

T-4692

TEM Data Transformation and Recognition

by

Brian E. Mielke

ProQuest Number: 10794098

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10794098

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Geophysics).

Golden, Colorado

Date Nov. 10, 1994

Signed: Brian E Mielke
Brian E. Mielke

Approved: Catherine K. Skokan
Dr. Catherine K. Skokan
Thesis Advisor

Golden, Colorado

Date 10 Nov 94

PR Romig
Dr. Phillip R. Romig
Professor and Head,
Department of Geophysics

Abstract

The usefulness of the time domain, or transient electromagnetic (TEM) method in characterizing complex two- or three-dimensional structures has been limited by a lack of simple and efficient methods of analyzing the data. This comes not from lack of effort, but from the complexity of the data generated. While the information about the actual structure may be contained in the data, a practical method is needed to evaluate it. Fourier transforms can be applied to the data of complex structures, transforming them into characteristic patterns that can then be recognized through the use of trained neural networks. The network could be used alone for analysis or to provide an initial model for a modeling/inversion program.

As an example of a possible form of such a pattern recognition paradigm, a series of two-dimensional TEM modeled data of dikes of varying depths and angle of dip was created, transformed and classified by a neural network. The dikes represented a feature that could not be casually characterized from the data alone, but had one or two

attributes of importance to a geophysical interpreter. The results demonstrate promise for the method, as the network not only learned the training patterns, but was also capable of interpolation, and was not strongly affected by noise. The network and transform programs are sufficiently simple that eventually a system could be created which a field operator could use with reasonable success.

Table of Contents

Abstract	iii
List of Figures	vi
List of Tables	ix
Acknowledgments	xi
1 Introduction	1
2 Background	6
2.1 TEM Basics	6
2.2 Fourier Transforms	15
2.3 Neural Networks	18
3 Methods	27
3.1 Dike Models	27
3.2 Fourier Transformation Routine	36
3.3 Neural Networks	43
4 Results	48
4.1 Noise-free Results	48
4.2 Frechet Derivative Analysis	52
4.3 Noisy Data Results	57
4.4 Results of Other Models	61
5 Conclusions	62
References Cited	65
A Noisy Data Results	68
B Fourier Time Transforms	89
C Fourier Time and Space Transforms	99
D Program Listings	109

List of Figures

2.1	Field due to dipole oriented in positive x-direction	10
2.2	TEM receiver data from dike model	13
2.3	Dike model TEM data for 20 time points and 31 receivers	14
2.4	A sample node	18
2.5	Typical network layout	19
2.6	Sigmoidal transfer function	22
2.7	Example of a tessellated network	24
3.1	Example finite difference grid	29
3.2	Diagram of dike model	33
3.3	Schematic of Slab Model	35
3.4	Schematic of Diamond Model	35
3.5	Temporal transform for 60 degree dike model	41
3.6	Spatial and temporal transform for 60 degree dike model	42
3.7	Spatial and temporal transform normalized for neural net input for 60 degree dike model	46
3.8	Spatial and temporal transform normalized and clipped for neural net input for 60 degree dike model	47
4.1	Frechet derivative for angles 20 and 15 @ 8 m depth	53
4.2	Frechet derivative for angles 20 and 15 @ 0 m depth	54

4.3	Frechet Derivative for angles 45 and 50 @ 8 m depth	55
B.1	Temporal transform for 30 degree dike model @ 0 m depth	90
B.2	Temporal transform for 60 degree dike model @ 0 m depth	91
B.3	Temporal transform for 90 degree dike model @ 0 m depth	92
B.4	Temporal transform for 30 degree dike model @ 8 m depth	93
B.5	Temporal transform for 60 degree dike model @ 8 m depth	94
B.6	Temporal transform for 90 degree dike model @ 8 m depth	95
B.7	Temporal transform for 30 degree dike model @ 20 m depth	96
B.8	Temporal transform for 60 degree dike model @ 20 m depth	97
B.9	Temporal transform for 90 degree dike model @ 20 m depth	98
C.1	Spatial and temporal transform for 30 degree dike model @ 0 m depth	100
C.2	Spatial and temporal transform for 60 degree dike model @ 0 m depth	101
C.3	Spatial and temporal transform for 90 degree dike model @ 0 m depth	102
C.4	Spatial and temporal transform for 30 degree dike model @ 8 m depth	103

