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$ represent the actual and predicted values of the depth profile respectively, and n is the number of test samples in each category. The numerical errors in each category are given in Table 4.2.

Although the values in Table 4.2 provide a relative comparison of the different approaches, they do not provide reasonable estimates of individual performances. To get

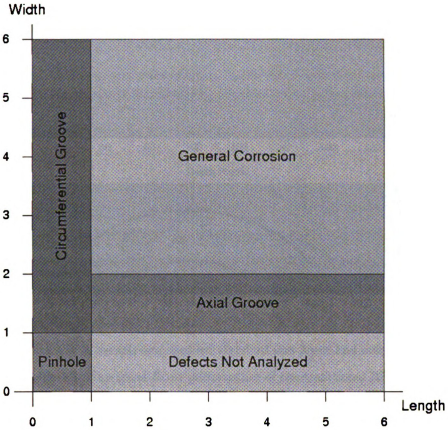


Figure 4.4. Categorization of defects based on axial length and circumferential width.

better idea of the performance of each neural network, cross sections of the depth profiles from test results are shown in Figures 4.6 and 4.7.

In order to improve the performance of RBFNN some modifications are proposed. The details of each modification are described in the following sections.

4.7 Modifications to the Traditional RBFNN Architecture

4.7.1 Dimensionality Considerations

Dimensionality of the data to be analyzed is continuously increasing. One reason behind this is the easy storage of large amounts of data using cheap hardware and the availability of

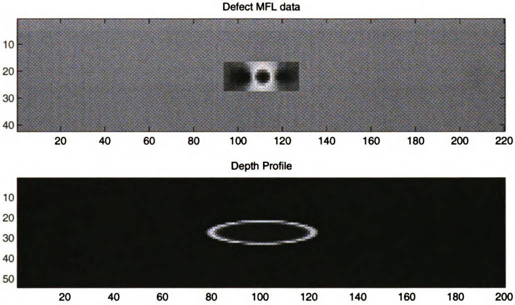
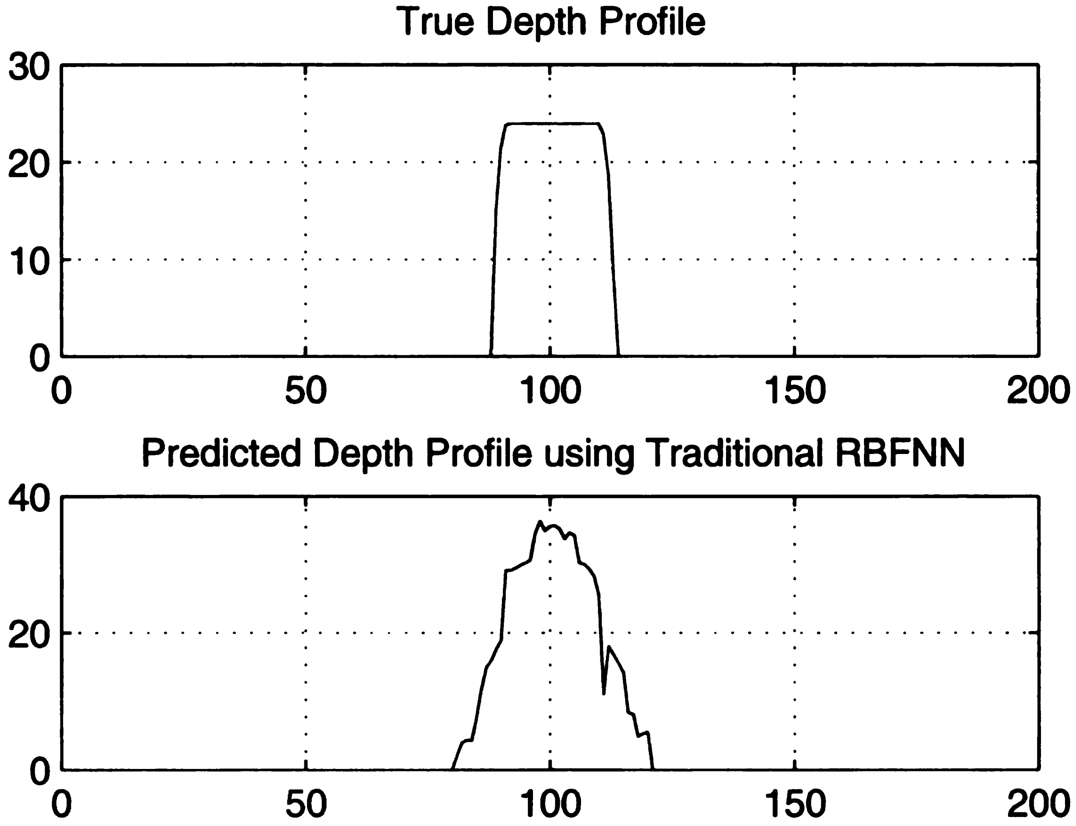


Figure 4.5. A sample zero padded standard size input and output pair.

Table 4.2. Numerical Error performance of the traditional RBFNN.

Defect Category	$\bar{\epsilon}$
Pinhole	0.66
Axial	0.77
Circumferential	0.64
Corrosion	0.48

fast processing computers. On the same note, with advances in instrumentation technology, higher resolution sensors are available which produce data with higher dimensionality. With higher dimensionality, the volume of storage space grows exponentially, and training data requirements also grow exponentially. This phenomenon is called the curse of dimensionality (COD), [3], [34], [12] and [38]. Increased dimensionality does not always translate into proportional increase in information content in the data. A measure was defined for getting the true dimensionality of the data by Fukunaga and Olsen in 1971 [13], which is referred to as the intrinsic dimensionality (ID) of the data. They suggest a step by step algorithm for computing the true dimensionality of the data by inspecting local neighborhoods in the


True Depth Profile

Predicted Depth Profile using Traditional RBFNN

Figure 4.6. Sample result using traditional RBFNN.

data. They use the traditional Karhunen-Loève Transform (KLT) to get the dimensionality of local neighborhoods. They define a threshold in the range of 0.05 to 0.01 for discarding dimensions in the KLT domain. However, with different problems at hand, one has to choose the threshold based on *a priori* information. Other attempts at finding the ID, based on the local neighborhood, can be found in [29], [39]. However, these methods suffer from a common drawback of high sensitivity to noise and edge effects (distortion of data near the class boundaries). A more sophisticated approach was proposed based on optimal topology preserving maps of the data in [6]. An identical problem of finding the ID can also be tackled with global scope in the form of Principal Component Analysis (PCA) or Multi-dimensional Scaling (MDS) [37].

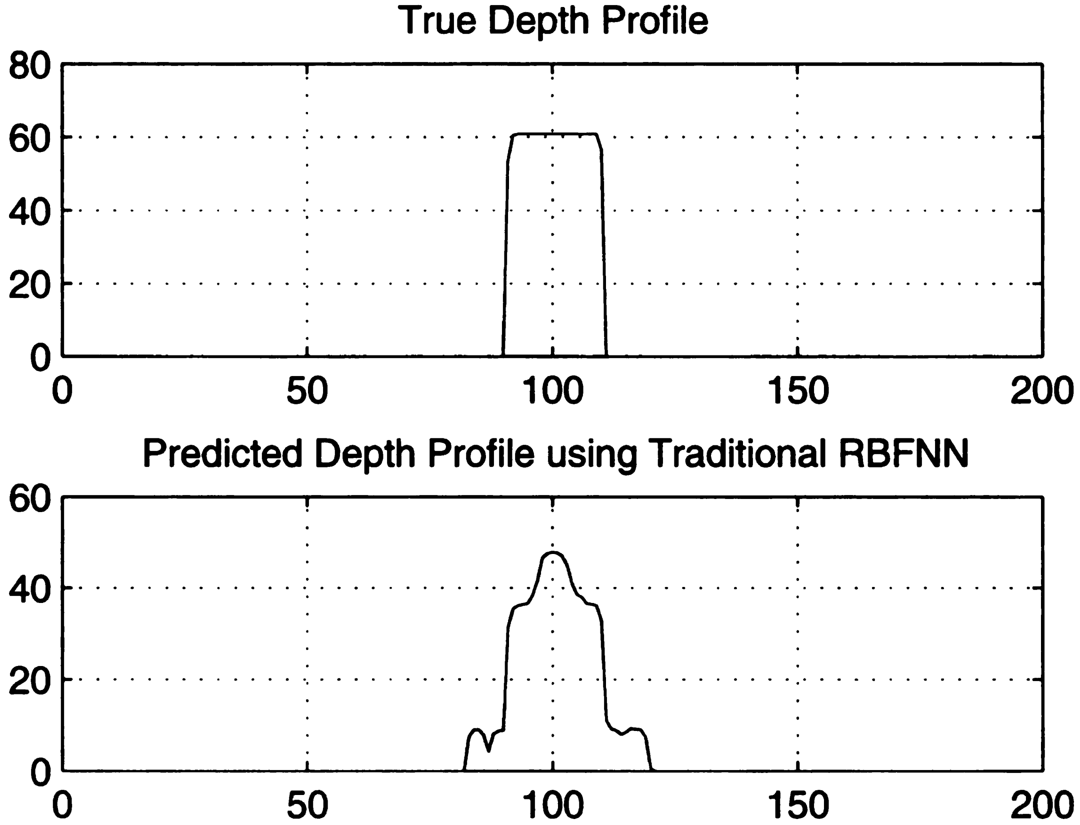


Figure 4.7. Another result using traditional RBFNN.

The estimate of such an ID is necessary to gauge the true performance of a neural network. Given that the MFL signal dimensionality is as high as few thousand and training samples as low as few hundred, theoretically it is not feasible to apply any kind of pattern classifier. All the algorithms described above try to find the ID of the data using unsupervised methods. However, liberty was taken to use the supervised methods for ID estimation, due to the knowledge of true depth profile for each MFL training signal. It is important to note that the ID estimation is only required for theoretical assurance of reliable performance of the RBFNN classifier.

The ID of MFL signal is estimated based on the depth profile data rather than on the MFL signal. As the depth profile data truly decides the search domain, this is a valid assumption. Applying a singular value decomposition (SVD) technique on the 10800 dimensional 363 depth profile samples, it is found that more than 99% of the energy is contained in the first 26 eigenvalues. A semi-log plot of the first 100 eigenvalues of the depth profile data is shown in Figure 4.8. This analysis gives an idea of the ID of the MFL signal and establishes theoretical ground for analysis of the data with RBFNN.

