A "divide-and-conquer" strategy for NDE signal inversion in gas transmission pipelines

Justin Gary Bram
Rowan University

12-31-2006

Let us know how access to this document benefits you - share your thoughts on our feedback form.

Follow this and additional works at: https://rdw.rowan.edu/etd

Part of the Electrical and Computer Engineering Commons

Recommended Citation
https://rdw.rowan.edu/etd/842

This Thesis is brought to you for free and open access by Rowan Digital Works. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Rowan Digital Works. For more information, please contact LibraryTheses@rowan.edu.
A “Divide-and-Conquer” strategy for NDE signal inversion in gas transmission pipelines

by

Justin Gary Bram

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

Department: Electrical and Computer Engineering
Major: Engineering (Electrical Engineering)

Approved:

In Charge of Major Work

For the Major Department

For the College

Members of the Committee:

Rowan University
Glassboro, NJ
2006
ABSTRACT

JUSTIN GARY BRAM
A "DIVIDE-AND-CONQUER" STRATEGY FOR NDE SIGNAL INVERSION IN GAS TRANSMISSION PIPELINES
2006
ADVISOR: SHREEKANTH MANDAYAM
ELECTRICAL AND COMPUTER ENGINEERING

Signal inversion in nondestructive evaluation (NDE) applications is a critical step before remediation decisions are made. The accuracy and confidence of the signal inversion results therefore play a key role in evaluating the effectiveness of the NDE procedure. Conventional NDE signal inversion algorithms that employ artificial neural networks treat all geometric regions of the NDE signal equally. Consequently, when the inversion algorithm is presented with input data that is significantly different from the training data, the performance of the network deteriorates significantly. This thesis presents a superior alternative for NDE signal inversion. Different geometric regions of the NDE signature are assigned different confidence levels; separate neural network inversion algorithms are applied to each region and the results are combined. The neural network inversion algorithm consists of radial basis functions that implement geometric transformations of the input NDE signals. It is shown that this "divide-and-conquer" strategy yields robust results, especially when applied to test data that the neural network has not seen before. While the algorithm is exercised theoretically using simple 1-D and 2-D defect geometries, the technique is also validated using NDE inspection images from a suite of test specimens representative of the in-line inspection of gas transmission pipelines.
Acknowledgements

This work is funded in part by:

National Science Foundation
Major Research Instrumentation (MRI) Program
Award #0216348

&

National Energy Technology Laboratory (NETL)
United States Department of Energy,
Grant DE-FC26-02NT41648
Table of Contents

List of Figures................................................................................................................... vi
List of Tables .................................................................................................................. ix

1. Introduction ............................................................................................................. 1
   1.1. Nondestructive Evaluation ................................................................................ 1
   1.2. Inverse Problem ............................................................................................. 2
   1.3. Virtual Reality ................................................................................................ 4
   1.4. Thesis Overview .............................................................................................. 7
       1.4.1. Motivation .............................................................................................. 7
       1.4.2. Objectives and Scope of the Thesis ....................................................... 9
       1.4.3. Expected Contributions .......................................................................... 9
       1.4.4. Organization ......................................................................................... 10

2. Background ............................................................................................................... 11
   2.1. Solution methods in NDE .............................................................................. 11
       2.1.1. Regularization ...................................................................................... 12
       2.1.2. Iterative Algorithms ............................................................................ 13
       2.1.3. Neural networks .................................................................................. 16
   2.2. RBF Neural Networks ..................................................................................... 19
   2.3. NDE Inspection Techniques ........................................................................... 21
       2.3.1. Magnetic Flux Leakage ........................................................................ 21
       2.3.2. Ultrasound ............................................................................................ 23
   2.4. Virtual Reality ................................................................................................ 26
       2.4.1. Components of a VR World ................................................................... 27
       2.4.2. VR Hardware ....................................................................................... 28
       2.4.3. Reasons for using VR .......................................................................... 30
   2.5. Chapter Summary ............................................................................................ 31

3. Approach .................................................................................................................. 34
   3.1. Definition of Regions ...................................................................................... 35
   3.2. The “divide-and-conquer” Strategy for NDE Signal/Image Inversion ......... 36

4. Results ...................................................................................................................... 52
   4.1. Test Specimen Suite ....................................................................................... 52
   4.2. Test Setup ....................................................................................................... 54
       4.2.1. UT Test Setup ....................................................................................... 54
       4.2.2. MFL Test Setup .................................................................................... 55
   4.3. Training and Test Data Sets ............................................................................ 57
   4.4. Training and Test Results ............................................................................. 61
   4.5. Visualization of Results ................................................................................. 104
   4.6. Discussion of Results ..................................................................................... 112

5. Conclusions ............................................................................................................. 115
List of Figures

Figure 1-1: Generalized illustration of NDE inspection techniques.............................................. 2

Figure 1-2: Representation of a NDE system. .................................................................................. 3

Figure 1-3: Interactions that create a virtual world. ........................................................................ 5

Figure 1-4: An example of advanced scientific visualization of weather data in virtual reality. .............................................................. 5

Figure 1-5: An example of a virtual world used for data analysis of pollution data. ................. 6

Figure 1-6: An example of virtual reality for remote planetary surface exploration............. 6

Figure 1-7: An example of virtual reality used for virtual prototyping of the engine room of the U.S. Navy's DD(X) Destroyer. ................................................................. 7

Figure 1-8: A MFL pig; an in-line gas pipeline inspection tool. .................................................. 8

Figure 2-1: Outline of the direct neural approach to solving the inverse problem in NDE. ................................................................. 19

Figure 2-2: Through transmission ultrasonic inspection method. .............................................. 23

Figure 2-3: Pulse-echo UT system................................................................................................. 24

Figure 2-4: Flat transducer and a focused transducer, which has the addition of a lens. ........ 25

Figure 3-1: Illustration of a method for designating areas of information content for use in an inversion algorithm. ................................................................. 33

Figure 3-2: A direct, one-dimensional numerical example illustrating the process of inversion................................................................................................. 36

Figure 3-3: A “divide-and-conquer”, one-dimensional numerical example illustrating the process of inversion. ................................................................. 38

Figure 3-4: Direct, one-dimensional inversion through a RBF network. ................................. 39

Figure 3-5: A one-dimensional, “divide-and-conquer”, numerical example of a test output for RBF network inversion. ................................................................. 40

Figure 3-6: A two-dimensional, direct example of inversion..................................................... 41
Figure 3-7: A two-dimensional, "divide-and-conquer", numerical example of inversion.

Figure 4-1: Side view of test specimens illustrating the difference in thickness for three specimens.

Figure 4-2: A top view of two specimens. The specimen on the left has no defect while the specimen on the right does.

Figure 4-3: Rowan University NDE Lab's immersion ultrasonic test system.

Figure 4-4: MFL test stand setup.

Figure 4-5: Close-up of test specimen loading area on the MFL system.

Figure 4-6: UT data set 1.

Figure 4-7: MFL data set 1.

Figure 4-8: UT data set 2.

Figure 4-9: MFL data set 2.

Figure 4-10: Data set 1.

Figure 4-11: Data set 2.

Figure 4-12: Training and test results for data set 1. Test instances are labeled as such; all other instances are used for training.

Figure 4-13: Training and test results for data set 2. Test instances are labeled as such; all other instances are used for training.

Figure 4-14: Advanced visualization of the input, outputs, measurements and test specimen used in creating, training and implementing the "divide-and-conquer" neural network inversion algorithm.

Figure 4-15: Advanced visualization of the MFL inspection image used in this data instance.

Figure 4-16: Advanced visualization of a graphical model of the test specimen from which the MFL and UT measurements were taken. A digital image of the actual test specimen was used to texture the graphical model.

Figure 4-17: Advanced visualization of the UT inspection image used in this data instance.
Figure 4-18: Advanced visualization of the result of the lower confidence network for this data instance. ................................................................. 101

Figure 4-19: Advanced visualization of the result of the recombination of the higher and lower confidence network outputs for this data instance. .................................................. 102

Figure 4-20: Advanced visualization of the result of the higher confidence network for this data instance. ............................................................................................................ 103

Figure 4-21: MSE for training data.................................................................................................................. 104

Figure 4-22: MSE for test data.................................................................................................................... 105
List of Tables

Table 2-1: Summary of previous work in signal inversion in NDE. ......................... 11
Table 4-1: Test specimens. .................................................................................. 45
Table 4-2: Data set 1, training and test data. Test data is indicated by a gray highlight. 50
Table 4-3: Data set 2, training and test data. Test data is indicated by a gray highlight. 51
1. Introduction
Natural gas plays an important role in the energy infrastructure of the United States, accounting for 22.9% of energy consumption in 2003 [1]. In 2001, the U.S. consumed 22.6 Trillion cubic feet (Tcf) of natural gas [2]. Of this consumed amount, 85% is produced domestically, with most of the remaining 15% coming from neighboring Canada [2]. In order to transmit and then distribute these trillions of cubic feet of natural gas throughout the country, a vast network of natural gas pipelines is needed. There are 280,000 miles of natural gas transmission lines, on top of the approximately 1 million miles of distribution lines, in use in the United States [2,3]. With such large amounts of a combustible fuel being distributed throughout the nation, it is imperative for the pipelines to be in proper working order. Erosion of pipe-wall integrity may lead to an explosion causing costly property damage, loss of natural gas availability and even loss of life [4]. Hence, it becomes necessary to periodically inspect natural gas transmission pipelines for defects in order to perform preventative maintenance and repairs. This is accomplished using nondestructive evaluation (NDE) [5].

1.1. Nondestructive Evaluation
Nondestructive evaluation is the area of science and engineering that attempts to quantitatively characterize the integrity of materials without affecting the usefulness of those materials [6,7]. In this way, defects or variations within a material may be located and quantified; a piece of equipment or some other object may be post-production integrity-tested while still allowing the object to continue being utilized. Naturally, this has applications in many areas of industry such as airplane maintenance, manufacturing, pipeline inspection, the power industry, etc. Employing nondestructive evaluation
techniques may prevent accidents, ensure customer satisfaction, aid in product design, help control manufacturing processes, lower manufacturing costs and maintain a uniform quality level [6].

Nondestructive testing techniques operate by sending some type of energy signal into a test specimen and then measuring any changes to the energy signal. This is illustrated in Figure 1-1.