C.5	Spatial and temporal transform for 60 degree dike model @ 8 m depth	104
C.6	Spatial and temporal transform for 90 degree dike model @ 8 m depth	105
C.7	Spatial and temporal transform for 30 degree dike model @ 20 m depth	106
C.8	Spatial and temporal transform for 60 degree dike model @ 20 m depth	107
C.9	Spatial and temporal transform for 90 degree dike model @ 20 m depth	108

List of Tables

3.1	Parameters for finite difference grid	30
3.2	Modeled time points	31
4.1	Results of noise-free non-clipped test data	50
4.2	Results of noise-free clipped test data	51
4.3	Results of $\pm 5\%$ noise non-clipped data	59
4.4	Results of $\pm 5\%$ noise clipped data	60
A.1	Results of $\pm 5\%$ noise clipped training data from noise-free trained network	69
A.2	Results of $\pm 5\%$ noise clipped test data from $\pm 5\%$ noise trained network	70
A.3	Results of $\pm 10\%$ noise clipped training data from noise-free trained network	71
A.4	Results of $\pm 10\%$ noise clipped test data from noise-free trained network	72
A.5	Results of $\pm 10\%$ noise clipped training data from $\pm 5\%$ noise trained network	73
A.6	Results of $\pm 10\%$ noise clipped test data from $\pm 5\%$ noise trained network	74
A.7	Results of $\pm 10\%$ noise clipped test data from $\pm 10\%$ noise trained network	75
A.8	Results of $\pm 5\%$ noise clipped training data from $\pm 10\%$ noise trained network	76
A.9	Results of $\pm 5\%$ noise clipped test data from $\pm 10\%$ noise trained network	77

A.10	Results of noise-free clipped training data from $\pm 10\%$ noise train network	78
A.11	Results of noise-free clipped test data from $\pm 10\%$ noise trained network	79
A.12	Results of $\pm 5\%$ noise non-clipped training data from noise-free trained network	80
A.13	Results of $\pm 5\%$ noise non-clipped test data from $\pm 5\%$ noise trained network	81
A.14	Results of $\pm 10\%$ noise non-clipped training data from noise-free trained network	82
A.15	Results of $\pm 10\%$ noise non-clipped test data from noise-free trained network	83
A.16	Results of $\pm 10\%$ noise non-clipped training data from $\pm 5\%$ noise trained network	84
A.17	Results of $\pm 10\%$ noise non-clipped test data from $\pm 5\%$ noise trained network	85
A.18	Results of $\pm 10\%$ noise non-clipped test data from $\pm 10\%$ noise trained network	86
A.19	Results of $\pm 5\%$ noise non-clipped training data from $\pm 10\%$ noise trained network	87
A.20	Results of $\pm 5\%$ noise non-clipped test data from $\pm 10\%$ noise trained network	88

Acknowledgments

I would like to thank Cathy Pfeifer and INEL for their support of this work. Of course, I cannot forget Dr. H. T. Andersen who was most helpful as my ex-advisor.

Chapter 1

Introduction

The time-domain or transient electromagnetic method (TEM) has historically been used in anomaly finding and layered earth model modes. The reason for this is quite basic. The mode of electromagnetic transmission from source to receiver is a combination of diffusion through the earth, induction between the source, conductive bodies and the receiver, and self induction of the conductive bodies all combine in ways not necessarily linear; see Ward and Hohmann (1987), Hansen et al. (1967), Aktarakci (1993), Kaufman (1992), and Telford, Geldart, and Sheriff (1990). There are no simple means of analyzing the data. Most schemes capable of such work require dense data and more than one component of the magnetic and/or electric fields to be measured.

Modes of data collection vary from sparse to dense time and spatial data points, and one to five components of the electric and magnetic fields. The gathered data set is tailored towards a particular use. The data set can be large or small, and in the time domain difficult to directly

interpret. However, if the data are viewed entirely in frequency space the patterns are characteristic of the geology of the surveyed area. Further, large data sets may be reduced by choosing only the important frequencies in the patterns. It will be shown that these characteristic patterns may then be recognized by automated means, namely through the use of a neural network.

Neural networks have proven their ability to recognize patterns in many areas, such as seismic first break picks, spectral analysis, and sonar detection; see Elkington (1991), McCormack, Zaucha and Dushek (1993), Williams (1991), and Leighton (1992). The key to their success is to have a distinct set of patterns that can be classified through a series of weighted decisions. For complex pattern recognition purposes, about the only way to determine if you have such a data set is to attempt to train a neural network on the set. So in order for a network to be able to classify TEM data it must be amenable to such classification. A few attempts have been already made to do so. Poulton (1992) has had success; however her data was acquired by a custom 3-axis frequency system and tested on essentially one-dimensional or point type targets rather than two- or three-dimensional targets containing

identifiable structures. Other recent proposals have similar targets and use frequency methods (Cisar et al.).