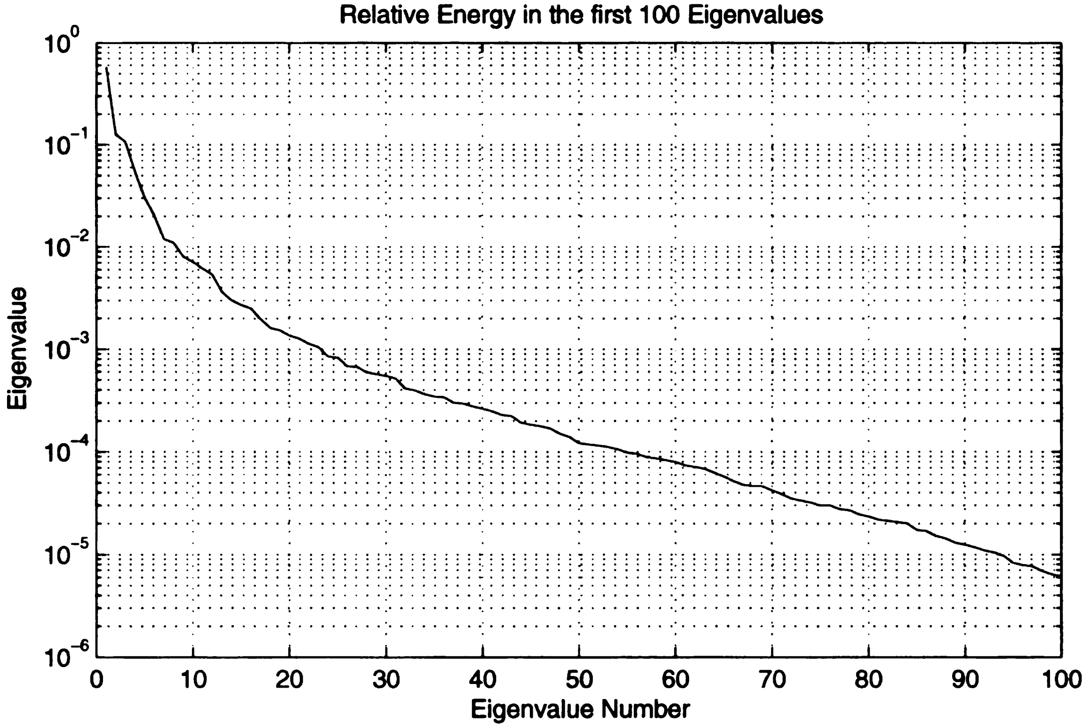


Figure 4.8. Semi-log plot of the energy contained in the first 100 eigenvalues of the depth profile data.

4.7.2 Use of New Basis Function

The most commonly used radially symmetric functions are the Gaussian (f_G) and inverse multiquadrics (f_{IM}) defined in Equations 4.13 and 4.14.

$$f_G(\|\mathbf{x}-\mathbf{t}_i\|,\sigma_i) = \exp(-\frac{\|\mathbf{x}-\mathbf{t}_i\|^2}{\sigma_i^2}) \tag{4.13}$$

$$f_{\mathrm{IM}}(\|\mathbf{x}-\mathbf{t}_i\|,\sigma_i) = \frac{1}{\sqrt{\|\mathbf{x}-\mathbf{t}_i\|^2+\sigma_i^2}} \qquad (4.14)$$

Any function which is continuous on $[0,\infty)$ and differentiable on $(0,\infty)$ can be used as a basis function, [30]. Considering these constraints, a new basis function referred to as the power function (f_{P}) is proposed herein. The power function is defined in Equation 4.15 [22].

$$f_{\mathrm{P}}(\|\mathbf{x}-\mathbf{t}_i\|,\sigma_i) = (\|\mathbf{x}-\mathbf{t}_i\|^2+1)^{-1/\sigma(i)^2} \qquad (4.15)$$

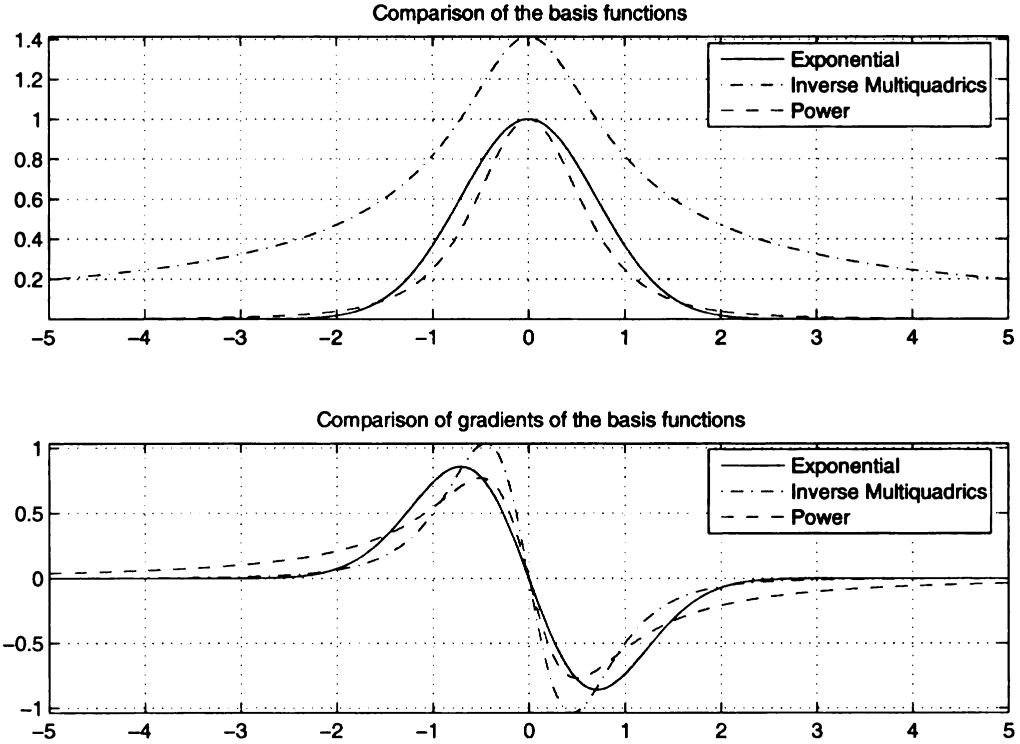


Figure 4.9. Comparison of the three basis functions and their gradients when the constant σ is 0.5.

The three basis functions given in Equations 4.13, 4.14 and 4.15 are plotted in Figure 4.9. The functions f_{G} and f_{IM} tend to become flat at large distances from the center, at which point their gradients also become extremely small. Consequently these gradients cannot be computed numerically with sufficient accuracy. In other words, these functions lose their discriminating capabilities at large distances. At large distances, it is seen that f_G decays rapidly in contrast to f_{IM} which decays at a lower rate. The proposed power function f_P and its gradient decay even more slowly, so that it preserves its discriminating properties at large distances, which is an essential requirement while working with very high dimensional (dimensionality of few thousand) data.

4.7.3 Use of Adaptive Radii

In section 4.4.2, two different approaches for choosing the radius, or spread of each center, are described. Both the methods are inadequate to handle the high dimensional sparse data presented by the MFL signal. In order to combine the advantages of both the methods, a new approach is proposed that makes the radii of basis functions adaptive with respect to the test sample. The resulting RBFNN is called as RBFNN-Ar. In this approach, the s_i are used to choose spreads during the training phase in order to preserve discriminating properties of each cluster. However, these radii are modified to adapt to test data at hand during the test phase. If a test sample is found to lie outside all the clusters, the cluster nearest to the test sample is found and its radius is increased to accommodate this test point. The implementation details of the algorithm are given below.

RBFNN-Ar Algorithm

1. Let distances of a sample $\mathbf{x}_n$ from all the centers be $d_i, i = 1, ..., P$.

2. The minimum distance is identified as $d_k, 1 \leq k \leq P$. In other words, the given test sample lies closest to center $\mathbf{t}_k$.

3. Threshold ρ, $0 < \rho \leq 1$ is chosen as a measure of the maximum allowable value of r that affects the discriminating properties of the cluster. Empirically the optimal value of ρ is chosen to be 0.1.

4. Ratio r, is computed as $r = d_k/\sigma_k$, where σ_k^2 is the spread of the k^{th} center. This ratio gives information about the location of the test sample with reference to the cluster

at center $\mathbf{t}_k$. If the value of r is less than or equal to ρ, then $\mathbf{x}_n$ lies well inside the cluster and possess all the properties of the cluster.

5. If $r > \rho$, then the spread of the corresponding cluster is modified as $\sigma_k = d_k/\rho$, otherwise the spread is kept unchanged.

6. The traditional RBFNN algorithm is applied using the modified σ_k.

The results obtained by applying the RBFNN-Ar algorithm with all three basis functions are discussed in the next section.

4.8 Testing of RBFNN with Modifications

The RBFNN with f_{G} basis function is called as RBFNN-G. Similarly, RBFNN with f_{IM} and f_{P} basis functions are called RBFNN-IM and RBFNN-P, respectively. When used in conjunction with adaptive radii algorithm, the networks are referred to as RBFNN-G-Ar, RBFNN-IM-Ar and RBFNN-P-Ar, respectively. In each case, data is randomly divided into a training set and a test set such that 70% of the data is in training set and 30% is in test set. Performance results are shown in Figures 4.10, 4.11, 4.12 and 4.13.

4.9 Concluding Remarks

The normalized error in prediction is computed using the formula $\epsilon = \frac{\|\mathbf{y}-\mathbf{y}'\|}{\|\mathbf{y}\|}$ for every iteration. Mean of the normalized error $\bar{\epsilon}$ is computed for each category of defect. These mean normalized errors, corresponding to the different RBFNNs,$\bar{\epsilon}_{\mathrm{G}}, \bar{\epsilon}_{\mathrm{IM}}, \bar{\epsilon}_{\mathrm{P}}, \bar{\epsilon}_{\text{G-Ar}}, \bar{\epsilon}_{\text{IM-Ar}}, \bar{\epsilon}_{\text{P-Ar}}$, are reported in Table 4.3. Except for the category of corrosion, RBFNN-P-Ar performs better than all the other algorithms. There are only 21 defects in the category of general corrosion and, hence, the results based on this data are not statistically significant. Future work includes testing of these methods with more training data.