![Figure 1-1: Generalized illustration of NDE inspection techniques.](image)

Analysis of the measured output then aids in characterizing the specimen and drawing conclusions. Various signal processing techniques may be performed on the original output signal to enhance understanding of the state of the specimen.

There are a plethora of such NDE inspection techniques including optical, electromagnetic, ultrasonic, acoustic, thermal and X-ray techniques. This thesis deals with magnetic flux leakage (an electromagnetic technique) and ultrasonic testing, both of which are described in more detail in Chapter 2.

1.2. **Inverse Problem**

A NDE system may be represented as seen in Figure 1-2, where the excitation source, e(x), is convolved with the transfer function, h(x), to produce the probe measurement, f(x).
Figure 1-2: Representation of a NDE system.

This system layout presents three types of problems:

1. Forward problem – Given the input $e(x)$ and the transfer function, $h(x)$, find the probe measurement, $f(x)$.
2. System identification – Given the input $e(x)$ and the probe measurement, $f(x)$, find the transfer function, $h(x)$.
3. Inverse problem – Given the transfer function, $h(x)$, and the probe measurement, $f(x)$, find the input, $e(x)$.

The third problem type, the inverse problem, is also commonly known as deconvolution. If both the input, $e(x)$, and the transfer function, $h(x)$, are unknown while the output, $f(x)$, is known, the problem of finding the input and/or the transfer function is known as blind deconvolution [8,9,10,11].

Well-posedness was first introduced by Hadamard from his work with partial differential equations [10]. In accordance with Hadamard, a solution to an inverse problem can be considered well-posed if it possesses three properties:

1. Existence
2. Uniqueness
3. Continuous dependence of the output on the input

However, in general, inverse problems lack uniqueness and lack continuous dependence. Therefore, inverse problems are not well-posed, they are ill-posed [8,9,10,11].
1.3. **Virtual Reality**

Virtual reality (VR) is an advanced visualization system that also provides a human-machine interface (HMI) [12,13,14,15]. VR systems generate and display virtual worlds and enable a user to navigate through and interact with the computer generated artificial environment, as illustrated in Figure 1-3. Virtual reality possesses a number of unique advantages as a display and interaction device that makes it useful in a number of application areas, such as “education, training, high-level programming, teleoperation, remote planetary surface exploration, data analysis, scientific visualization”, virtual prototyping and entertainment [15]. Figure 1-4 through Figure 1-7 illustrate examples of several of the application areas just described.

![Figure 1-3: Interactions that create a virtual world.](image)
Figure 1-4: An example of advanced scientific visualization of weather data in virtual reality.

Figure 1-5: An example of a virtual world used for data analysis of pollution data.
Figure 1-6: An example of virtual reality for remote planetary surface exploration.
1.4. Thesis Overview

1.4.1. Motivation

Each year, a number of pipeline incidents occur as a result of accidents or vandalism. On December 23rd, 2000, a gas pipeline explosion in Philadelphia left approximately 1,000 customers with low gas pressure on a day that had temperatures only reaching into the twenties. Additionally, portions of I-76, a major highway, were closed off, delaying traffic for several hours [4].

Incidents such as the one in Philadelphia may be prevented with appropriate inspections of pipelines. The most prevalent method for in-line pipe inspection is with the use of a device known as a “pig,” seen in Figure 1-8. Most pigs use magnetic flux
leakage imaging, with some pigs using ultrasonic imaging. A MFL pig uses a permanent magnet to induce a current in a ferromagnetic pipe wall. An array of flux-sensitive sensors then measures the flux leakage. These measurements are stored on board the pig, where they may be downloaded to a computer after the inspection operation is complete. An analysis of the collected data will then help to identify defects within the pipeline.

Magnetic flux leakage NDE signals are the result of a static process described by elliptic differential equations; as such, the information content is lower compared to wave-based phenomena such as X-rays and ultrasound. The finite aperture of a Hall-effect probe which is used to measure the flux leakage “blurs” the received signal making the inverse problem of predicting the flaw characteristics even more challenging. This thesis attempts to address the signal inversion and defect characterization problem present in such low-information, static NDE phenomena, by explicitly defining geometric regions for information content. The motivation for this approach is that the defect characterization accuracy will increase when the goals of the signal inversion algorithm are set to match the reality of the physical process.

Figure 1-8: A MFL pig; an in-line gas pipeline inspection tool.
1.4.2. Objectives and Scope of the Thesis
The main objectives of this thesis are:

1. The development of an inversion algorithm for the prediction of information measures for signal/image inversion. Chapter 2 describes previous work in NDE signal inversion that lead up to the creation of this algorithm. The algorithm itself is developed in Chapter 3.

2. Validation of the inversion algorithm using an analytical approach with simple geometries, sources and sensors. This validation is detailed in Chapter 3.

3. Implementation of the inversion algorithm on experimentally obtained NDE signals. The results of the inversion algorithm are presented in Chapter 4.

4. Demonstration of the inversion results inside a virtual reality environment. The virtual world constructed to house the inversion results is illustrated with various screenshots in Chapter 4.

1.4.3. Expected Contributions
This thesis describes a method for performing signal/image inversion using the developed algorithm. The strength of the algorithm is shown both analytically and empirically through testing on experimental signals. The results of the algorithm are demonstrated inside a virtual reality environment; to accomplish this, a framework for the visualization of multiple sets of multi-dimensional data inside a virtual environment was developed. The development of the inversion algorithm will allow for a signal from a measured defect to be inverted to the defect geometry that the measurement was obtained from; that is, the original defect information will be deconvolved from the measurement. Visualizing the results of the inversion inside a virtual environment leverages the special
abilities of virtual reality to allow a user to more completely understand the available data through visual inspection in three dimensions. Finally, an analysis of results is performed and recommendations for future work are made.

1.4.4. Organization
Chapter 1 of this thesis provides an introduction to the problem and a description of the problem area. Chapter 2 provides background information on the key areas addressed in this thesis, as well as reviewing previous work performed in the problem areas of inversion and visualization. Chapter 3 details the approach taken in solving the problem and provides an analytical treatment of the inversion algorithm. Chapter 4 describes the process of obtaining experimental results with magnetic flux leakage and ultrasound NDE techniques, as well as the process of implementing the inversion algorithm on the experimental signals that were obtained. The results of the algorithm, as well as the measurement data, are also presented in this chapter. Chapter 5 discusses the results of the algorithm, conclusions that were reached as a result of the work performed for this thesis and finishes with recommendations for future work.
2. Background

This chapter presents a brief summary of previous work in the area of inverse problems, specifically with respect to nondestructive evaluation (NDE) applications. Radial basis function neural networks are also briefly discussed. An explanation of the NDE inspection techniques and technologies used in this research work is provided. Lastly, virtual reality technologies that are used for the advanced scientific visualization of research results are described.

2.1. Solution methods in NDE

The ill-posed nature of inverse problems has initiated much research into finding solutions to these problems. Several different methods are summarized in Table 2-1.

<table>
<thead>
<tr>
<th>Authors and Titles</th>
<th>Description of Work</th>
</tr>
</thead>
</table>
The following sections describe in greater detail the content of each of the titles listed in Table 2-1. Regularization, the classical approach to inverse problems is discussed first. This is followed by a discussion of iterative approaches to inverse problems. Finally, neural network approaches to inverse problems in NDE are discussed.

### 2.1.1. Regularization

Regularization was first introduced by Tikhonov and is considered a classical approach to obtaining a solution to inverse problems [10,11,16]. Regularization approaches inversion as a function approximation problem, where the function approximation is performed by minimizing a cost functional that includes an error term and a smoothness term. The smoothness term is usually known as a stabilizer and attempts to provide a smooth
solution to problems with sparse data. More to the point, regularization attempts to cure ill-posedness by incorporating a priori information in the form of smoothness. In this case, a smooth function is said to be one where two similar inputs correspond to two similar outputs. This constrains the solution space and allows the desired solution to be calculated. The solution to a regularization problem is then the function that minimizes the functional seen in equation 2.1:

$$H[f] = \sum_{i=1}^{N} [f(x_i) - y_i]^2 + \lambda \phi[f]$$  \hspace{2cm} (2.1)$$

where the first term imposes closeness to the data while the second term imposes smoothness. The smoothness term includes the smoothness functional $\phi[f]$ as well as the regularization parameter $\lambda$. Here, the regularization parameter provides an exchange between closeness to data and smoothness. Hence, the smoothness of the solution may be controlled by varying the second term of equation 2.1 [11].

It has been shown [11] that regularization techniques lead to a network approximation scheme with a single hidden layer. Furthermore if the functional, $\phi$, is radial, this leads to the "well-known" radial-basis functions [11]. In this way, approximations using radial basis function networks are directly related to regularization techniques.

2.1.2. Iterative Algorithms

One of the first methods of solving the magnetostatic inverse problem to gain widespread use was the calibration method, which performs a type of parameter estimation. In this simple solution, a set of calibration plots are generated from a test specimen by measuring characteristic features. The particular features vary with the inspection
method but may include features such as peak-to-peak voltage, peak voltage, etc. These measurements are then associated with several geometrical parameters such as depth, length, width, the angle of the defect with respect to the surface and, for sub-surface defects, the distance of the defect from the surface. The calibration curves are then generated by varying a single defect parameter while keeping all other parameters constant and recording the changes in measurements. This method has been employed for magnetic flux leakage measurements as well as eddy current signals [8].

One iterative approach used to solve the inverse problem performs a search for the optimal parameters. An example of this is a state space search of the solution space where each state describes the defect geometry. M. Yan et. al. [17, 18] describe the defect geometry using the finite element method. The objective of the algorithm is to minimize a given error function. If the search algorithm were allowed to exhaustively search the possible solutions states, the method would become inefficient and would not be useful. However, programming dynamic search techniques and incorporating a priori knowledge about the defect will speed up the search by limiting the solution space.

Using this method, the solution space may be efficiently searched for a state that adequately represents the solution to the inverse problem by meeting the error minimization goal set forth. M. Yan et. al. [17] also describe a method of iteratively updating the forward model by updating the finite element mesh. By updating the mesh in such a way as to minimize an error function, an optimal solution to the inverse problem may be found.