It may be debatable as to whether TEM data is of a type conducive to neural network interpretation. The complicated interactions associated with the method lead to certain nuances in the data. There are positions of source, target, and receiver where coupling might be poor, and thus information about the target can be limited as demonstrated by Gallagher et al. (1985). The physics of the method often leads to ill-posed problems, where there is no hope to resolve certain features, cf. Kaufman (1992). With care, however, problems may be chosen and executed, such that success is possible. Certainly, of the electrical and electromagnetic methods, TEM will often contain the most amount of characterizing data, albeit in a somewhat intractable form.

Once it is accepted that TEM data contains enough information to be specific about features of a target, such as the depth and dip of a dike, a method is needed to accentuate the individuality of the data derived. This is the reason for performing Fourier transforms on the data. The frequency character, in both time and space, creates patterns in which the features may be easier to recognize

and be more characteristic. The use of the transforms may also remove the effect of the spacing interval between receiver stations and time points, allowing for consistent patterns from data of different time and receiver spacing, but of the same target. Unfortunately the time data collected with some TEM systems does not lend itself well to being transformed into frequency domain. This is due to the lack of enough time data points. A Fourier transform algorithm that allows irregularly sampled data to overcome insufficient, exponentially spaced time points and, choosing of the desired frequencies to provide consistent patterns is used.

Once suitable patterns are available they are used to train a neural network. If the network can be trained it is hoped the network can interpolate between the models in the training data set and produce reliable results for the parameters falling between or beyond the parameters included in the training set. This was the goal of this project; having created recognizable patterns from typical TEM data, to train a network that is then capable of interpolation.

To obtain a useful data set, a two dimensional modeling program was used. The models chosen were conductive dikes in a less conductive medium, which could be considered any

conductive tabular feature. The depth to the top and dip angle of the dikes were varied. Success was measured by the ability of the trained network to interpolate the depth and dip angle of dikes that were not included in the training set. Further testing included non-dike models as well.

Chapter 2

Background

In order to familiarize the reader with the methods and systems used for this project, the following is a brief review of those methods. Only the most important features that are relevant to the goal of pattern recognition are included. The areas include the TEM method, Fourier transforms and neural networks.

2.1 TEM Basics

The use of the TEM method is chosen because each receiver provides more information per use of a single source than frequency electromagnetic or DC-resistivity methods. This stems from the fact the system measures response over a time range rather than at a single frequency or suite of discrete frequencies. Vanyan and Bobrovnikov's (1967), Kaufman and Keller (1983), Ward and Hohmann (1987), and Telford et al. (1990) derivations of the TEM response,

which are done in the frequency domain, demonstrate that the response is a summation of a series of frequency responses. Because the depth in the earth to which useful signal reaches is a function of frequency, ideally, each frequency component in TEM data yields different information about the subsurface. The fact that a layered medium sounding can be done with data from a single source receiver pair is used by Aktarakci (1993), Kaufman and Keller (1983), and Fitterman and Stewart (1986). It is this breadth of information that one attempts to tap.

To acquire this information a variety of TEM systems are used. All systems consist of a source and one or more receivers. The source is generally considered to be an electrical or magnetic dipole. Magnetic dipoles are square or round loops of wire with sizes from hundreds of meters to less than a meter. They can be oriented along any of the three axes. Electrical dipoles are a single line of wire run along the earth's surface. The current in TEM sources begins constant and is then rapidly terminated. This may be done cyclically so repeated measurements can be made.

The measurements are started from the beginning of the ramp period to tenths of seconds later, and may continue for

several seconds. The time range depends upon the type of target being searched for. Receivers generally consist of wire coils to measure the changing magnetic field and potential electrode meters for the electrical field. All three of the magnetic field components and the two electrical field components coplanar with the earth's surface may be measured. In some TEM systems the source and receiver are the same coil, coincident source-receiver systems, e.g. the SIROTEM.

The signal measured by the receiver is usually digitally sampled. The source signal is repeated and the responses are averaged to remove random noise. Each time point actually is a small time range over which the input signal is integrated. The result is stored in an individual time bin. The number of time bins varies from a few, the Geonics EM-61, to 20, the Geonics EM-47, to one or two thousand.