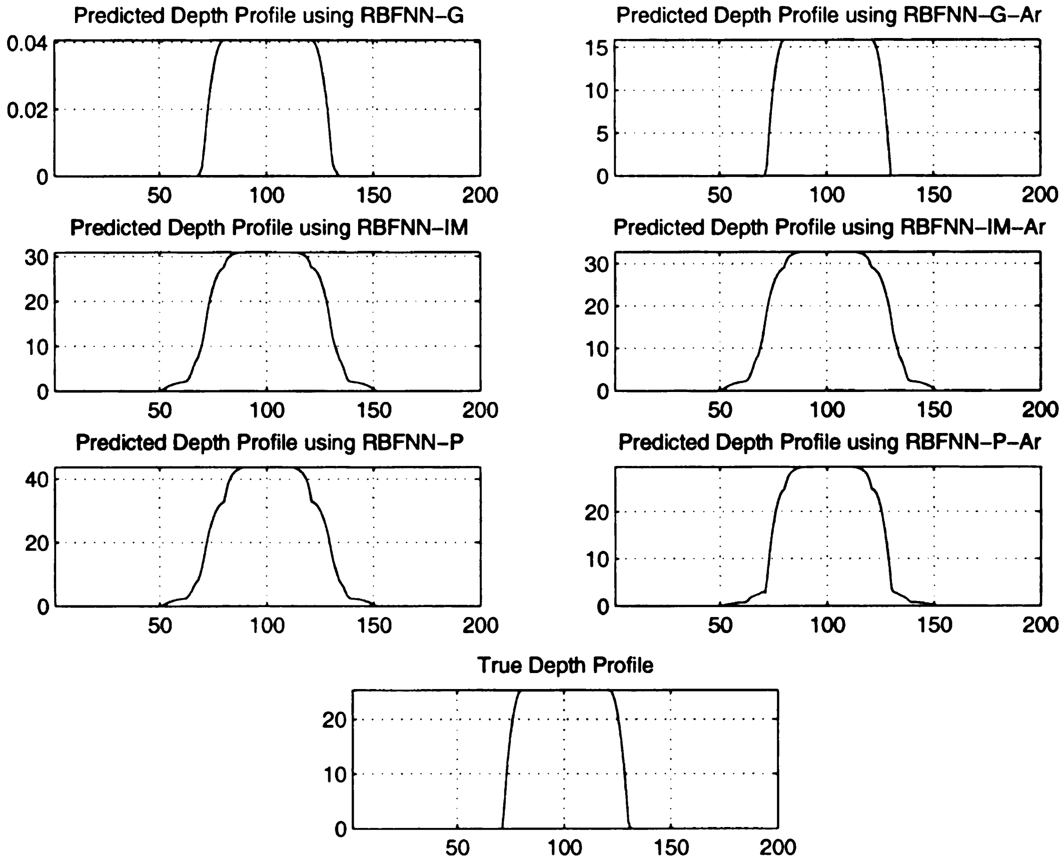


Figure 4.10. Performance of various RBFNNs on an axial defect.

Table 4.3. Quantitative Comparison of the various RBFNNs.

Defect Category	$\bar{\epsilon}_{\mathrm{G}}$	$\bar{\epsilon}_{\mathrm{IM}}$	$\bar{\epsilon}_{\mathrm{P}}$	$\bar{\epsilon}_{\mathrm{G\text{-}Ar}}$	$\bar{\epsilon}_{\mathrm{IM\text{-}Ar}}$	$\bar{\epsilon}_{\mathrm{P\text{-}Ar}}$
Pinhole	0.66	0.47	0.46	0.36	0.68	0.34
Axial	0.77	0.24	0.26	0.34	0.42	0.22
Circumferential	0.64	0.42	0.53	0.41	0.70	0.26
Corrosion	0.48	0.30	0.51	0.52	0.83	0.36

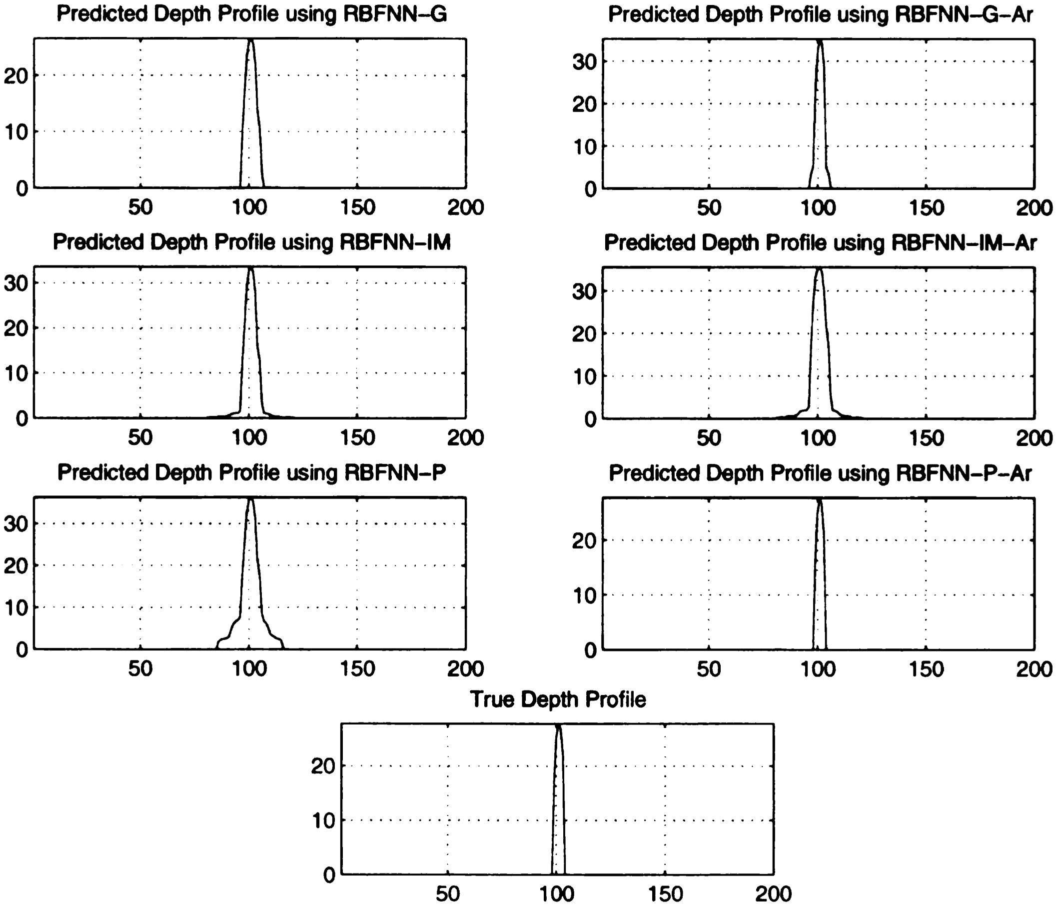


Figure 4.11. Performance of various RBFNNs on a circumferential defect.

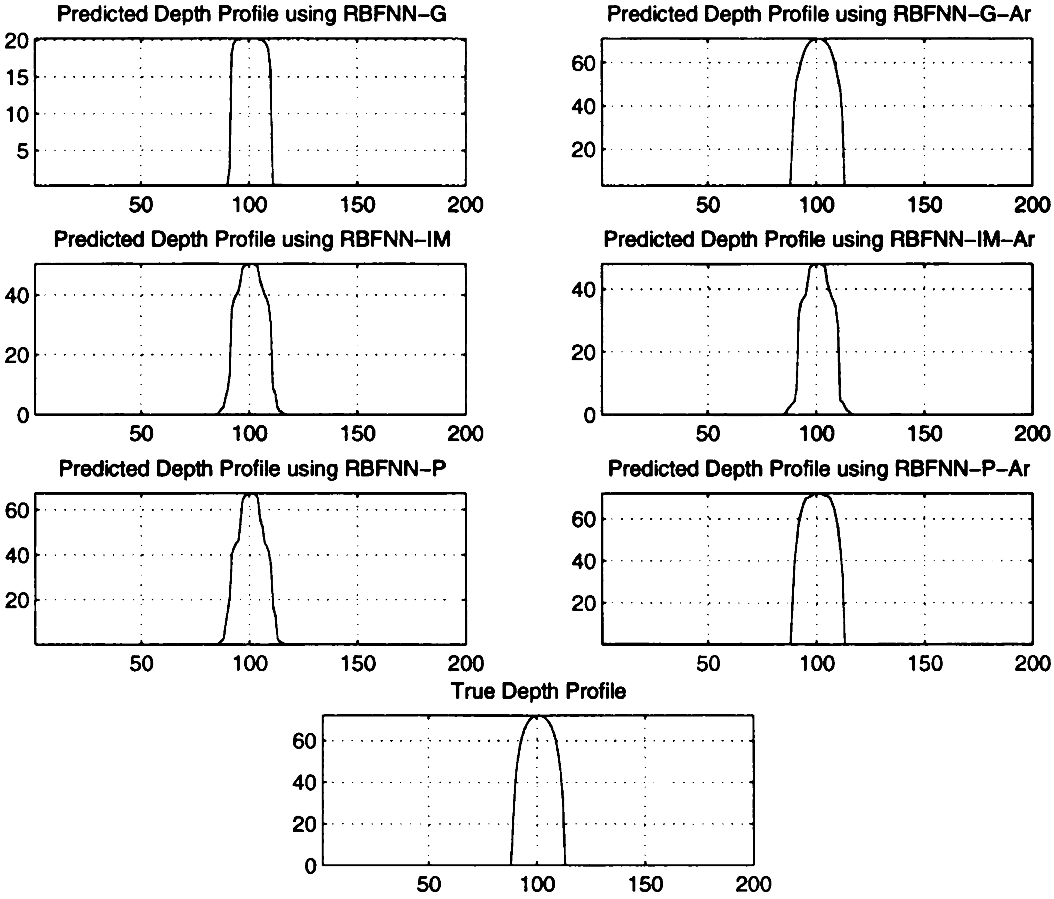


Figure 4.12. Performance of various RBFNNs on a pinhole defect.