Genetic algorithms have also been used as an iterative search method for finding an optimal solution to inverse problems in NDE. In the genetic algorithm (GA)
approach, a population of potential solutions is randomly generated. Each individual in the population is then associated, through a fitness function, with a probability for reproduction. A higher probability for reproduction is the result of an individual being closer to the optimal solution. Individuals are then selected for reproduction and their children, or descendents, are formed based on a genetic operator. Two popular genetic operators are crossover and mutation. In crossover, a child is formed by combining features of the parents. In mutation, a feature from a selected individual is randomly altered. Through the genetic operators of crossover and mutation, successive generations tend toward an optimal solution.

Y. Li et. al. [19] describe using geometric parameters as the features of an individual solution in a genetic algorithm. This GA is used to search for the optimal solution for inverting eddy-current probe measurements. In particular the features used are as follows: the location of the left-most corner of a rectangular defect in the X direction, the location of the left-most corner of a rectangular defect in the Y direction, the length of the defect in the X direction, the width of the defect in the Y direction and the depth of the defect in the Z direction. These features are measured in terms of the number of block elements assigned to each. An alternative feature set-up used is to have each X-Y gridded block segment of the defect area become a feature, while each feature is measured in terms of the depth of the block segment. Using a specified feature setup, the genetic algorithm is then engaged to find an optimal solution; that is, the solution which most closely matches the true defect profile.
2.1.3. Neural networks
A natural extension of the finite element mesh forward model is described by P. Ramuhalli et al. [20]. Where as M. Yan et. al. [17] updated the forward model by numerically modifying the finite element mesh, P. Ramuhalli et. al. [20] use a specialized neural network to update the finite element mesh. Embedding finite element models into a neural network allows for the forward model to be updated much more quickly, providing for a faster solution to the inverse model.

Yet another improvement on the schemes described by P. Ramuhalli et. al. [20] and M. Yan et. al. [17] is to avoid the use of finite elements in the forward model, as described by P. Ramuhalli et. al. [21]. The use of function approximation neural networks allows for a signal to be directly estimated from the defect profile that was provided by the inverse model. That is, the function-approximation neural networks become the entire forward model. This has been successfully implemented using both radial-basis function neural networks and wavelet-basis function neural networks.

One final improvement on the previously described iterative and/or neural network algorithms is described by P. Ramuhalli et. al. [22]. This approach goes one step further than the one described by Ramuhalli et. al. [21] and uses a neural network for the inverse model. That is, the forward model described by Ramuhalli et. al. [21] remains in tact as a wavelet basis function neural network (WBF NN) while the numerical version of the inverse model is replaced by a neural network. This places the two neural networks in a feedback configuration. Successive iterations eventually yield a solution that meets error minimization criteria such that the forward model's signal matches the initial measured signal.
It should be noted that one thing M. Yan et. al. [17], P. Ramuhali et. al. [20], P. Ramuhalli et. al. [21] and P. Ramuhalli et. al. [22] all have in common is the assumption that if the measured signal resembles, in the least squares or other sense, the forward model’s signal, than the inverse model’s defect profile will resemble the true defect profile.

Hopfield neural networks have also been used to solve the electromagnetic inverse problem [23]. Hopfield neural networks converge to a minimum by minimizing an energy function. The inverse problem is then represented as a function minimization problem as described by integral equations. A Fredholm integral equation is used to describe the electromagnetic inverse problem where \( g(x) \) is the measurement, \( f(y) \) is the source and \( k(x,y) \) is the kernel and the objective is to solve for the source. This may be rewritten in matrix form:

\[
\begin{align*}
\int_{a}^{b} k(x,y) f(y) dy &= g(x) \\
FV &= g + n \\
Fy &= \int_{a}^{b} k(x,y) R_i(x) dx \\
g_i &= g(x_i) \\
V &= (v_1, v_2, \ldots, v_n)^T
\end{align*}
\] (2.2, 2.3, 2.4, 2.5, 2.6)

where \( R_i \) are basis functions chosen to be sine or cosine functions, \( n \) is noise and \( V \) is the solution.

A Hopfield neural network is employed to iteratively find a solution, \( V \), by minimizing a cost function:
\[ E = \frac{1}{2} (FV - g)^T (FV - g) + \lambda V^T DV + \lambda_i (\Psi V - f_p)^T (\Psi V - f_p) \] (2.7)

where \( E \) is the error, \( F, V \) and \( g \) are described in equations 2.2-2.6, \( \lambda \) and \( \lambda_i \) are Lagrange multipliers, \( D \) is the network weights, \( f_p \) is the system state at a given point \( p \) and \( \Psi \) is a diagonal matrix with elements that are basis functions.

A classification approach to inverse problems segregates solutions into a finite number of groups [8]. These groups may include anomalous signals as well as more benign signals. For example, in MFL testing of underground pipelines, benign signals are generated from the scanning of sleeves, welds, T-sections and assorted valves while anomalous signals are generated by stress corrosion cracking and pitting defects. This method has been successfully implemented on MFL signals obtained from pipelines using an artificial neural network algorithm with incremental learning abilities to perform pattern recognition-based automated signal classification [24].

Two different types of neural networks were used by J. Lee et. al. [25] to classify MFL signals. The first neural network used a single, traditional back-propagation architecture. All of the features extracted from the MFL measurement were input to the network as a vector. The second neural network scheme used a hierarchical multi-layer perceptron architecture in which the different features were split up as inputs to six different neural networks. That is, the input to each of the networks was a subset of the overall input vector that was provided to the single back-propagation network. Breaking the classification problem up using this hierarchical approach had the effect of reducing both Type I and Type II errors. However, the classification method does not reveal detailed geometric information about a defect area; it merely applies a label to it.
Neural networks have been used to obtain direct solutions to inverse problems [8]. The basic premise is that based upon examples shown to the network during training, a network will be able to successfully predict the output for a test input. That is, the fully trained neural network provides the complete inverse model. This is illustrated in Figure 2-1. Given MFL or some other type of measurements and the corresponding defect profiles as training, the network will be able to predict the defect profile for a given measurement during testing. Radial basis function (RBF) networks are adept at function approximation and interpolating a result from sparse input data [26,10,11]. These features offer the ability for RBF networks to provide solutions to inverse problems. Furthermore, using neural networks to directly obtain solutions to inverse problems has been successfully implemented in NDE applications [8]. Neural networks, specifically radial basis function networks, are described in more detail in the next section.

Figure 2-1: Outline of the direct neural approach to solving the inverse problem in NDE.

2.2. **RBF Neural Networks**
Artificial neural networks (ANN’s) are a type of artificial intelligence that attempts to mimic the human brain by using an architecture that is complex, non-linear and parallel. Neural networks use a training procedure to adapt themselves to prescribed input-output mappings by adjusting the weights between parallel layers of artificial neurons. This,
which provides the ability of ANN’s to possess non-linear mappings, makes them a powerful tool in a variety of areas, though not without limitations. The quality of the test outputs of a fully trained network will depend on the quality and quantity of the training data that was used to map out the synaptic weights during the training phase [26].

Radial basis function neural networks consist of three layers of nodes: an input layer, a single hidden layer and an output layer. Radial basis functions take a form seen in equation 2.8:

\[
f(x) = \sum_{i=1}^{N} \lambda_i \phi(\|x - x_i\|)
\]

(2.8)

where \(\|\|\) denotes a norm, usually a Euclidian norm, \(\lambda_i\) denotes the weights, \(x\) is the input and \(x_i\) are the basis centers.

For the case where Gaussian basis functions are used the equation may be rewritten as seen in equation 2.9:

\[
f(x) = \sum_{i=1}^{N} \lambda_i \exp \left(\frac{-\|x - x_i\|^2}{2\sigma_i^2}\right)
\]

(2.9)

where \(\sigma_i^2\) is the variance of the \(i^{th}\) Gaussian function. Training of the network is accomplished by estimating the parameters of the RBF, namely the variance, the basis centers and the weights. The variance can be assumed to be constant. The basis centers can be assigned random numbers or the basis centers can be calculated using a clustering algorithm such as the K-means clustering algorithm. The advantage of computing the centers using the K-means clustering algorithm is that the network will be able to converge to its error minimization goal more quickly, reducing the training time for the
network. This leaves the calculation of the weights as the real key to training the network. This is accomplished using equation 2.10:

\[ W = \Phi^{-1} X \]  

(2.10)

where \( W \) is the vector of weights, \( X \) is the vector of desired responses and \( \Phi^{-1} \) is the inverse matrix of basis function values. It has been shown that for a large class of radial-basis functions, the basis function matrix is indeed non-singular. Specifically, this has been shown to be true for Gaussian functions as long as the basis centers are distinct [26,8].

2.3. NDE Inspection Techniques

A variety of NDE inspection techniques exist that are commonly used today in various areas of industry and academia, such as magnetic, ultrasonic, acoustic, optical, X-ray and thermal techniques. These techniques are usually employed using an excitation source that is conveyed through a transducer and received, after interaction with the measurand, at the measurement transducer, with post-processing that is dependent upon the inspection technique. Of particular interest in this thesis are the magnetic flux leakage and ultrasonic techniques.

2.3.1. Magnetic Flux Leakage

Magnetic flux leakage (MFL) techniques have been used to inspect materials for many years. As early as 1934, the Magnaflux Corporation was founded as a supplier of inspection equipment, including magnetic testing equipment. MFL sensors were developed on the 1920’s and 30’s and began to allow for the quantitative measurement of magnetic fields around a defect. In 1965, the company Tuboscope released the first in-line pipeline inspection tool to measure MFL. Since then, the gas-transmission pipeline
industry has widely adopted MFL inspection techniques as the method of choice for in-line pipeline inspection [27].

MFL testing is basically a two step process. First, the material being tested is magnetized. Second, a flux-sensitive sensor is used to scan the magnetized material. In some cases, the magnetization is performed by applying a current directly to the ferromagnetic material being inspected. In other instances the magnetizing system may keep the magnetizing conductor in close proximity to the material under test, sometimes maintaining a slight air gap between the magnetization system and the test material. In the case of a discontinuity/defect in the test material, the discontinuity causes the magnetic flux to leak out of the material in an anomalous manner. This is then detected by the measurement scan and the location of the discontinuity is identified. Ideally, an increasing amplitude of flux leakage indicates an increasing defect size. However, it is very difficult to determine the depth and exact shape of a defect based solely on a MFL measurement. This is because the signal generated by the measurement of the flux leakage of a defect is caused by a plethora of factors including the size of the discontinuity (depth, length, width) and the orientation of the discontinuity to the magnetic activation field [6].