Two major source and receiver array combinations are used. First is the fixed source array. For this array the source remains stationary and only different receiver positions are used. The second is constant offset profiling. For this array the source and receiver pair

remain at a fixed distance from each other and both are moved along profiles. For a good background of the effects of various arrays see Hansen et al. (1967), and Keller and Frischknecht (1966). An overview is contained in Telford (1990).

While TEM data can have a great deal of information, it may not be in readily usable form. This is due to the physics of the method itself. The response of the method is due mainly to induction and then decay of current within confined conducting bodies, and diffusion through a less conductive earth. Initial induction between source and target create eddy currents within the conductor that decay according to its resistivity and geometry. These create secondary electric and magnetic fields one or more of which are measured. At the same time currents within the earth approximately form a "current filament" that diffuses and attenuates as it travels through the earth, like a "smoke ring" (Nabighian, 1979). As time progresses this ring expands and travels deeper, and as it passes roughly under the receiver the direction of the changing magnetic field reverses. Therefore, theoretically, by observing at the proper times, or using a given suite of transformed

frequencies, one can observe the effect of different confined conductors or layers. The problem is that all effects combine in ways that are not necessarily linear.

Further complicating the response, is the effect of a dipole source on induction. A typical TEM source is a magnetic or electric dipole. The field generated by the dipole source is like two elongated, asymmetric, spheroids end to end with the longer ends touching. A simple diagram of a dipole field is shown in figure 2.1 (Hansen et al., 1967).

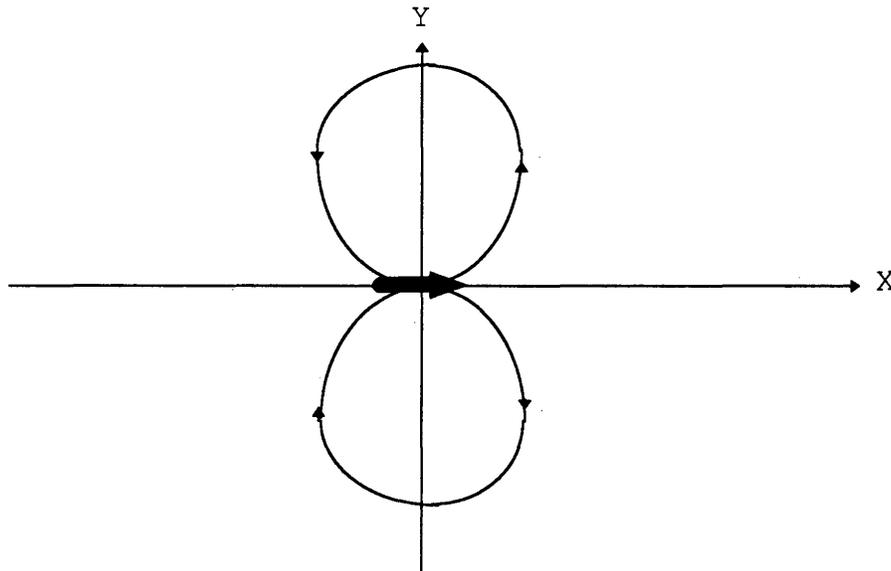


Figure 2.1 : Field due to dipole oriented positive x-direction

This field creates areas of greater response, and areas of lesser or possibly no response, so one must be careful of placement of the array and be aware of the target's possible location and geometry. For example, all dikes modeled for this paper were dipping away from the transmitter because inductive coupling with dikes dipping toward the transmitter can be weak, leading to poor resolution similar to the thin plates in Gallagher, Peggie and Ward (1985). Thus the layout of source and receivers affect the ability to resolve particular parameters (Keller and Frischknecht, 1966). For the dike models generated a single source with a line of receivers was used.

The final factor affecting the ability of TEM to resolve is the acquisition of the data at various times. The decay rates of the currents in confined conductors and layer media are exponential, thus requiring a great range of sample times. Further, many commercial systems commonly in use only acquire between 20 and 50 data points, exponentially increasing in time. For example, the PROTEM 47 has 20 time points that can range from 6 to 701 μs , or 48 to 2825 μs , or 100 to 7032 μs . Further, not only is the acquired data exponential in time, but also in amplitude,

which can have both positive and negative values. Important information may occur at an extremely wide range of times and amplitudes. The combination of exponential data in time and in amplitudes makes even display of such data a difficult task. A typical method of TEM raw data display is shown in figures 2.2 and 2.3.