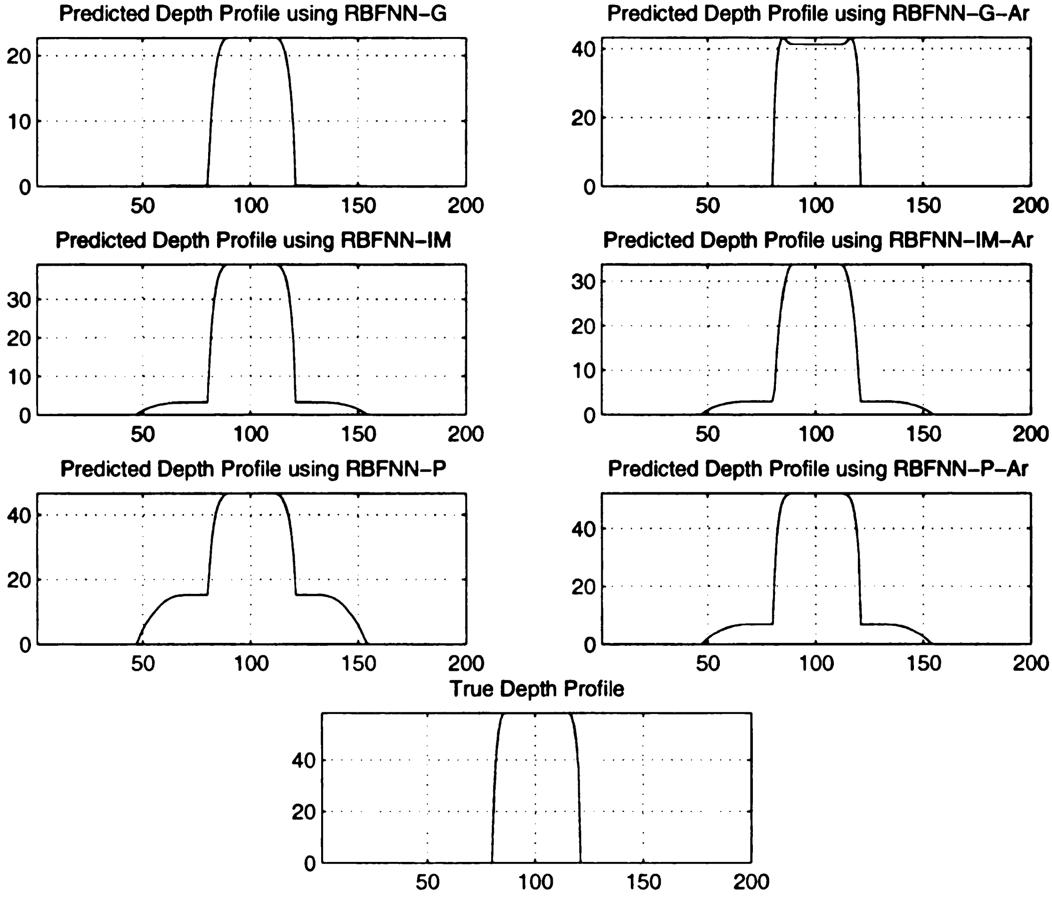


Figure 4.13. Performance of various RBFNNs on a general corrosion type defect.

CHAPTER 5

Defect Profiling: Iterative Inversion Techniques

5.1 Basic Iterative Inversion

In the iterative approach for the solution of an inverse problem, a model of the well-behaved forward process was first developed. The forward model predicts the MFL signal for a given defect geometry. The forward model can be obtained using analytical or numerical analysis techniques [28], [24]. In this work, the forward model is obtained using the RBFNN architecture as described in the previous chapter with the input and output vectors reversed. The input to the forward model is the depth profile and the output is the corresponding MFL signal. As discussed in Chapter 4, RBFNN-P (RBFNN with power basis function as defined in Equation 4.15) provides the most accurate results among the different variations of RBFNN. Hence, RBFNN-P is chosen to serve as the forward model. The RBFNN-Ar algorithm is not used, as the model must remain unchanged during the test phase of iterative inversion. For the purpose of simplicity, RBFNN-P is referred to as RBFNN henceforth in this chapter.

5.1.1 Training and Test Phases of Basic Iterative Inversion

The RBFNN representing the forward process is trained using the training data as described in the previous chapter. A block diagram of the iterative inversion approach in the test phase is shown in Figure 5.1.

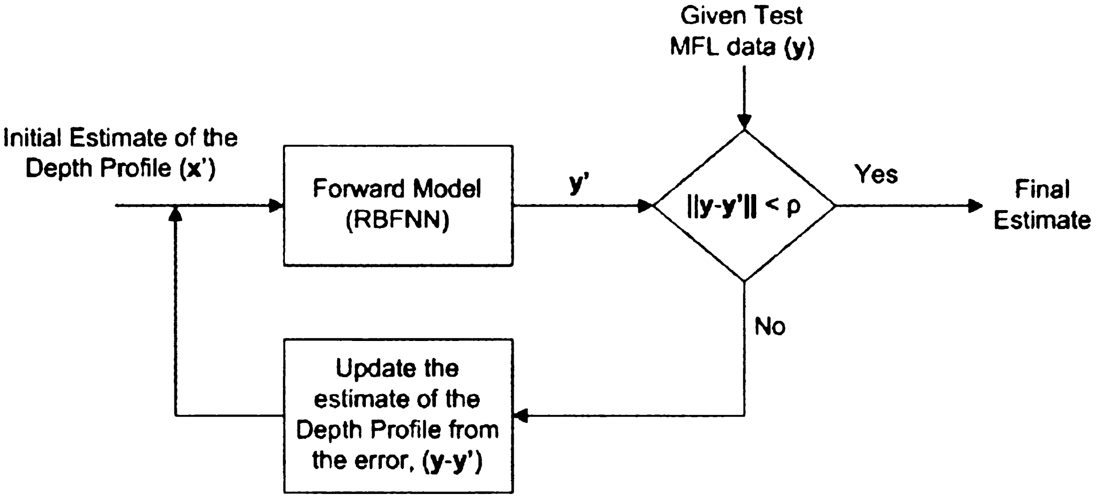


Figure 5.1. Basic architecture of iterative inversion for prediction of depth profile.

MFL measurement data **y** is given and depth profile of the corresponding defect is to be predicted. The process starts with an initial guess of the depth profile $\mathbf{x}'$. The initial guess is typically taken as a zero vector. Then the forward RBFNN is used to predict the corresponding MFL signal, $\mathbf{y}'$. Scalar valued squared error is computed from the knowledge of given MFL signal **y**, as ($e^2 = ||\mathbf{y} - \mathbf{y}'||^2$). This error is used to update the estimate of the depth profile $\mathbf{x}'$ using a steepest descent method. In order to have a mathematical expression for the update rule, let us denote the RBFNN transformation of $\mathbf{x}'$ as $\xi(\mathbf{x}')$, hence $\mathbf{y}' = \xi(\mathbf{x}')$. The expression for squared error in terms of $\mathbf{y}'$ and $\xi(\mathbf{x}')$ can be computed as,

$$e^2 = ||\mathbf{y} - \mathbf{y}'||^2 \tag{5.1}$$

$$e^2 = ||\mathbf{y} - \xi(\mathbf{x}')||^2 \tag{5.2}$$

$$e^2 = [\mathbf{y} - \xi(\mathbf{x}')]^T[\mathbf{y} - \xi(\mathbf{x}')] \tag{5.3}$$

The update equation for the steepest descent method is given as, $\mathbf{x}'(n+1) = \mathbf{x}'(n) - \eta\nabla_{\mathbf{x}'}e^2$, where n denotes the iteration number and η denotes the step size parameter. η is computed using a line minimization algorithm. Gradient of the squared error with respect to vector $\mathbf{x}'$ is computed as,

$$\nabla_{\mathbf{x}'}e^2 = -2[\nabla_{\mathbf{x}'}\xi(\mathbf{x}')]^T[\mathbf{y} - \xi(\mathbf{x}')] \tag{5.4}$$

Gradient computation is essentially a vector operation. Gradient of a scalar is a vector and gradient of a vector results in a matrix. Thus, gradient of the squared error e^2 is a vector, however gradient of $\xi(\mathbf{x}')$ is a matrix. This matrix is called Jacobian. The analytical expression for the Jacobian $\nabla_{\mathbf{x}'}\xi(\mathbf{x}')$ is computed from the definition of RBFNN given in Equation 4.3 in Chapter 4 as,

$$\begin{aligned}
\nabla_{\mathbf{x}'}\xi(\mathbf{x}') &= \nabla_{\mathbf{x}'}(\sum_{i=1}^{P}\mathbf{w}_i(\|\mathbf{x}' - \mathbf{t}_i\|^2 + 1)^{-1/\sigma(i)^2}) + \mathbf{w}_0 && (5.5)\\
\nabla_{\mathbf{x}'}\xi(\mathbf{x}') &= \nabla_{\mathbf{x}'}(\sum_{i=1}^{P}\mathbf{w}_i(\|\mathbf{x}' - \mathbf{t}_i\|^2 + 1)^{-1/\sigma(i)^2}) && (5.6)\\
\nabla_{\mathbf{x}'}\xi(\mathbf{x}') &= \sum_{i=1}^{P}\mathbf{w}_i\nabla_{\mathbf{x}'}(\|\mathbf{x}' - \mathbf{t}_i\|^2 + 1)^{-1/\sigma(i)^2} && (5.7)\\
\nabla_{\mathbf{x}'}\xi(\mathbf{x}') &= -\sum_{i=1}^{P}\frac{\mathbf{w}_i}{\sigma_i^2}(\|\mathbf{x}' - \mathbf{t}_i\|^2 + 1)^{-1/\sigma(i)^2} - 1\nabla_{\mathbf{x}'}([\mathbf{x}' - \mathbf{t}_i]^T[\mathbf{x} - \mathbf{t}_i] + 1) && (5.8)\\
\nabla_{\mathbf{x}'}\xi(\mathbf{x}') &= -\sum_{i=1}^{P}\frac{\mathbf{w}_i}{\sigma_i^2}(\|\mathbf{x}' - \mathbf{t}_i\|^2 + 1)^{-1/\sigma(i)^2} - 1(2[\mathbf{x}' - \mathbf{t}_i]) && (5.9)
\end{aligned}$$

The above expression is a matrix of size M x M. As discussed in Chapter 4, for experimental data, the value of M is 10800. In order to store the large size matrix in double precision format, 933.12 megabytes of memory is required. Computation of inverse of such large matrices makes this algorithm slow in convergence and also prone to numerical errors. On a typical PC with about 1 gigabytes of memory, this algorithm cannot be executed. Therefore dimensionality of the data must be reduced in order to be able to use the iterative inversion. The problem of high dimensionality is addressed using a wavelet based multi-resolution approach discussed in the next section.