There are a number of different sensors that can be used to detect magnetic flux leakage, including coils, magnetodiodes, Förster microprobes and magnetic particle sensors. Hall element sensors may also be used. The main advantages of Hall element sensors are that they possess very small active areas and they can be configured such that amplitude measurements of the tangential or normal flux are not dependent upon the speed of the sensor. The main disadvantage of Hall element sensors is that in arrays of
multiple Hall element sensors, the sensors must be properly calibrated so that all sensors are electronically balanced [6].

2.3.2. Ultrasound
Ultrasonic testing uses high-frequency sound waves to perform testing and obtain measurements. Ultrasonic techniques are especially useful for discontinuity detection. As with MFL, ultrasonic testing techniques have existed for many years in various incarnations. As early as 1929, studies were being performed in the area of ultrasonics. In 1931, a patent was issued to Mulhauser for using two transducers to detect flaws in solids using ultrasonic waves. Through continued research, ultrasonic testing technology eventually reached the point at which it stands today [28].

Today, ultrasonic testing techniques are superior to other nondestructive inspection techniques in penetrating materials for the detection and measurement of subsurface defects. Ultrasonic testing is also an excellent method for inspecting surface flaws. Typically using frequencies between 1 and 25 MHz (though sometimes as low as 25 kHz and as high as several hundred GHz), two different methods may be employed. Through-transmission ultrasonic testing uses two transducers: a transmitter on one side of the specimen and a receiver on the opposite side of the specimen. The decrease of energy at the receiver allows for detection and measurement. This is illustrated in Figure 2-2.
The reflection or pulse echo ultrasonic testing method uses a single transducer for transmitting and receiving; hence the transmitter and receiver are located on the same side of the specimen. This is illustrated in Figure 2-3.

This method measures the energy that is reflected from the specimen under interrogation back to the receiver. The advantage of using a pulse echo configuration is that it only requires access to one side of the specimen under investigation [6].

Another important component of a UT system is the couplant that transfers ultrasonic energy first between the transmitter and the specimen and then between the specimen and the receiver. A couplant is necessary in order to effectively transmit energy. If there is air located between the transducers and the specimen, too much
energy will be reflected, resulting in a test with unclear or false indications. One of the main coupling techniques used is immersion. In immersion testing, a fluid is used to delay the pulses such that the reflection from the front or top of the test specimen is distinguishable, in the time domain, from the initial excitation pulse of the transducer. The most widely used method of immersion testing is performed by placing the test specimen in an immersion tank filled with the fluid couplant [6].

The fluid couplant most typically used in immersion tank ultrasonic testing is water, due to its availability and low cost. In using water, a difference is created between the velocity of the ultrasonic wave in the couplant and the velocity in the test object. For instance, in most metals, the velocity of the ultrasonic wave in the test specimen is four times faster than the velocity of the ultrasonic wave in water [6,29]. Water also increases beam divergence; the ultrasonic beam generated by the transducer spreads out more quickly as it propagates through water. This divergence effect may be counteracted by focusing the ultrasonic beam. Focusing may be accomplished by coupling a concave lens, often made of an epoxy material, to the transducer. This lens focuses the ultrasonic beam towards a point (or line, depending on the transducer). This is illustrated in Figure 2-4.

![Flat and Focused Transducer](image)

*Figure 2-4: Flat transducer and a focused transducer, which has the addition of a lens.*
The focal distance of the transducer is dependent upon the radius of the transducer and its frequency. Calculating a focal distance provides a minimum and maximum focal range for each transducer; these focal ranges are readily available in table format [29]. Positioning the transducer closer to the maximum focal range increases the penetration depth of the testing. Positioning the transducer closer to the minimum focal range increases the sensitivity of the testing towards smaller and smaller defects. Therefore, positioning the transducer midway between the minimum and maximum focal lengths provides the best trade-off between sensitivity to defects and depth of penetration.

2.4. Virtual Reality
Virtual reality is generally considered to have begun with Ivan Sutherland’s 1965 paper “The Ultimate Display” in which he described the ultimate display as being “a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal. With appropriate programming such a display could literally be the Wonderland into which Alice walked,” [30]. Sutherland went on to develop a monoscopic head-mounted display that was connected to a computer in 1966 and then created a head-mounted display that produced stereoscopic images in 1970 [14]. Eventually, the U.S, military went on to adopt virtual reality as a training and simulation technique. As technology has progressed through the decades, the feasibility of acquiring a useful VR system has increased to the point that today, many scientists and engineers have the ability to utilize VR as a visualization tool [14].
2.4.1. Components of a VR World
There are three main components that must be present when creating a virtual reality world: time, space and objects. In order to have an interesting and useful virtual world, all three of these components must be defined and must exist in some form. Virtual objects must exist in a certain place in a certain time, just like real-world objects.

Virtual reality is capable of displaying several different types of objects, which can be categorized into several main areas: raw data, object models and processed information. Raw data is data that has been collected through some type of measurement process and is then directly displayed. For example, to relate this definition to NDE, an ultrasonic scan of a material could be displayed in virtual reality as raw data. Object models represent objects as they would exist in reality, such as a chair or a pipe or a car. The key difference between object models and the other virtual objects is that they are constructed outside of the virtual world using a CAD-type application such as AutoCAD, SolidWorks or 3D StudioMax. In this way, the geometry of an object model is already defined. Information represents raw data that has been processed in some way. This may be simple data conditioning, like filtering, or a more complex process like a pattern recognition algorithm. For example, if raw data is processed through a neural network to produce some output that is displayed in virtual reality, this output would be processed information, as opposed to raw measurement data.

Raw and processed data may be represented as 1-, 2- or 3-dimensional grids. The dimensions (X, Y and Z) of such a grid may be completely independent of the other dimensions; X is independent of Y and Z, Y is independent of X and Z, and Z is independent of X and Y. On the other hand, the dimensions of a data grid may be completely dependent on all other dimensions; X depends on both Y and Z, Y depends on
both X and Z, and Z depends on X and Y. Alternatively, a grid dimension may only be dependent on one other dimension; X depends on Y but not Z, Y depends on Z but not X, etc. With these different combinations, it is possible to have a grid that, for example, has an independent X dimension, a Y dimension dependent on only X and a Z dimension dependent on both X and Y. In addition, varying the data through time adds a 4\textsuperscript{th} dimension and varying a property, such as color, of the data representation based upon a data variable may add a 5\textsuperscript{th} dimension [12].

Once any object is created, that object must be associated with a time and space definition so that it exists in virtual reality. A time definition may cause an object to only exist for a preset amount of time or it may define that object to exist for all time; either way, a time definition, whether explicit or implied, is required. The object must also be instantiated into a coordinate system so that it will exist in a specific space in virtual reality. After objects are created and placed in time and space, a user may begin navigating through this virtual world’s coordinate systems and observing the objects.

2.4.2. VR Hardware

There are three main classifications of VR platforms: fully immersive, semi-immersive and non-immersive. In order to understand the differences and abilities of each, it is necessary to have knowledge of the main components which comprise them.

The display screen (or screens) is an obviously important component of any VR system. Display screens may vary from a simple desktop CRT monitor to multi-wall projection systems to head-mounted displays. Any VR display must be capable of providing high frame rates in order to provide a realistic rendering and prevent skipping. The frame rate issue becomes especially important when stereoscopic 3D is enabled.
Stereoscopic 3D (stereo 3D) provides realistic depth perception by presenting the left eye and right eye of a user with two slightly different views. Stereo 3D may be enabled using LCD shutter glasses that are synchronized with the display device. LCD shutter glasses, such as CrystalEyes, place an LCD over each eye that filters out images. With the aid of such technology, stereo 3D allows the user to become more completely immersed in a virtual world.

A key feature of virtual reality is the ability to navigate in six degrees of freedom (6 DOF). That is, a user may translate through the X, Y or Z (horizontal, vertical or depth, respectively) dimensions as well as rotate around the X, Y or Z (pitch, roll or yaw, respectively) dimensions. In order to perform this type of navigation, a hand-tracked controller is used. VR controllers themselves are often of ordinary make, such as a joystick, game pad or mouse. It is the tracking of the controller’s position that allows for the user to intuitively navigate in 6 DOF in a virtual environment. A tracking system, such as Ascension’s Flock of Birds, tracks changes in six degrees of freedom (horizontal, vertical, depth, pitch, roll, and yaw) as a user moves a controller around. Additionally, attaching a 6 DOF tracker to the stereo-enabling glasses allows for head-tracking. This enables a VR system to adapt the display to the perspective of the user.

The display screen (or screens) is an obviously important component of any VR system. Display screens may vary from a simple desktop CRT monitor to multi-wall projection systems to head-mounted displays. Multi-wall projection systems, such as the FakeSpace CAVE, typically contain 3, 4 or 6 projection walls, each with its own projector. With this configuration, it is possible for a user to physically walk inside of the display area. This ability is a key characteristic of fully-immersive virtual reality systems.
in that such a display almost entirely shuts out the real world and fully immerses the user in the virtual world. Head-mounted displays also provide users with fully-immersive displays, and at a much lower cost. Large, single-wall projection screens are characteristic of semi-immersive systems in that they partially shut out the outside world but fail to completely immerse the user in the virtual environment. Non-immersive displays, such as simple desktop computer monitors, completely fail to shut out the outside world and do not immerse the user in the virtual world.

2.4.3. Reasons for Using VR
Virtual reality provides users with the ability to visualize multiple data sets of multi-dimensional data in stereoscopic 3D in a comprehensive, immersive environment with 6 DOF navigation. However, do these added features actually provide increased or faster understanding of data over visualizing the data in an everyday computer workstation? It has been shown by Ware and Franck [13] that for visualizing data in 3D, “head-coupled stereo viewing can increase the size of an abstract graph that can be understood by a factor of three; using stereo alone provided an increase by a factor of 1.6 and head coupling alone produced an increase by a factor of 2.2.” Ware and Franck also showed that structured movement in three dimensions improved understanding.