5.2 Use of Wavelet Transform

The wavelet transform is a relatively new development in the field of signal processing. The fourier transform has been the most widely used transform in signal processing and analysis. The fourier transform represents any signal in terms of an infinite series of sinusoids. The forward fourier transform is defined in Equation 5.10 and the inverse transform is defined in Equation 5.11.

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx \tag{5.10}$$

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du \tag{5.11}$$

In general, when a signal is transformed into the frequency domain, it loses all temporal information. In an wavelet transform, the signal is represented in terms of a series of functions of finite duration waves called wavelets. The wavelets should satisfy the condition of orthogonality for unique and complete representation of the signal [26]. Due to a finite duration of the wavelets, the wavelet domain representation preserves temporal or spatial information as well as frequency domain information simultaneously. Also, with the variable duration or scale of the wavelets, multi-resolution analysis is possible using just one transform. A wavelet transform can be defined using continuous wavelets (continuous wavelet transform or CWT) as well as discrete wavelets (discrete wavelet transform or DWT). A continuous wavelet series expansion (Inverse CWT) [15] is given in Equation 5.12 and the computations of the wavelet coefficients (Forward CWT) is given in Equations 5.13 and 5.14 respectively.

$$f(x) = \sum_{k} c_{j_0}(k) \varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty} \sum_{k} d_j(k) \psi_{j,k}(x) \tag{5.12}$$

$$c_{j_0}(k) = \langle f(x), \varphi_{j_0,k}(x) \rangle = \int f(x) \varphi_{j_0,k}(x) dx \tag{5.13}$$

$$d_j(k) = \langle f(x), \psi_{j,k}(x) \rangle = \int f(x) \psi_{j,k}(x) dx \tag{5.14}$$

$\psi(x)$ is called as wavelet function and $\varphi(x)$ is called as scaling function. The variable j_0 is an arbitrary starting scale, $c_{j_0}(k)$ are called approximation or scaling coefficients and $d_j(k)$

are called detail or wavelet coefficients.

When the signals are available in discrete form, it is more convenient to use the DWT. DWT is used herein for the proposed iterative inversion algorithm and is explained in detail in the following section.

5.2.1 Discrete Wavelet Transform (DWT)

Let $f(x)$ be a 1D discrete function. The DWT and inverse DWT (IDWT) with respect to $f(x)$ are defined as [15],

$$\mathbf{W}_{\varphi}(j_0, k) = \frac{1}{\sqrt{M}} \sum_{x} f(x) \varphi_{j_0,k}(x) \tag{5.15}$$

$$\mathbf{W}_{\psi}(j, k) = \frac{1}{\sqrt{M}} \sum_{x} f(x) \psi_{j,k}(x) \tag{5.16}$$

for $j \geq j_0$,

$$f(x) = \frac{1}{\sqrt{M}} \sum_{x} \mathbf{W}_{\varphi}(j_0, k) \varphi_{j_0,k}(x) + \frac{1}{\sqrt{M}} \sum_{x} \mathbf{W}_{\psi}(j, k) \psi_{j,k}(x) \tag{5.17}$$

The functions $\varphi_{j_0,k}(x)$ and $\psi_{j,k}(x)$ are functions of discrete variable x, $x = 0, 1, ..., M-1$. Normally $j_0 = 0$ and M is an integer power or 2, $M = 2^J$. The summations are performed over $x = 0, 1, ..., M-1$, $j = 0, 1, ..., J-1$ and $k = 0, 1, ..., 2^j - 1$. φ is called a scaling function and ψ is called a wavelet function similar to the definition of continuous wavelet transform. The coefficients $\mathbf{W}_{\varphi}(j_0, k)$ are called approximation coefficients and $\mathbf{W}_{\psi}(j, k)$ are called detail coefficients. Together, $\mathbf{W}_{\varphi}(j_0, k)$ and $\mathbf{W}_{\psi}(j, k)$ represent the complete DWT of $f(x)$.

Different systems of wavelet functions are proposed that satisfy the condition of orthogonality, eg. Haar, Morlet, Mexican Hat, Daubechies. The Haar system (scaling and wavelet function) is one of the most simple and yet effective systems and is used herein. The scaling and wavelet functions of the Haar system are shown in Figure 5.2.

In order to illustrate the DWT operation, consider an arbitrary finite duration function $f(x) = [-2, -1, 0, 2, 3, -3, 1, 0, 2]$, for $x = 0, 1, ..., 8$ as shown in Figure 5.3.

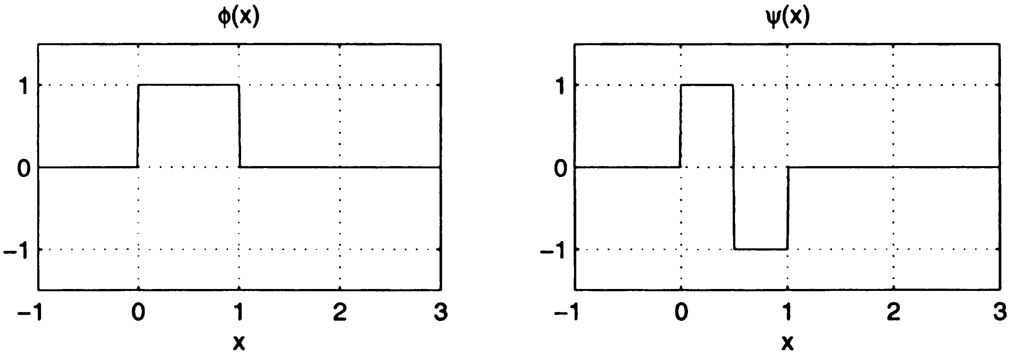


Figure 5.2. The scaling and wavelet functions of Haar system.

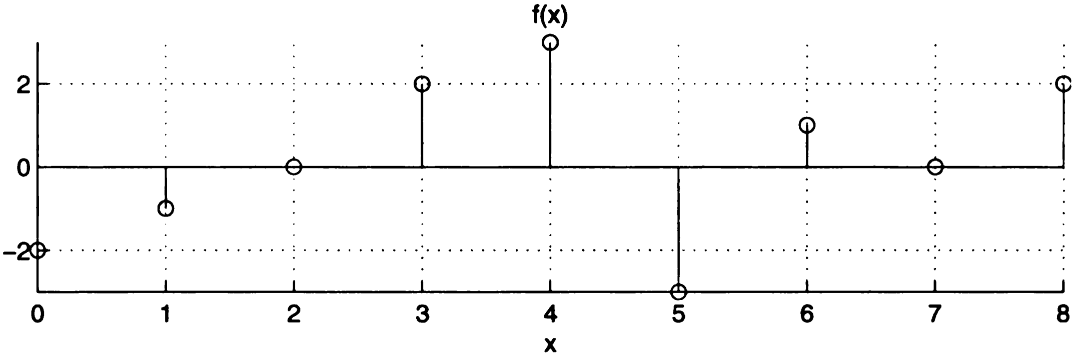


Figure 5.3. The function $f(x)$.

The Discrete Wavelet Transform of $f(x)$ (the approximation coefficients $\mathbf{W}_\varphi$ and the detail coefficients $\mathbf{W}_\psi$) is computed using Equations 5.15 and 5.16. The coefficients are plotted in Figure 5.4.

The approximation coefficients $\mathbf{W}_\varphi$ approximate the function $f(x)$ using half the number of points. A detail that is missing in these coefficients is represented in the detail coefficients $\mathbf{W}_\psi$. Using these two coefficients the original function $f(x)$ can be reconstructed using Equation 5.17 without error.

The same concept of 1D-DWT is extended to the case of 2D functions of discrete variables and is used herein for data representation. The original depth profile consists of dimensions (42 x 220). This data is re-sampled using a simple averaging to a new resolu-

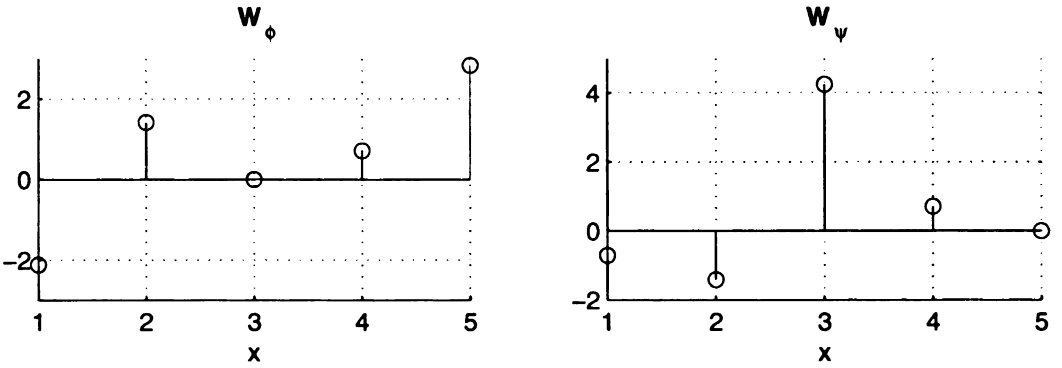


Figure 5.4. DWT of $f(x)$ using Haar system.

tion of (32 x 32). This resolution is easier to use with DWT as the dimensions are integer powers of 2. The original depth profile, with full resolution, is stored as vector $\mathbf{x}$ and the re-sampled depth profile is denoted as vector $\mathbf{h}$. The 2D-DWT of $\mathbf{h}$ is computed and the resulting coefficients (approximation and detail) are denoted as vector $\mathbf{c}$. When all the coefficients are preserved, the dimensionality of $\mathbf{c}$ remains same as that of $\mathbf{h}$. In order to reduce the dimensionality of the data with minimal loss of information, it is transformed to a wavelet domain using the *Adaptive Wavelet Selection Scheme* (AWSS) [36] developed in this thesis. The AWSS algorithm provides a systematic procedure for converting $\mathbf{h}$ to $\mathbf{c}$.

5.3 Iterative Inversion with Adaptive Wavelet Compression

The iterative algorithm described in section 5.1 provides a sound theoretical framework for solving the inverse problem. However, when the data dimensionality is high the algorithm tends to be slow and prone to computational errors when implemented on a regular computer. To address this issue, an innovative approach using discrete wavelet transform is proposed. The fundamental idea underlying the algorithm can be stated as follows: "Obtain the coarse representation of the desired depth profile and then increase the resolution in only those spatial areas that need to be represented in higher resolution". This approach helps in having a compact representation of the depth profile using fewer parameters. Before giving the details of the algorithm the different data vector formats used in the algorithm are described.

5.3.1 Transformations

The MFL signal and depth profiles are both represented in the form of 2D matrices. For ease of representation, both matrices are converted to one-dimensional column vectors. Denote the given MFL signal vector as $\mathbf{y}, \mathbf{y} \in \Re^N$ and the estimated depth profile vector as $\mathbf{x}$, $\mathbf{x} \in \Re^M$. A 2D Haar wavelet transform is used to represent the depth profile in the wavelet domain. For simplicity all the 2D Haar wavelets are also represented in the form of vectors. The vector $\mathbf{x}$ is represented using Haar wavelets is given in Equation 5.18.

$$\mathbf{x} = \sum_j \sum_k \Psi(j,k)\mathbf{c}(j,k) \tag{5.18}$$

where variable j denotes the level of resolution and variable k denotes the wavelets number at j^{th} resolution. $\Psi(j,k)$ is the vector representation of the k^{th} 2D Haar wavelet at j^{th} resolution level. $\mathbf{c}(j,k)$ is the corresponding scalar coefficient. Equation 5.18 is similar to Equation 5.17. However, rather than representing it separately, the scaling function is also included in Ψ.

In order to further simplify the wavelet representation, another variable denoted as $\mathbf{I}, \mathbf{I} \in \Re^{n+1 \times 2}$ is introduced such that for each $\Psi(i)$, $\mathbf{I}(i) = \{j,k\}$, where j is the level of resolution and k is the wavelet number in that resolution. Thus using variable I, the double summation involved in the DWT expansion can be written using single summation as,

$$\mathbf{x} \ = \ \sum_{i=0}^{i=n} \Psi(i)\mathbf{c}(i) \tag{5.19}$$

where all the wavelets in all the resolutions are incorporated in $\mathbf{I}(i), i = 0, 1, ..., n$.

Equation 5.19 can also be written in matrix form as,

$$\mathbf{x} = \Psi\mathbf{c} \tag{5.20}$$

where, $\Psi, \Psi \in \Re^{M \times n+1}$. Each column of matrix Ψ is obtained from $\Psi(i)$ in Equation 5.19.

$\mathbf{c}, \mathbf{c} \in \Re^n + 1$ contains all the corresponding coefficients.

The steps in the proposed Algorithm

1. Initialize the current level of resolution, $L = 1$ and fix highest resolution level as L_{MAX}.
2. Store the paired list containing level number (j) and corresponding wavelet decomposition number (k) at that level as, $I(i) = [j, k]$. Store the corresponding 2D Haar wavelets (scaling as well as wavelet function matrices) in the form of vectors as $\mathbf{\Psi}(i)$.
3. Compute the coefficients for each of the wavelets in I for the representation of $\mathbf{h}$ and list them as vector $\mathbf{c}$ of length n_c, (2D-DWT). $\mathbf{c}(i) = \mathbf{\Psi}(i) \circ \mathbf{h}$. Where the symbol $\circ$ denotes correlation operation.
4. Compute the relative energy of all the coefficients as, $e(i) = (c(i))^2 / \sum_{i=1}^{i=nc} (c(i))^2$.
5. Discard the coefficients in $\mathbf{c}$ for which $e(i) < \epsilon_c$ and also remove the corresponding wavelets in I. ϵ_c is a pre-determined constant and is chosen empirically. Currently chosen as 0.02.
6. Re-compute the relative energy for the remaining coefficients in $\mathbf{c}$ as given in step 4.
7. Select the coefficients for which $e(i) > \epsilon_g$, and store them in vector $\mathbf{c}_g$, the corresponding wavelets in I are stored in I_g. ϵ_g is a pre-determined constant and is chosen empirically. Currently chosen as 0.2.
8. If $\mathbf{c}_g$ is empty go to step 11, otherwise increase the current level, $L = L + 1$.
9. If $L > L_{\text{MAX}}$ go to step 11 else, compute the wavelets at level L for all the wavelets in I_g, and store them as I_g^L.
10. Append the set of wavelets to I as, $I = \{I, I_g^L\}$ and go to step 3.
11. Report the final wavelet representation in terms of I and $\mathbf{c}$.

The inverse process of obtaining **h** from given **c** is done using the IDWT defined by,

$$\mathbf{h} = \sum_{i=1}^{i=nc} \mathbf{c}(i)\boldsymbol{\Psi}(i) \tag{5.21}$$

Figure 5.5 is a representation of the proposed iterative inversion using adaptive wavelet compression (AWSS) [23] in the form of a flow chart.

The forward neural network is trained as described in Chapter 4. This network is capable of predicting the MFL signal **y** corresponding to the given depth profile **h**. In the basic approach described in the first section of this chapter, the process of iterative inversion begins with an estimate of the depth profile. In the given situation the depth profile is represented in wavelet domain as coefficients vector **c**. Hence the iterative inversion process begins with initial estimate of **c**, which is typically assumed as a zero vector or a flat depth profile. **c** is converted into the estimate **h**. The forward neural network is then used to obtain the corresponding estimate **y**′ of the MFL signal. The squared error between the given MFL signal **y** and the estimate **y**′ is computed as $e^2 = (\|\mathbf{y} - \mathbf{y}'\|)$. The gradient of the squared error is a Jacobian and is used for updating the value of estimate **c** using the gradient descent method described before as, $\mathbf{c}(n+1) = \mathbf{c}(n) - \eta\nabla_{\mathbf{c}}e^2$, where n is the iteration number. This update reflects as update of the depth at each point in the depth profile. The Jacobian is computed as,

$$\nabla_{\mathbf{c}}e^2 = \nabla_{\mathbf{h}}e^2\nabla_{\mathbf{c}}\mathbf{h} \tag{5.22}$$

The Jacobian with respect to **h** is computed as given in Equations 5.4 and 5.9. The Jacobian $\nabla_{\mathbf{c}}\mathbf{h}$ is computed from Equation 5.21 as,

$$\begin{aligned} \nabla_{\mathbf{c}}\mathbf{h} &= \nabla_{\mathbf{c}} \sum_{i=1}^{i=nc} \mathbf{c}(i)\boldsymbol{\Psi}(i) & (5.23) \\ \nabla_{\mathbf{c(i)}}\mathbf{h} &= \boldsymbol{\Psi}(i) & (5.24) \end{aligned}$$

The iterations are continued till the error becomes lower than the pre-determined threshold or the maximum number of iterations is reached. The full resolution depth profile **x** is finally computed by re-sampling the final estimate **h**.

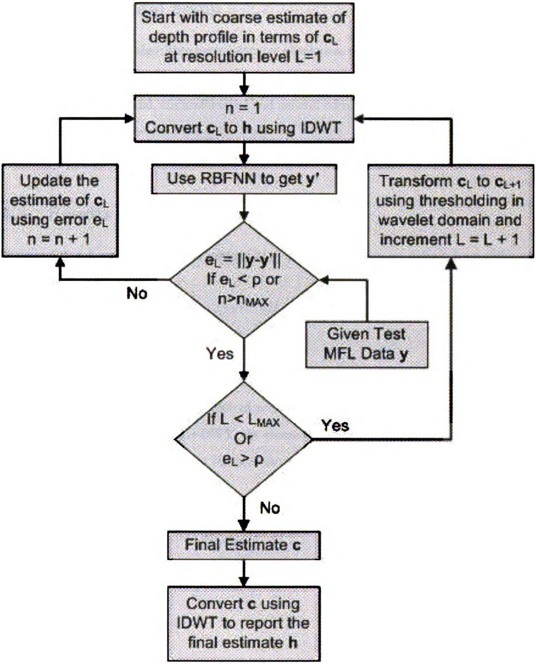


Figure 5.5. Flowchart of AWSS based iterative inversion process.

5.4 Experimental Setup

The iterative inversion algorithm using adaptive wavelets as described in the previous section is tested on the same data that was used in chapter 4. There are total of 363 defects in four categories (pinholes, axial, circumferential and general corrosion) as described in Table 4.1. The forward model is trained using leave-one-out scheme. In this scheme all the training samples in the given defect category are used except for the one that is used for testing.

5.5 Results and Discussion

Figures 5.6, 5.7 and 5.8 show the performance of the algorithm on defects from different categories.

5.6 Concluding Remarks

The iterative inversion using adaptive wavelets provide a sophisticated framework for prediction of the depth profile at any desired precision. Due to the dimensionality reduction obtained with adaptive use of 2D-DWT, the computation is fast and accurate. The parameters ϵ_c and ϵ_g provide control over the process of adaptive DWT representation and make the algorithm more robust.