The experimental setup consisted of eighteen trials that were conducted for nine different visualization conditions; a list of the nine conditions follows:

1. 2D – no Z-axis information
2. Static perspective – same as 2D but with depth cues included
3. Stereo
4. Passive rotation, no stereo – the scene rotated at a constant velocity about a fixed axis
5. Stereo, passive rotation
6. Hand coupled movement, no stereo
7. Stereo, hand coupled movement
8. Head coupled perspective
9. Stereo, head coupled perspective

The scene used was a randomly generated graph with 75 nodes which were interconnected by 100 arcs. In this scene, unhighlighted nodes were drawn in a dark gray color while two highlighted nodes were drawn in red. The task of the test subjects was to determine if there was a path of arc length two between the two highlighted nodes or if there was no path. There was a 50% probability of each case occurring. Eleven subjects were used for this experiment, with eighteen trials for each subject. The experiment then recorded the accuracy of the subjects’ responses as well as response times.

The results of this experiment found the main difference between the nine conditions in the error rates, with a high of 26% for the 2D condition and a low of 6.1% for the stereo, hand coupled condition. Furthermore, the experiment showed that the combination of stereo and motion produced the lowest error rate, with an average error of 7.5%. This provides evidence that the benefits of virtual reality aid the user in obtaining a better understanding of the data being visualized.

2.5. Chapter Summary
This chapter described various solution methods for inverse problems in nondestructive evaluation. In particular, regularization, iterative algorithms and neural networks were
discussed. Regularization, being a classical approach, is a good starting point for solving basic inverse problems. However, for larger, more complex problems, such as those regularly encountered in NDE signals, a more powerful approach is needed. Iterative algorithms find an optimal solution to inverse problems by representing geometric parameters. These algorithms connect a forward model to an inverse model in a feedback configuration that iteratively converges towards some error goal. This goal describes the geometric parameters of the defect that corresponds to a measured signal.

This chapter also introduced the basic concepts behind radial basis function neural networks, a type of function approximator. The layers of interconnected nodes in a RBF network make for a massively connectionist, parallel, function approximator.

Several NDE inspection technologies were also described in this chapter. Magnetic flux leakage inspection provides a way for areas inaccessible to humans, such as underground pipelines, to be monitored. However, MFL measurements are inherently “blurry” and do not provide distinct geometric information about defects. Alternatively, ultrasonic inspection techniques are capable of providing both surface and sub-surface measurements of objects that reveal much more about the geometry of defects. Ultrasonic testing requires a physical connection between the ultrasonic transducer/receiver and the test specimen. In practice, this is often accomplished using water immersion in immersion tanks, making it difficult to perform in-line inspection with ultrasonic inspection technology.

Lastly, this chapter described VR technology. Objects present in a VR world must exist in time and space. When displayed on appropriate hardware, the stereoscopic
and navigable properties of VR provide an increased understanding of information over conventional display methods.
3. Approach
The main theoretical objectives for this thesis are revisited below:

1. The development of an inversion algorithm for the prediction of information measures for signal/image inversion.

2. Validation of the inversion algorithm using an analytical approach with simple geometries, sources and sensors.

Inversion algorithms applied to the nondestructive evaluation of components typically predict the size, shape and/or geometry of flaws or defects that are discovered in the test specimen. However, the information in the measured NDE signature that is related to the defect geometry is not present uniformly throughout the geometric extent of the signature. That is, there may be greater information content in some regions of the NDE image than in others, as related to the defect geometry. The image inversion methods that were described in Chapter 2 treat the entire NDE image uniformly. Neural-network based inversion methods, which operate on the basis of training data could be placed at a particular disadvantage when presented with test data, if all the input nodes are treated equally in terms of information content. This thesis attempts to recognize the inhomogeneous and anisotropic nature of the information present in the input NDE data that is being processed for inversion using appropriate data transformation methods. Defining specific regions of higher or lower information content allows the applied transformation to better extract useful information from raw measurements. This strategy is, in essence, a “divide-and-conquer” method. The approach for defining specific regions is described next.
3.1. **Definition of Regions**

The region of higher information content within a measurement is determined by comparing the initial measurement with a secondary measurement obtained using an alternate inspection method. The areas in which these two measurements corroborate are the areas of higher information content. The areas in which the two measurement levels differ are the areas of lower information content. In other terms, these areas of higher information can be said to represent the information that is redundant between the two measurements while the areas of lower information can be said to represent the information that is complementary [31, 32, 33]. This concept is illustrated in Figure 3-1.

![Diagram](image)

**Figure 3-1**: Illustration of a method for designating areas of information content for use in an inversion algorithm.

The information in the redundant portion is corroborated by two different measurement methods and so there is higher information content. The information in the complementary portion is not corroborated by a secondary measurement and so there is lower information content.

In implementation, the inversion algorithm utilizes geometric transformations to extract the higher and lower information regions. Geometric transformations use spatial
transformations and gray-level interpolation to reverse image distortion [32, 33]. In the case of the algorithm developed in this thesis, the geometric transformation is performed by the RBF neural networks. The overall approach taken for the "divide-and-conquer" method is illustrated in Figure 3-2. The MFL signature is separated into two regions which are separately inverted. The results of these two inversions are then recombined into a final solution.

3.2. The "divide-and-conquer" Strategy for NDE Signal/Image Inversion

Designating portions of the initial measurement as being of either higher or lower information content effectively breaks up the data into two separate pieces. These data may be inverted separately and then recombined after inversion. A comparison is made of analytical and radial basis function neural network signal inversion methods employing this strategy. A space-invariant, linear transformation is assumed for this purpose.

We begin with the inversion of a one-dimensional signal using conventional analytical and neural network methods:

\[ d(x) = e^{-bx^2} \] (3.1)
\[ h(x) = e^{-ax^2} \]  
\[ m(x) = d(x) * h(x) \]

where \( x \) defines the spatial dimension, \( d(x) \) is the geometry of the Gaussian-shaped defect, \( h(x) \) is the transfer function of the measurement probe and \( m(x) \) is the measured NDE signal, given by the linear convolution of \( d(x) \) and \( h(x) \). The process can be described in the spatial-frequency domain as

\[ \mathcal{F}[d(x)] = D(f) = \sqrt{\frac{\pi}{b}} e^{-\frac{\pi^2 f^2}{b}} \]

\[ \mathcal{F}[h(x)] = H(f) = \sqrt{\frac{\pi}{a}} e^{-\frac{\pi^2 f^2}{a}} \]

\[ \mathcal{F}[m(x)] = M(f) = D(f)H(f) \]

where \( f \) is the spatial frequency, \( \mathcal{F} \) is the Fourier transform, \( D(f), H(f) \) and \( M(f) \) are the Fourier transforms of \( d(x), h(x) \) and \( m(x) \) respectively. In order to recover the defect profile from the measurement a restored version of \( d(x), d_{res}(x) \), must be calculated:

\[ d_{res}(x) = \mathcal{F}^{-1}\left[ \frac{M(f)}{H(f)} \right] \]

where \( \mathcal{F}^{-1} \) is the inverse Fourier transform.

The inverse Fourier transform can estimated using a discrete space and spatial frequency approximation as

\[ d_{res}(x) = \frac{1}{N} \sum_{f=0}^{N-1} \left[ \frac{M(f)}{H(f)} \right] e^{\frac{2\pi j f x}{N}} \]

for \( x = 0, ..., N - 1 \).
Calculating $d_{res}(x)$ using a radial basis function neural network approximation yields another equation:

$$d_{res}(x) = \sum_{n=0}^{N-1} \lambda_n e^{-\frac{(x-x_n)^2}{2\sigma_n^2}}$$

(3.9)

where $\lambda_n$ are the weights, $\sigma_n$ is the variance of the $n_{th}$ basis function and $\parallel \parallel$ is some norm. For a Euclidian norm, equation 3.9 becomes:

$$d_{res}(x) = \sum_{n=0}^{N-1} \lambda_n e^{-\frac{(x-x_n)^2}{2\sigma_n^2}}$$

(3.10)

The equations 3.8 and 3.10 are functionally equivalent; they should yield the same result. The implementation of this one-dimensional numerical example may be seen in Figure 3-3.
Figure 3-3: A direct, one-dimensional numerical example illustrating the process of inversion.

(a): defect profile, \( d(x) \).
(b): defect profile in spatial frequency domain, \( D(f) \).
(c): transfer function, \( h(x) \).
(d): transfer function in spatial-frequency domain, \( H(f) \).
(e): measurement, \( m(x) \).
(f): measurement in spatial-frequency domain, \( M(f) \).
(g): inverse filter in the spatial-frequency domain, \( 1/H(f) \).
(h): inverse filter in the spatial domain.
(i): restored defect profile in spatial-frequency domain, \( D_{res}(f) \).
(j): restored defect profile, \( d_{res}(x) \).

This example plots the defect profile, \( d(x) \), in (a) the transfer function of the probe, \( h(x) \), in (c) and the probe’s measurement of the defect profile, \( m(x) \) in (e). Each of these is also plotted in the frequency domain, seen in (b), (d) and (f). The inverse filter is constructed in the frequency domain, seen in (g), and then brought into the spatial domain for plotting, seen in (h). Finally, \( D_{res}(f) \) is calculated, seen in (i) and then brought into the spatial domain, seen in (j). It may be seen that \( d(x) \) matches \( d_{res}(x) \) for this one-dimensional, numerical example.
The “divide-and-conquer” strategy that is developed in this thesis work is now illustrated with the same one-dimensional signal using both conventional analytical and neural network methods:

\[ H_{\text{inv}}(f) = H_{\text{inv1}}(f) + H_{\text{inv2}}(f) \]  \hspace{1cm} \text{(3.11)}

\[ H_{\text{inv1}}(f) = \frac{1}{\sqrt{\pi}} \frac{\sqrt{\alpha}}{e^\alpha} \bigg|_{f=k}^{0}, 0 \bigg|_{f=0} \]  \hspace{1cm} \text{(3.12)}

\[ H_{\text{inv2}}(f) = \frac{1}{\sqrt{\pi}} \frac{\sqrt{\alpha}}{e^\alpha} \bigg|_{f=0}^{k}, 0 \bigg|_{f=-k} \]  \hspace{1cm} \text{(3.13)}

In 3.11-3.13, the inverse filter is broken up into two separate portions. Each of these separate portions is then used to find a partial \( d_{\text{res}}(x) \):

\[ d_{\text{res1}}(x) = \frac{1}{N} \sum_{f=0}^{N-1} \left[ \frac{M(f)}{H_1(f)} \right] e^{\frac{2\pi i f x}{N}} \]  \hspace{1cm} \text{(3.14)}

\[ d_{\text{res2}}(x) = \frac{1}{N} \sum_{f=0}^{N-1} \left[ \frac{M(f)}{H_2(f)} \right] e^{\frac{2\pi i f x}{N}} \]  \hspace{1cm} \text{(3.15)}

From 3.8 and 3.10, we know that 3.14 and 3.15 are functionally equivalent to an RBF implementation:

\[ d_{\text{res1}}(x) = \sum_{n=0}^{N-1} \lambda_n e^{\frac{-(x-x_n)^2}{2\sigma_n^2}} \]  \hspace{1cm} \text{(3.16)}

\[ d_{\text{res2}}(x) = \sum_{n=0}^{N-1} \lambda_n e^{\frac{-(x-x_n)^2}{2\sigma_n^2}} \]  \hspace{1cm} \text{(3.17)}
That is, 3.14 is functionally equivalent to 3.16 and 3.15 is functionally equivalent to 3.17.