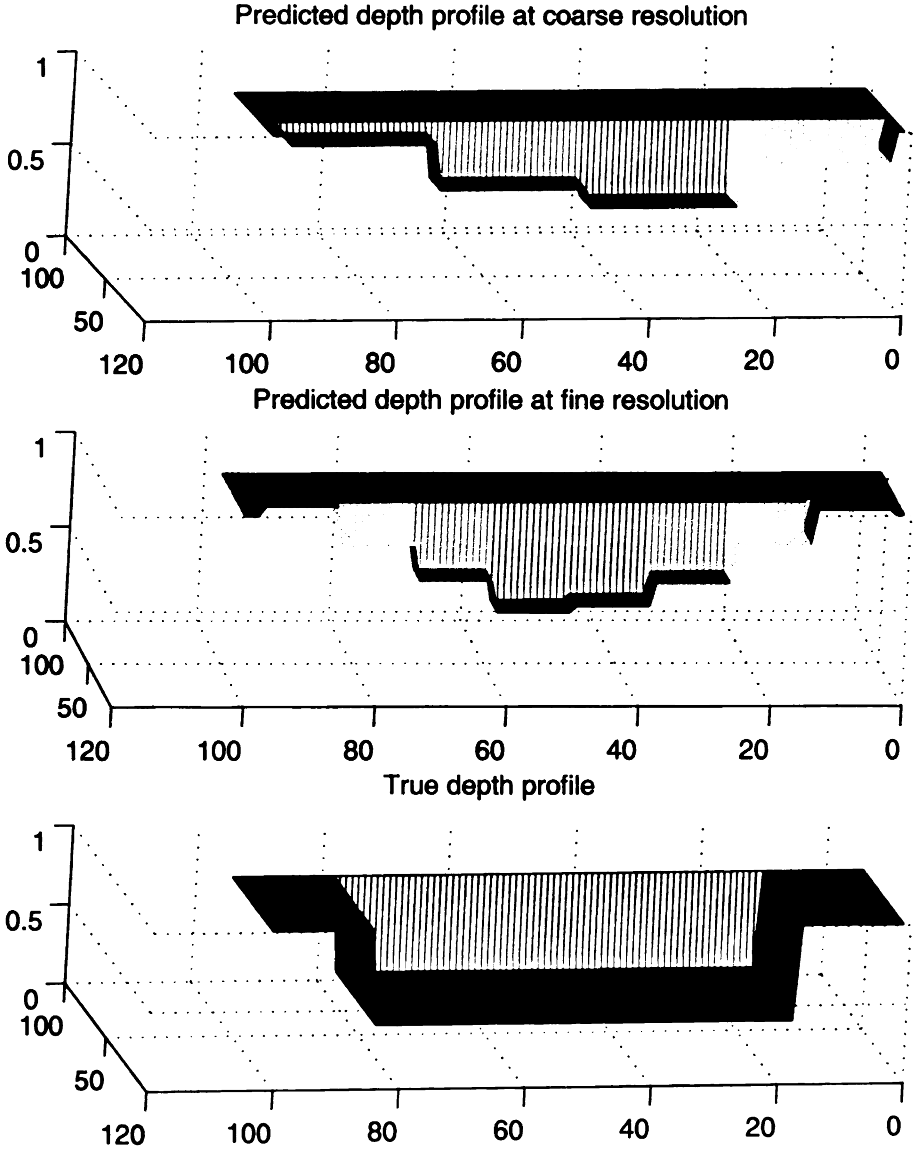
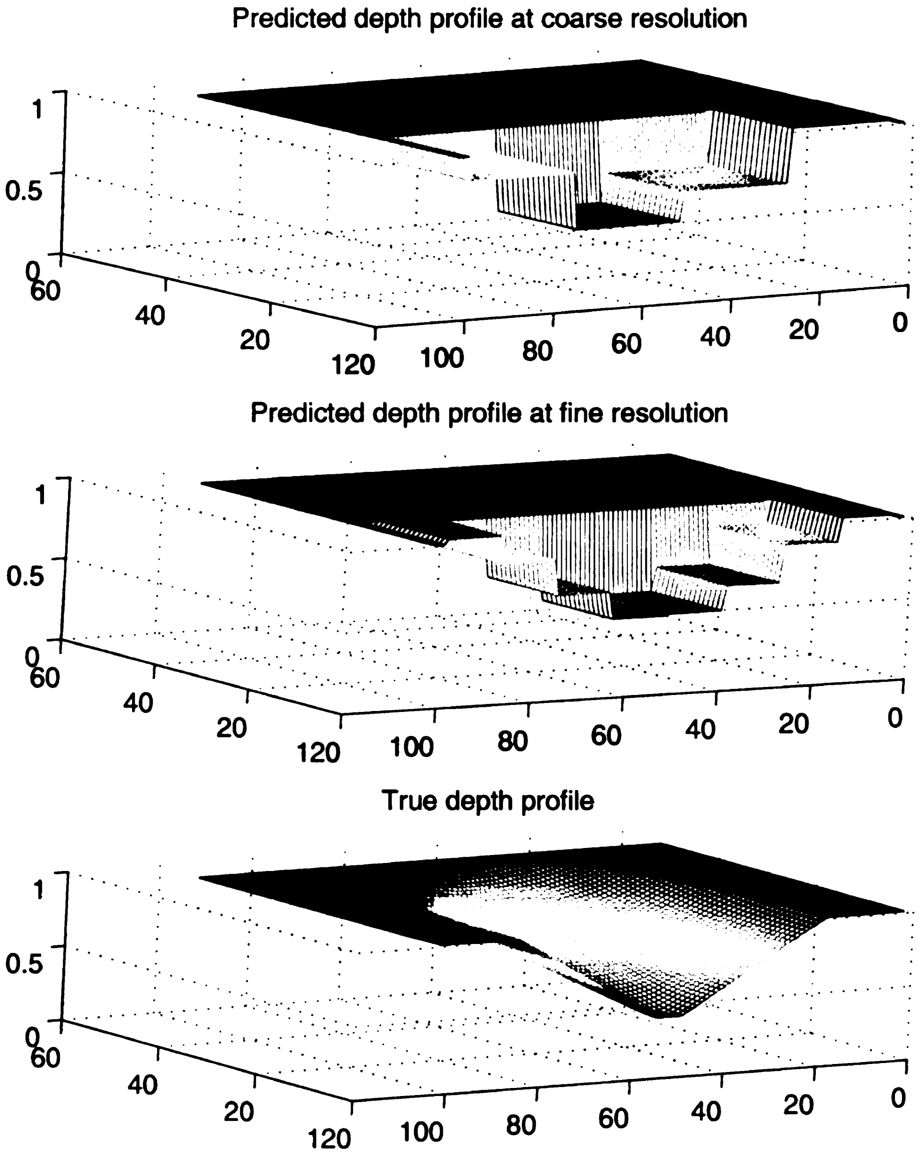


Figure 5.6. Result of predicted depth profile at second level of resolution.



Predicted depth profile at coarse resolution

Predicted depth profile at fine resolution

True depth profile

Figure 5.7. Another result of predicted depth profile at second level of resolution.

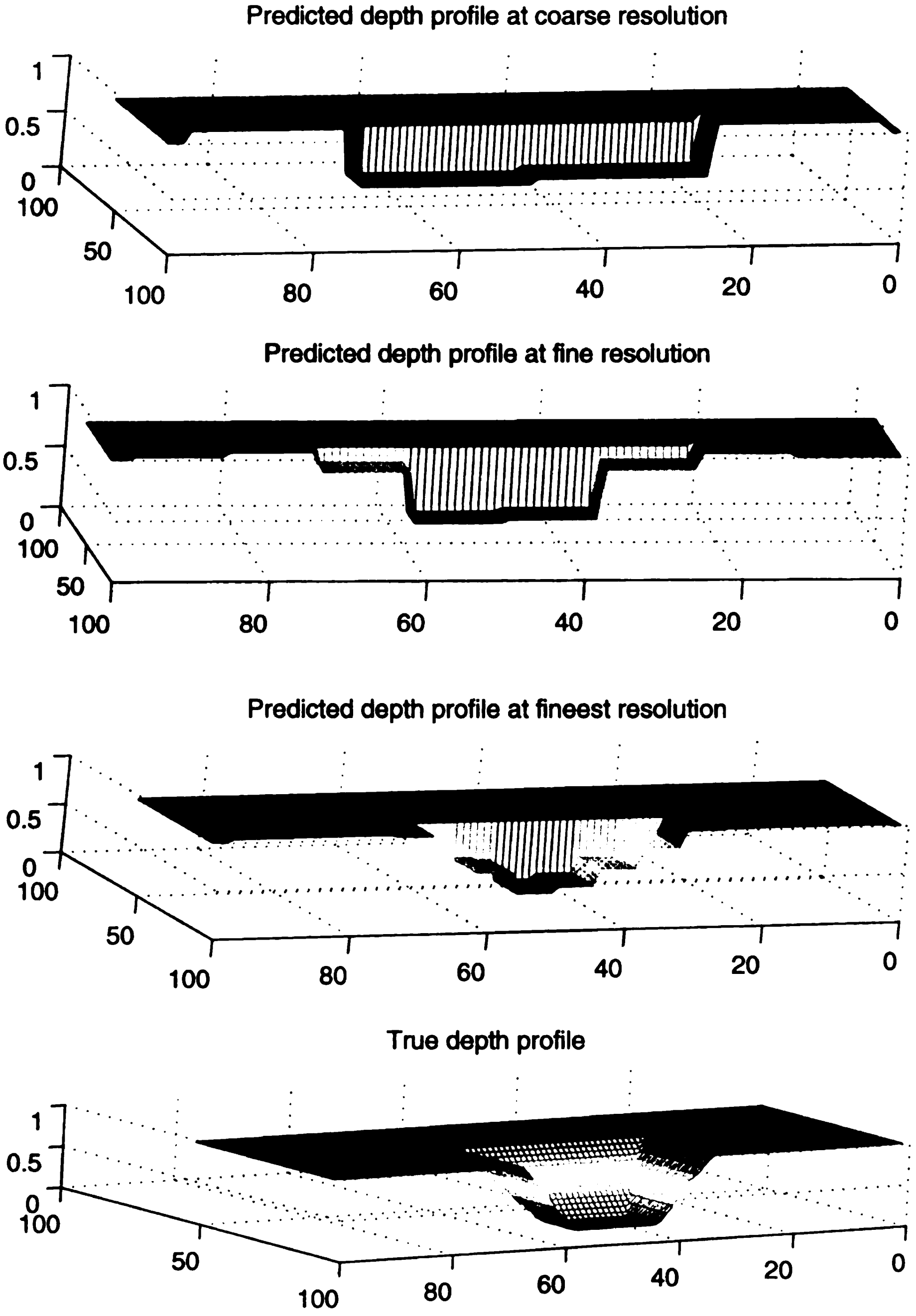


Figure 5.8. Result of predicted depth profile at third level of resolution.

CHAPTER 6

Conclusions and Future Work

6.1 Scope and Summary

Development of a fully automated analysis system for interpretation of signals obtained from the inspection of gas transmission pipelines is the primary objective of this dissertation. The final output of the proposed system is a 3-dimensional depth profile estimate of each defect on the pipe surface.

It is assumed that the MFL signal is processed offline (there is no direct feedback from the analysis system to the inspection system) after the inspection is completed. The data used in this dissertation is provided by Microline^{TM}, who also funded the research.