Once the separate $d_{res}(x)$ are obtained, they may be recombined to obtain a final $d_{res}(x)$.

$$d_{res}(x) = d_{res1}(x) + d_{res2}(x)$$

(3.18)

$d_{res}(x)$ in 3.18 is functionally equivalent to $d_{res}(x)$ in 3.8 and 3.10.

The implementation results are illustrated in Figure 3-4. Again, since the analytical and neural network estimations of the defect profile from the measured NDE signal are functionally equivalent, the results are correspondingly similar.

Figure 3-4: A “divide-and-conquer”, one-dimensional numerical example illustrating the process of inversion.
(a): central portion of the inverse filter, $H_{inv-central}(f)$.
(b): edge portion of the inverse filter, $H_{inv-edge}(f)$.
(c): central portion of the restored defect profile in the spatial-frequency domain, $D_{res-central}(f)$.
(d): edge portion of the restored defect profile in the spatial-frequency domain, $D_{res-edge}(f)$.
(e): central portion of the restored defect profile in the spatial domain, $d_{res-central}(x)$.
(f): edge portion of the restored defect profile in the spatial domain, $d_{res-edge}(x)$.
(g): recombination of edge and central portions to form restored defect profile through “divide-and-conquer method.
(h): comparison of direct and “divide-and-conquer” methods of obtaining a restored defect profile.
In the “divide-and-conquer”, one-dimensional example, the inverse filter is segregated into two separate portions, a central portion, seen in (a), and an edge portion, seen in (b). These inverse filters are then used to obtain two separate restored defect profiles, a central $d_{res}(x)$, seen in the spatial domain in (e) and the spatial frequency domain in (c), and an edge $d_{res}(x)$, seen in the spatial domain in (f) and the spatial frequency domain in (d). In the spatial domain, the central portion provides the basic shape and the edge portion provides higher detail. When recombined in the spatial domain into a single “divide-and-conquer” $d_{res}(x)$, seen in (g), the solution is the same as the direct $d_{res}(x)$, as compared in (h). This may be further extended through numerical example to show that appropriately trained RBF networks provide the same solution as direct inversion, where the input to the RBF network is the measurement, $m(x)$, and the output of the RBF network is the restored defect profile, $d_{res}(x)$. The difference between the training and testing inputs for the RBF networks is seen in Figure 3-5.
Figure 3-5: Training and test inputs extracted from $m(x)$ for use in RBF network implementations of the inverse filters.
After appropriate training, it may be seen that the RBF network test output was able to replicate the inverse filter, as seen in Figure 3-6. This illustrates that the output for the test case of the RBF network matches the $d_{res}(x)$ that was calculated using the standard inverse filter for this one-dimensional, direct numerical example. Figure 3-7 illustrates a recombined RBF test case output for a one-dimensional, "divide-and-conquer", numerical example.
A second numerical example is shown in Figure 3-8. In this example, the defect profile is a Gaussian with an added sinusoid. This results in the spatial-frequency domain representation of the defect profile that is seen in Figure 3-8-b. Notice the frequency spikes at -4 and 4 on the x-axis index, indicating the sinusoidal frequency. When the defect profile is reconstructed using the "divide-and-conquer" strategy, as seen in Figure 3-9, the sinusoid's frequency spikes fall into the edge inverse filter while the Gaussian function falls into the central inverse filter. These separate portions are then recombined to obtain a final solution, a restored defect profile.
Figure 3-8: A direct, one-dimensional numerical example illustrating the process of inversion. The defect profile is represented by a Gaussian function with an added sinusoid.
(a): defect profile, \( d(x) \).  (b): defect profile in spatial frequency domain, \( D(f) \).
(c): transfer function, \( h(x) \).  (d): transfer function in spatial-frequency domain, \( H(f) \).
(e): measurement, \( m(x) \).  (f): measurement in spatial-frequency domain, \( M(f) \).
(g): inverse filter in the spatial-frequency domain, \( 1/H(f) \).  (h): inverse filter in the spatial domain.
(i): restored defect profile in spatial-frequency domain, \( D_{\text{re}}(f) \).  (j) restored defect profile, \( d_{\text{re}}(x) \).
Figure 3-9: A “divide-and-conquer”, one-dimensional numerical example illustrating the process of inversion.

(a): central portion of the inverse filter, $H_{inv-central}(f)$.
(b): edge portion of the inverse filter, $H_{inv-edge}(f)$.
(c): central portion of the restored defect profile in the spatial-frequency domain, $D_{res-central}(f)$.
(d): edge portion of the restored defect profile in the spatial-frequency domain, $D_{res-edge}(f)$.
(e): central portion of the restored defect profile in the spatial domain, $d_{res-central}(x)$.
(f): edge portion of the restored defect profile in the spatial domain, $d_{res-edge}(x)$.
(g): recombination of edge and central portions to form restored defect profile through “divide-and-conquer” method.
(h): comparison of direct and “divide-and-conquer” methods of obtaining a restored defect profile.
Figure 3-10: A two-dimensional, direct example of inversion.

(a): defect profile, \(d(x,y)\).  
(b): defect profile in spatial frequency domain, \(D(f_x,f_y)\).
(c): transfer function, \(h(x,y)\).  
(d): transfer function in spatial-frequency domain, \(H(f_x,f_y)\).
(e): measurement, \(m(x,y)\).  
(f): measurement in spatial-frequency domain, \(M(f_x,f_y)\).
(g): inverse filter in the spatial-frequency domain, \(1/H(f_x,f_y)\).  
(h): inverse filter in the spatial domain.
(i): restored defect profile in spatial-frequency domain, \(D_{res}(f_x,f_y)\).  
(j) restored defect profile, \(d_{res}(x)\).

Figure 3-10 depicts a two-dimensional case of the direct inverse filter being employed. Plots of two-dimensional \(d(x,y)\), seen in (a), \(h(x,y)\), seen in (c) and \(m(x,y)\), seen in (e), along with their Fourier transforms \(D(f_x,f_y)\), seen in (b), \(H(f_x,f_y)\), seen in (d), and \(M(f_x,f_y)\), seen in (f), respectively. It may be seen that the
restored defect profile, \( d_{\text{res}}(x, y) \), seen in the spatial-frequency domain in (i) and the spatial domain in (j), matches the original defect profile \( d(x, y) \). This example is repeated, as seen in Figure 3-11, for a “divide-and-conquer” case of a two-dimensional example. As in the one-dimensional “divide-and-conquer”, the inverse filter was split into a central region, seen in (a) and an edge region, seen in (b). These lead to a central \( d_{\text{res}} \), seen in the spatial-frequency domain in (c) and the spatial domain in (e) and an edge \( d_{\text{res}} \), seen in the spatial-frequency domain in (d) and the spatial domain in (f). When these are recombined in the spatial domain, a final restored defect profile, seen in (g), is obtained through the “divide-and-conquer” method for a two-dimensional case.
Figure 3-11: A two-dimensional, “divide-and-conquer”, numerical example of inversion.
(a): central portion of the inverse filter, $H_{\text{inv-central}}(f_x f_y)$.
(b): edge portion of the inverse filter, $H_{\text{inv-edge}}(f_x f_y)$.
(c): central portion of the restored defect profile in the spatial-frequency domain, $D_{\text{res-central}}(f_x f_y)$.
(d): edge portion of the restored defect profile in the spatial-frequency domain, $D_{\text{res-edge}}(f_x f_y)$.
(e): central portion of the restored defect profile in the spatial domain, $d_{\text{res-central}}(x, y)$.
(f): edge portion of the restored defect profile in the spatial domain, $d_{\text{res-edge}}(x, y)$.
(g): recombination of edge and central portions to form restored defect profile through “divide-and-conquer method.
(h): comparison of direct and “divide-and-conquer” methods of obtaining a restored defect profile.

Since the approach developed in this thesis yields similar results for analytical and radial basis function neural network implementations for simple defect geometries, the result can be generalized for arbitrarily shaped defects. In this case analytical implementations are not feasible – we have to rely on the neural network method exclusively. Chapter 4 demonstrates the effectiveness of this method by implementing the
algorithm on experimentally obtained NDE signals of metallic specimens embedded with rectangular slot-shaped defects.
4. Results

The “divide-and-conquer” strategy for inverting NDE signals was described in Chapter 3 using one-dimensional and two-dimensional defects with simple geometries. It was shown that the radial basis neural network implementations yielded identical results to the analytical approach. This chapter extends this work and demonstrates the effectiveness of the technique for arbitrarily shaped defects using the RBF neural network. The test setup and procedure for obtaining NDE data from test specimens is described first. This chapter also presents the advanced visualization of algorithm results in virtual reality.

4.1. Test Specimen Suite

A suite of 4” x 6” test specimens was fabricated from ASTM 836 steel to simulate the walls of gas transmission pipelines. There are three groups of specimens, which are grouped according to the thickness of the specimen. The specimens represent pipe wall thicknesses of 5/16, 3/8, and 1/2 inches. Each of the three sets contains seven specimens with varying defect depths as well as a defect-free specimen. The six defect-containing specimens possess target defect depths of 0.3, 0.25, 0.2, 0.15, 0.10 and 0.05 inches.

Table 4-1 describes all test specimens. Figure 4-1 and Figure 4-2 show images of actual test specimens. Figure 4-1 shows a front view that illustrates the varying thickness of the test specimens. Figure 4-2 shows a top view of two specimens, one with a defect and one without a defect.
Table 4-1: Test specimens.

<table>
<thead>
<tr>
<th>Specimen #</th>
<th>Plate thickness (in)</th>
<th>Actual Defect Depth (in)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0.5</td>
<td>N/A</td>
</tr>
<tr>
<td>01</td>
<td>0.5</td>
<td>0.3005</td>
</tr>
<tr>
<td>01C</td>
<td>0.5</td>
<td>0.25</td>
</tr>
<tr>
<td>02</td>
<td>0.5</td>
<td>0.198</td>
</tr>
<tr>
<td>02C</td>
<td>0.5</td>
<td>0.15</td>
</tr>
<tr>
<td>03</td>
<td>0.5</td>
<td>0.0945</td>
</tr>
<tr>
<td>03C</td>
<td>0.5</td>
<td>0.05</td>
</tr>
<tr>
<td>10</td>
<td>0.375</td>
<td>N/A</td>
</tr>
<tr>
<td>11</td>
<td>0.375</td>
<td>0.298</td>
</tr>
<tr>
<td>11C</td>
<td>0.375</td>
<td>0.25</td>
</tr>
<tr>
<td>12</td>
<td>0.375</td>
<td>0.199</td>
</tr>
<tr>
<td>12C</td>
<td>0.375</td>
<td>0.15</td>
</tr>
<tr>
<td>13</td>
<td>0.375</td>
<td>0.1105</td>
</tr>
<tr>
<td>13C</td>
<td>0.375</td>
<td>0.05</td>
</tr>
<tr>
<td>20</td>
<td>0.3125</td>
<td>N/A</td>
</tr>
<tr>
<td>21</td>
<td>0.3125</td>
<td>0.303</td>
</tr>
<tr>
<td>21C</td>
<td>0.3125</td>
<td>0.25</td>
</tr>
<tr>
<td>22</td>
<td>0.3125</td>
<td>0.1955</td>
</tr>
<tr>
<td>22C</td>
<td>0.3125</td>
<td>0.15</td>
</tr>
<tr>
<td>23</td>
<td>0.3125</td>
<td>0.0995</td>
</tr>
<tr>
<td>23C</td>
<td>0.3125</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Figure 4-1: Side view of test specimens illustrating the difference in thickness for three specimens.
4.2. Test Setup

4.2.1. UT Test Setup

The ultrasonic inspection system that was employed to obtain UT measurements possesses an immersion tank, three degrees of freedom (translation through x, y and z dimensions), a detachable transducer and a PC with motor controllers, analog to digital converters (ADC’s) and appropriate software. A test specimen may be placed into the immersion tank, which is filled with tap water. Transducers with various diameters, angles and frequencies are available, but a 5 MHz, 0.75” diameter transducer at a perpendicular angle to the test specimen was utilized for testing. The inspection system used is shown in Figure 4-3.

Figure 4-2: A top view of two specimens. The specimen on the left has no defect while the specimen on the right has a simulated slot-shaped defect.
4.2.2. MFL Test Setup
The MFL test system uses a 200 A current to produce magnetic flux leakage in test specimens. This is accomplished by attaching leads from a power supply directly to the test specimen. The flux leakage may then be measured using the three Hall probes present in the system. Together, the three hall probes are capable of measuring the tangential X, tangential Y and tangential Z components of the MFL. The three motors found on the MFL test system allow for three degrees of freedom (translation through the X, Y and Z axes). The PC's software interacts with the motor controllers and ADC's to perform a complete MFL scan of a test specimen and store measurements. The MFL
inspection system used is shown in Figure 4-4 with a close-up of the specimen loading area in Figure 4-5.

Figure 4-4: MFL test stand setup.
4.3. Training and Test Data Sets

The “divide-and-conquer” neural network algorithm was implemented using two data sets. Each specimen was scanned twice using the MFL test setup and twice again using the UT test setup. These data sets were used to train the neural network, except for specific instances from each data set that were selected for use as test instances. The two data sets were combined into a single large set for the purpose of training and test. Table 4-2 and Table 4-3 show which instances were used for training and which instances were used for test. The actual MFL and UT scans that comprise the two data sets are shown in Figure 4-6 to Figure 4-9.
Table 4-2: Data set 1, training and test data. Test data is indicated by a gray highlight.

<table>
<thead>
<tr>
<th>Specimen #</th>
<th>Plate thickness (in)</th>
<th>Actual Defect Depth (in)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0.5</td>
<td>N/A</td>
</tr>
<tr>
<td>01C</td>
<td>0.5</td>
<td>0.03085</td>
</tr>
<tr>
<td>02</td>
<td>0.5</td>
<td>0.25</td>
</tr>
<tr>
<td>02C</td>
<td>0.5</td>
<td>0.198</td>
</tr>
<tr>
<td>03</td>
<td>0.5</td>
<td>0.0945</td>
</tr>
<tr>
<td>03C</td>
<td>0.5</td>
<td>0.05</td>
</tr>
<tr>
<td>10</td>
<td>0.375</td>
<td>N/A</td>
</tr>
<tr>
<td>11C</td>
<td>0.375</td>
<td>0.2984</td>
</tr>
<tr>
<td>12</td>
<td>0.375</td>
<td>0.25</td>
</tr>
<tr>
<td>12C</td>
<td>0.375</td>
<td>0.199</td>
</tr>
<tr>
<td>13</td>
<td>0.375</td>
<td>0.1105</td>
</tr>
<tr>
<td>13C</td>
<td>0.3761</td>
<td>0.095</td>
</tr>
<tr>
<td>20</td>
<td>0.3125</td>
<td>N/A</td>
</tr>
<tr>
<td>21</td>
<td>0.3125</td>
<td>0.303</td>
</tr>
<tr>
<td>21C</td>
<td>0.3125</td>
<td>0.25</td>
</tr>
<tr>
<td>22</td>
<td>0.3125</td>
<td>0.1955</td>
</tr>
<tr>
<td>22C</td>
<td>0.3125</td>
<td>0.18</td>
</tr>
<tr>
<td>23</td>
<td>0.3125</td>
<td>0.0995</td>
</tr>
<tr>
<td>23C</td>
<td>0.3125</td>
<td>0.05</td>
</tr>
</tbody>
</table>
Table 4-3: Data set 2, training and test data. Test data is indicated by a gray highlight.

<table>
<thead>
<tr>
<th>Specimen #</th>
<th>Plate thickness (in)</th>
<th>Actual Defect Depth (in)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0.5</td>
<td>N/A</td>
</tr>
<tr>
<td>01</td>
<td>0.5</td>
<td>0.3005</td>
</tr>
<tr>
<td>01C</td>
<td>0.5</td>
<td>0.25</td>
</tr>
<tr>
<td>02</td>
<td>0.5</td>
<td>0.198</td>
</tr>
<tr>
<td>02C</td>
<td>0.5</td>
<td>0.15</td>
</tr>
<tr>
<td>03</td>
<td>0.5</td>
<td>0.0945</td>
</tr>
<tr>
<td>03C</td>
<td>0.5</td>
<td>0.05</td>
</tr>
<tr>
<td>10</td>
<td>0.375</td>
<td>N/A</td>
</tr>
<tr>
<td>11</td>
<td>0.375</td>
<td>0.298</td>
</tr>
<tr>
<td>11C</td>
<td>0.375</td>
<td>0.25</td>
</tr>
<tr>
<td>12</td>
<td>0.375</td>
<td>0.199</td>
</tr>
<tr>
<td>12C</td>
<td>0.375</td>
<td>0.15</td>
</tr>
<tr>
<td>13</td>
<td>0.375</td>
<td>0.1105</td>
</tr>
<tr>
<td>13C</td>
<td>0.375</td>
<td>0.05</td>
</tr>
<tr>
<td>20</td>
<td>0.3125</td>
<td>N/A</td>
</tr>
<tr>
<td>21</td>
<td>0.3125</td>
<td>0.303</td>
</tr>
<tr>
<td>21C</td>
<td>0.3125</td>
<td>0.25</td>
</tr>
<tr>
<td>22</td>
<td>0.3125</td>
<td>0.1955</td>
</tr>
<tr>
<td>22C</td>
<td>0.3125</td>
<td>0.15</td>
</tr>
<tr>
<td>23</td>
<td>0.3125</td>
<td>0.0995</td>
</tr>
<tr>
<td>23C</td>
<td>0.3125</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Figure 4-6: UT data set 1.
Figure 4-7: MFL data set 1.

Figure 4-8: UT data set 2.

Figure 4-9: MFL data set 2.
4.4. **Training and Test Results**

The results of the training and test phases of neural network implementation are displayed in this section. Each figure is composed of seven images: UT, MFL, desired HC, desired LC, HC result, LC result and recombin ed result. The UT and MFL images are the measurements obtained from UT and MFL inspections. The desired HC and LC images were used to train the neural networks. The HC and LC result images are the images that the neural network actually produces during simulation. The recombin ed result is the final output after the HC and LC results are combined. For training instances, it is desirable for the desired and result images to match. For test instances, those figures bordered in red, it is desirable for the recombin ed result to match the geometry of the test specimen.
Desired HC Dosimetry in HCF is compared to LC.

(d)
(s)
Figure 4-10: Data set 1.

(a): Specimen # 00 - 0.5” thickness, no defect; training data.
(b): Specimen # 01 - 0.5” thickness, 0.3” defect depth; test data.
(c): Specimen # 01C - 0.5” thickness, 0.25” defect depth; training data.
(d): Specimen # 02 - 0.5” thickness, 0.20” defect depth; training data.
(e): Specimen # 02C - 0.5” thickness, 0.15” defect depth; training data.
(f): Specimen # 03 - 0.5” thickness, 0.10” defect depth; training data.
(g): Specimen # 03C - 0.5” thickness, 0.05” defect depth; training data.
(h): Specimen # 10 - 0.375” thickness, no defect; training data.
(i): Specimen # 11 - 0.375” thickness, 0.30” defect depth, test data.
(j): Specimen # 11C - 0.375” thickness, 0.25” defect depth, training data.
(k): Specimen # 12 - 0.375” thickness, 0.20” defect depth, training data.
(l): Specimen # 12C - 0.375” thickness, 0.15” defect depth, training data.
(m): Specimen # 13 - 0.375” thickness, 0.10” defect depth, training data.
(n): Specimen # 13C - 0.375” thickness, 0.05” defect depth, test data.
(o): Specimen # 20 - 0.3125” thickness, no defect, test data.
(p): Specimen # 21 - 0.3125” thickness, 0.30” defect depth, training data.
(q): Specimen # 21C - 0.3125” thickness, 0.25” defect depth, training data.
(r): Specimen # 22 - 0.3125” thickness, 0.20” defect depth, training data.
(s): Specimen # 22C - 0.3125” thickness, 0.15” defect depth, test data.
(t): Specimen # 23 - 0.3125” thickness, 0.10” defect depth, training data.
(u): Specimen # 23C - 0.3125” thickness, 0.05” defect depth, training data.
(c)
(d)
Desired LC
Recombined Result
LC Result
MFL
Desired HC
Desired LC
HfC Fsut Racarbned ReslAt LC ReFat
(i)
MFL
Desired HC
Desired LC
HCF Rsult
Recombined Result
LC Result

(i)
Figure 4-11: Data set 2.