The proposed system processes the raw MFL signal to derive the final result, which is a 3-dimensional depth profile of each defect in the pipe-wall. The processing is divided into two phases: (1) Training Phase and (2) Test Phase. The initial preprocessing steps are common to both the phases of operation.

Common Steps of Preprocessing

1. Apply higher order statistics transformation (HOST) filters, (HOST-r and HOST-c) to the MFL signal to identify and segment the regions of interest (ROIs) containing potential defects. This step greatly reduces the data that need to be analyzed in the

subsequent steps.

2. Compensate the segmented MFL signal for variations in permeability and wall thickness of the pipe material and velocity of measurement.
3. Standardize the size of compensated MFL signal in each segment and convert it to a one dimensional vector.

Steps in Training Phase

1. Standardize the size of output depth profile vector in the same way as that for segmented and compensated MFL signal.
2. Use this data for training the radial basis function neural network (RBFNN) with the newly proposed *power basis function* as given in Equation 4.15. The network is called as RBFNN-P

Steps in Test Phase

1. Apply the MFL signals corresponding to the test measurements to RBFNN-P. The depth profiles of the defects in these measurements are known but are not used by the RBFNN-P. Predict the depth profile for each segmented defect using direct inversion methods described in Chapter 4 and iterative inversion method with adaptive wavelets described in Chapter 5.
2. These predictions are then compared with the known values to evaluate the performance of the proposed algorithms.

6.2 Original Contributions to the Field

The research carried out during the progress of the dissertation has made following new contributions to the field:

1. A novel method using higher order statistics (skewness and kurtosis) for the segmentation of defects in MFL signal corrupted with varied magnitudes of correlated noise is developed, implemented and evaluated. This method is shown to be better suited for fully automated analysis of MFL signal as opposed to the traditional adaptive filtering methods.

2. A new basis function, the *Power Function*, as given in Equation 4.15, is developed for use with radial basis function neural networks (RBFNNs). This function is a modified form of the popular gaussian function and satisfies the continuity and differentiability properties required by any basis function. This function verifies the improvement in the performance of RBFNN when the data dimensionality is high.

3. Use of adaptive wavelets is proposed as an intermediate step in the iterative inversion of a MFL signal. This modification reduces the dimensionality of the signal being processed without compromising the performance. The reduction in dimensionality makes the algorithm faster and more efficient. The adaptive wavelets algorithm enables coarse to fine improvement in the prediction of the depth profile thereby providing better control over the accuracy of the solution.

6.3 Future Work

This dissertation addresses the inverse problem of prediction of depth profile from the MFL signal using two different approaches. However, before processing the MFL data it needs to be compensated. Some of the linear compensation techniques for Wall thickness, permeability compensation and velocity compensation are discussed in Chapter 3. Developing non-linear techniques to improve the compensation performance is left as a future work.

Both the approaches for inversion discussed herein, are based on the use of RBFNNs. The direct inversion approach uses a RBFNN to model the inverse process of predicting the depth profile from a given MFL signal, while the iterative inversion algorithm uses a RBFNN to model the forward process of predicting the MFL signal from the given depth profile. Theoretically, the latter method is more robust and with the use of an adaptive multi-resolution wavelet transform, it also provides a better control over the accuracy of depth profile prediction.

This dissertation proposes several modifications to the traditional architecture of a RBFNN to achieve optimal performance with high dimensional data. The newly developed methods are tested on experimental data, and the results are promising. It is left as future work to validate the performance of these methods using more exhaustive set of experimental data.

Non-linear dimensionality reduction techniques have always been topics of great interest. The use of adaptive wavelets as proposed in this dissertation has addressed this issue to some level. However, other methods can be investigated in future to improve the compactness of data representation.

The Haar system of wavelets is used in this dissertation considering its ease of use and flexibility. The objective of using the adaptive wavelet algorithm is to represent the defect profile with the least possible error at coarsest possible resolution. It is left as a future work to investigate different systems of wavelet functions to obtain a more optimal representation of the depth profiles at lower levels of resolutions.

Last, but not the least, more exhaustive testing of all the algorithms developed herein is left as future work. The testing using actual field data would provide insight into the strengths and weaknesses in the algorithms in practical situations and help in further improvements of the same.

BIBLIOGRAPHY

[1] M. Afzal, R. Polikar, S. Udpa, and L. Udpa. Adaptive noise cancelation schemes for magnetic flux leakage signals obtained from gas pipeline inspection. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2001.

[2] G. H. Ball and D. J. Hall. Some fundamental concepts and synthesis procedures for pattern recognition preprocessors. *International Conference on Microwaves, Circuit Theory, and Information Theory*, 1964.

[3] R. Bellman. *Adaptive Control Processes: A Guided Tour.* Princeton University Press, Princeton, N.J., 1961.

[4] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging.* IOP Publishing, Bristol, 1998.

[5] N. K. Bose and P. Liang. *Neural Network Fundametals with Graphs, Algorithms and Applications.* McGraw-Hill Inc., USA, 1996.

[6] J. Bruske and G. Sommer. Intrinsic dimensionality estimation with optimally topology preserving maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):572–575, May 1998.

[7] M. Chao, S. S. Udpa, L. Udpa, and W. Lord. Characterization of magnetic flux leakage signals using radial basis function network. *Proceedings of the 3rd Annual Midwest Electrotechnology Conference*, pages 122–126, April 1994.

[8] A. Deforest. Magnetic testing of iron and steel. *Iron & Steel Engineer*, 9, 1932.

[9] F. Doane. Magnaflux inspection for cracks and seams. *Iron Age*, September 1933.

[10] G. Dobmann and P. Hller. *Physical Analysis Methods of Magnetic Flux Leakage*, volume IV. Academic Press, 1980.

[11] Richard Duda, Peter Hart, and David Stork. *Pattern Classification (Second Edition).* John Wiley & Sons, Inc., USA, 2001.

[12] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, USA, 1990.

[13] K. Fukunaga and D. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, c-20(2):176–183, February 1971.

[14] J. Barry Gomm. Selecting radial basis function network centers with recursive orthogonal least squares training. *IEEE Transactions on Neural Network*, 11:306–314, March 2000.

[15] Rafael Gonzalez and Richard Woods. *Digital Image Processing (Second Edition).* Addison-Weseley Publishing Company, USA, 1993.

[16] Simon Haykin. *Neural Networks: A Comprehensive Foundation.* Macmillan College Publishing Company, Inc., 866 Third Avenue, New York, New York 10022, 1994.

[17] Simon Haykin. *Adaptive Filter Theory (Fourth Edition).* Pearson Education, Singapore, 2002.

[18] Simon Haykin and Bernard Widrow. *Least Mean Square Adaptive Filters.* John Wiley and Sons Inc., USA, 2003.

[19] W. Hoke. *Principles of Magnetic Particle Testing.* Magnaflux Corporation, Chicago, 1967.

[20] J. Jackson. *Classical Electrodynamics.* John Wiley and Sons, Inc., 2nd edition, 1974.

[21] A. Joshi, L. Udpa, and S. Udpa. Use of higher order statistics for enhancing magnetic flux leakage pipeline inspection data. *International Symposium on Interdisciplinary Electromagnetic, Mechanic & Biomedical Problems (ISEM)*, September 2005.

[22] A. Joshi, L. Udpa, and S. Udpa. Use of radial basis function neural network to predict 3-dimensional depth profile from magnetic flux leakage inspection. *National Seminar on NDT by Indian Society of Nondestructive Testing (ISNT)*, December 2005.

[23] A. Joshi, L. Udpa, S. Udpa, and A. Tamburrino. Adaptive wavelets for characterizing magnetic flux leakage signals from pipeline inspection. *IEEE International Magnetic Conference (Intermag)*, May 2006.

[24] W. Lord. Applications of numerical field modeling to electromagnetic methods of nondestructive testing. *IEEE Transactions on Magnetics*, MAG-19(6):2437, November 1983.

[25] W. Lord and D. J. Oswald. Leakage field methods of defect detection. *International Journal of Nondestructive Testing*, 4:249–274, 1972.

[26] Stephane Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, USA, 1998.

[27] K. Z. Mao. Rbf neural network center selection based on fisher ratio class separability measure. *IEEE Transactions on Neural Networks*, 13:1211–1216, September 2002.

[28] R. Palanisamy and W. Lord. Finite element modeling of electromagnetic ndt phenomena. *IEEE Transactions on Magnetics*, MAG-15(6):1479–1481, November 1979.

[29] K. Pettis, T. Bailey, A. Jain, and R. Dubes. An intrinsic dimensionality estimator from near-neighbor information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(1):25–37, January 1979.

[30] Tomaso Poggio and Federico Girosi. A theory of networks for approximation and learning. Technical Report AIM-1140, 1989.

[31] G. J. Posakony and V. L. Hill. Assuring the integrity of natural gas transmission pipelines. Technical report, Gas Research Institute, 1992.

[32] P. Ramuhalli, Lalita Udpa, and Satish Udpa. Neural network-based inversion algorithms in magnetic flux leakage nondestructive evaluation. *Journal of Applied Physics*, 93(10):8274–8276, May 2003.

[33] L. Sibul, editor. *Adaptive Signal Processing.* IEEE Press, New York, USA, 1987.

[34] B. W. Silverman. Density estimation for statistics and data analysis. *Chapman and Hall*, 1996.

[35] D. A. Sprecher. On the structure of continuous functions of several variables. *Transactions of the Americal Mathematics Society*, 115:340–355, 1965.

[36] A Tamburrino, S. Ventre, and G. Rubinacci. Reconstruction techniques for electrical resistance tomography. *IEEE Transactions on Magnetics*, 36:1132–1135, July 2000.

[37] W. S. Torgerson. *Theory and Methods of Scaling.* John Wiley and Sons, New York, 1960.

[38] M. Verleysen, D. Franois, G. Simon, and V. Wertz. On the effects of dimensionality on data analysis with neural networks. In J.R. Alvarez eds. J. Mira, editor, *Artificial Neural Nets Problem solving methods*, Lecture Notes in Computer Science 2687, pages II105–II112. Springer-Verlag, 2003.

[39] P. Verveer and R. Duin. An evaluation of intrinsic dimensionality estimators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):81–86, Jauary 1995.

[40] Bernard Widrow and Samuel Stearns. *Adaptive Signal Processing.* Prentice Hall Inc., NJ, USA, 1985.