(a): Specimen # 00 – 0.5” thickness, no defect; training data.
(b): Specimen # 01 – 0.5” thickness, 0.3” defect depth; training data.
(c): Specimen # 01C – 0.5” thickness, 0.25” defect depth; training data.
(d): Specimen # 02 – 0.5” thickness, 0.20” defect depth; training data.
(e): Specimen # 02C – 0.5” thickness, 0.15” defect depth; training data.
(f): Specimen # 03 – 0.5” thickness, 0.10” defect depth; training data.
(g): Specimen # 03C – 0.5” thickness, 0.05” defect depth; training data.
(h): Specimen # 10 – 0.375” thickness, no defect; training data.
(i): Specimen # 11 – 0.375” thickness, 0.30” defect depth, training data.
(j): Specimen # 11C – 0.375” thickness, 0.25” defect depth, training data.
(k): Specimen # 12 – 0.375” thickness, 0.20” defect depth, training data.
(l): Specimen # 12C – 0.375” thickness, 0.15” defect depth, training data.
(m): Specimen # 13 – 0.375” thickness, 0.10” defect depth, training data.
(n): Specimen # 13C – 0.375” thickness, 0.05” defect depth, training data.
(o): Specimen # 20 – 0.3125” thickness, no defect, training data.
(p): Specimen # 21 – 0.3125” thickness, 0.30” defect depth, test data.
(q): Specimen # 21C – 0.3125” thickness, 0.25” defect depth, test data.
(r): Specimen # 22 – 0.3125” thickness, 0.20” defect depth, test data.
(s): Specimen # 22C – 0.3125” thickness, 0.15” defect depth, training data.
(t): Specimen # 23 – 0.3125” thickness, 0.10” defect depth, test data.
(u): Specimen # 23C – 0.3125” thickness, 0.05” defect depth, training data.
Figure 4-12: Training and test results for data set 1. Test instances are labeled as such; all other instances are used for training.
4.5. **Visualization of Results**

It has been established that VR techniques provide users with an increased ability to perceive the information that is present in a visualization. To accomplish this increased information perception through VR, a general framework for the advanced visualization of NDE images was created. This framework incorporates graphical, measurement and functional data, as seen in Figure 4-14.
Figure 4-14: Advanced visualization of the input, outputs, measurements and test specimen used in creating, training and implementing the “divide-and-conquer” neural network inversion algorithm.

Each of the six individual sections visible in Figure 4-14 is seen in higher resolution in Figure 4-15 through Figure 4-20. Figure 4-15 shows the MFL measurement visualized in a virtual world. This visualization helps to accentuate the point that it is difficult to determine geometric information about a defect from a raw MFL measurement. This is especially evident when the MFL visualization is compared to the neighboring graphical model of the test specimen from which the MFL measurement was obtained, Figure 4-16. A digital image of the test specimen was used to texture the graphical model, adding a
level or realism. Figure 4-17 shows the virtual instantiation of the UT measurement associated with the test specimen. This visualization serves to accentuate the fact that while MFL inspection techniques are inherently "blurry", UT inspection techniques provide accurate geometric information about the test specimen. Figure 4-18 and Figure 4-20 visualize the results of the lower content and higher content neural networks, respectively. This provides the ability to examine the way in which the "divide-and-conquer" neural inversion algorithm separated information. Finally, Figure 4-19 visualizes the recombination of the higher and lower content portions into the final result of the algorithm, which represents the algorithm’s prediction of the geometry of the test specimen.
Figure 4-15: Advanced visualization of the MFL inspection image used in this data instance.
Figure 4-16: Advanced visualization of a graphical model of the test specimen from which the MFL and UT measurements were taken. A digital image of the actual test specimen was used to texture the graphical model.
Figure 4-17: Advanced visualization of the UT inspection image used in this data instance.
Figure 4-18: Advanced visualization of the result of the lower content network for this data instance.
Figure 4-19: Advanced visualization of the result of the recombination of the higher and lower content network outputs for this data instance.
4.6. Discussion of Results

It may be seen that there is significant agreement between the desired output and the actual result for training instances for both HC and LC networks. This is a good indication that the training portion of the neural network construction was successful; the resultant matrices are non-singular. Mean squared error (MSE) was calculated for HC, LC and recombined results for both training and test data. For HC and LC portions, the MSE was calculated using the desired and actual outputs of the RBF networks. For the recombined results, the MSE was calculated using the recombined outputs and the appropriate template. The templates describe an optimal shape and depth for each specimen; in essence, they describe the defect profile. Figure 4-21 shows the MSE
calculated for training data for recombined, HC and LC data. It is expected that the MSE would be lower for the HC and LC training data because the network is specifically trained to approximate these values. Furthermore, it may be seen that there is also considerable agreement between most of the test outputs and the corresponding desired outputs. Test instances from specimen #01 - data set 1 (Figure 4-10-b) and specimen 22 - data set 2 (Figure 4-11-r) produced outputs that were slightly off from the desired output. This may be attributed to the fact that increasing the number of test instances has left the network with fewer training instances. However, even these two test instances are within one height step (approximately 0.05") of the desired output. This indicates that the goal of the neural network, to perform the function of an inverse algorithm, has been reached.

The neural network is successfully able to invert a MFL measurement into the defect profile that the measurement was obtained from. This is quantified in Figure 4-22, which shows the MSE for recombined, HC and LC test data. For test data, it may be seen that the recombined data has the lowest MSE. This is a positive indication because it suggests that the final reconstructed defect profile closely resembles the geometry of the test specimen. Furthermore, the difference in MSE for the recombined test results as compared to the individual HC and LC test results is explained by observing that the HC and LC portions for test data tend to overlap. This causes both the HC and LC test results to separately have higher error; when recombined the error is reduced.
Figure 4-21: MSE for training data.

Figure 4-22: MSE for test data.
Conclusions
Signal inversion in nondestructive evaluation applications is a critical step before remediation decisions are made. The accuracy of the signal inversion results therefore play a key role in deciding the effectiveness of the NDE procedure. Conventional NDE signal inversion algorithms are hampered by the fact that they treat all geometric regions of the NDE signal equally. This thesis has shown an alternative method for NDE signal inversion. Different geometric regions of the NDE signature are ascribed with different content levels; separate neural network inversion algorithms are applied to each region and the results are combined. It is shown that this “divide-and-conquer” strategy yields robust results, especially when applied to test data that the neural network has not seen before. While the algorithm is exercised theoretically using simple 1-D and 2-D defect geometries, the technique is also validated using NDE inspection images from a suite of test-specimens representative of the in-line inspection of gas transmission pipelines.

5.1. Summary of Accomplishments
The principal contributions of this thesis are listed along with the set of original objectives below:

1. The development of an inversion algorithm for the prediction of information measures for signal/image inversion. A “divide-and-conquer” strategy using RBF neural networks was developed for signal/image inversion. The NDE signatures were segmented into regions of varying information content and geometric transformations were used to predict the corresponding defect profile.

2. Validation of the inversion algorithm using an analytical approach with simple geometries, sources and sensors. 1- and 2-dimensional Gaussian functions were
used to represent the defect geometry and the transducer transfer function. A linear, space-invariant response was assumed and the measured signal was calculated. Analytical inverse filters were employed for the deconvolution of the defect geometry from the measured response – the results were compared to show agreement with those obtained from exercising the neural network algorithm on the same signals.

3. **Implementation of the inversion algorithm on experimentally obtained NDE signals.** Magnetic flux leakage and ultrasonic NDE inspections were conducted on a set of 21 steel test-specimens varying in thickness and embedded with slot-shaped defects that vary in depth from 0.3” to 0” in steps of 0.05”. These signatures were used in the implementation of the inversion algorithm. These results show that the “divide-and-conquer” strategy to neural inversion of NDE signals/images successfully fulfills its intended purpose, especially when presented with appropriate training data.

4. **Demonstration of the inversion results inside a virtual reality environment.**

Placing the inversion results in a virtual environment allows the user to freely navigate and observe them. Including the experimentally obtained NDE images, the divided results, the recombined result and even a model of the original test specimen further allows the user to comprehend the entire inversion process.

**5.2. Directions for Future Work**

A significant amount of progress was achieved in the advancement of the development of an inversion algorithm for NDE signals/images in this thesis. Many concerns and issues still require attention to develop more versatile and comprehensive NDE inversion
methods. The objectives of this thesis have been met as evident from the results presented, but some amount of research still must be applied before this “divide-and-conquer” inversion strategy can be used at industry level. Future developments of this research should include:

1. Additional experimentally obtained NDE signals/images for further validation of the algorithm. This should include additional defect shapes.

2. In order to truly test the “divide-and-conquer” inversion algorithm, actual pipeline data must be obtained or collected for use in the algorithm. This will validate the ability of the algorithm to perform on “real world” data.

3. Apply the inversion algorithm to applications outside the NDE realm in order to determine its versatility. Areas of science such as image analysis/restoration and geophysics regularly rely on inversion algorithms to produce results.

4. Further development of the virtual reality platform is needed to accommodate “real world” data. This becomes especially important because of the vast amounts of data generated from NDE in-line inspection of gas transmission pipelines.

The integration of neural network inversion and advanced visualization techniques holds considerable promise to revolutionize the way in which NDE procedures are implemented in the future. This thesis has demonstrated the possibility of this integration using gas transmission pipeline inspections as a test-platform. However, the technique is sufficiently general; there is no doubt that it is portable to a variety of other NDE applications.
References