Both figures represent data taken from a two dimensional model of a dike at the surface and with a 60 degree dip. This model, along with others will be described in detail in the next chapter. Figure 2.2 shows the transient response from a single receiver directly over the dike. The late times in the curve are essentially exponential decay. The characteristic change of sign in the changing magnetic field occurs around 10^{-5} seconds. Figure 2.3 shows a line of receivers, with each curve of overall lower amplitude being of later time. Each horizontal axis, except the last, contains 3 time gates that correspond to times which are discussed in the next chapter. The dike begins at the 40 m position and dips to the right. Data of this sort is not usable by the typical discrete Fourier transform program, which is the subject of the next section.

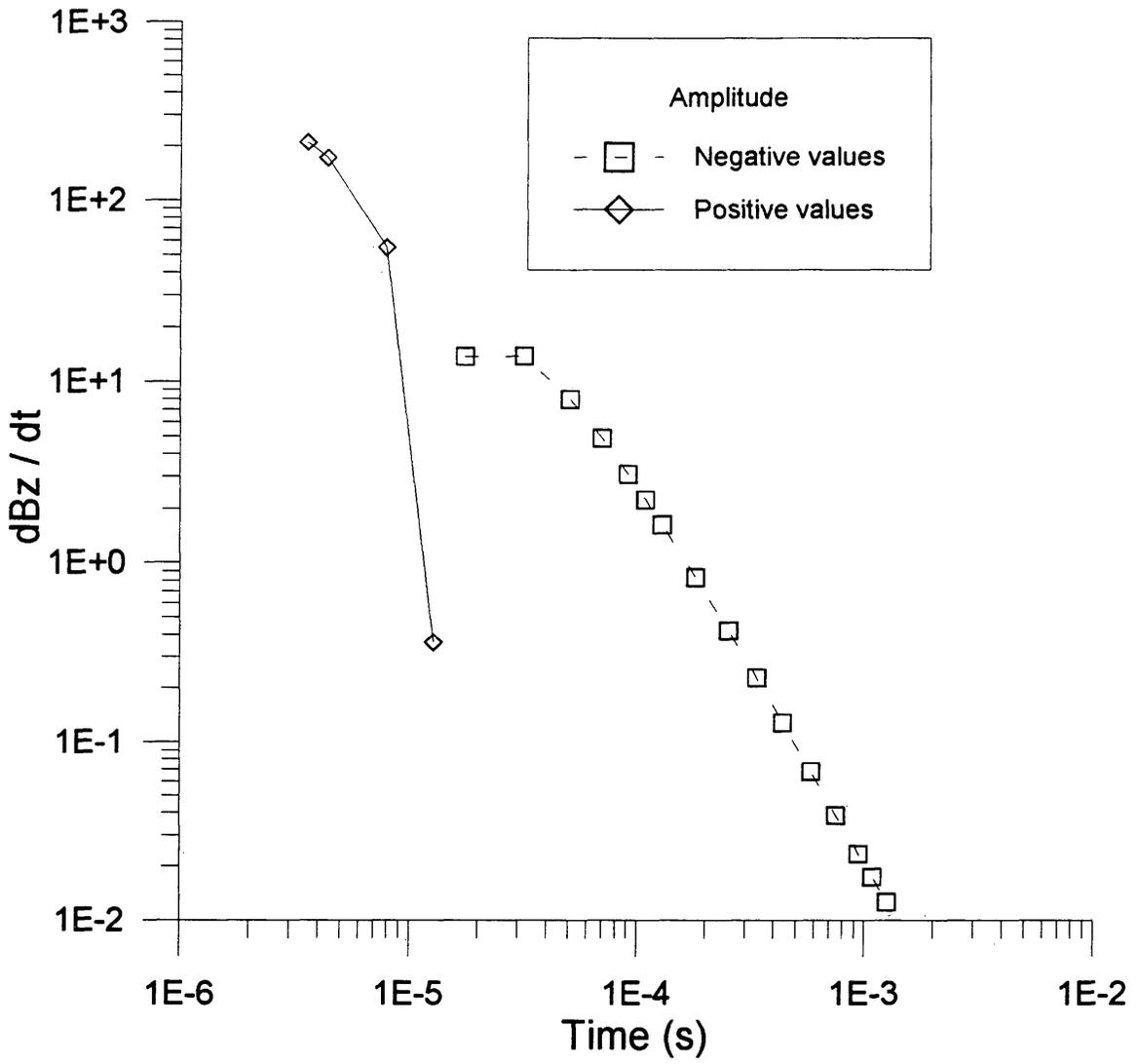


Figure 2.2 : TEM receiver data from dike model

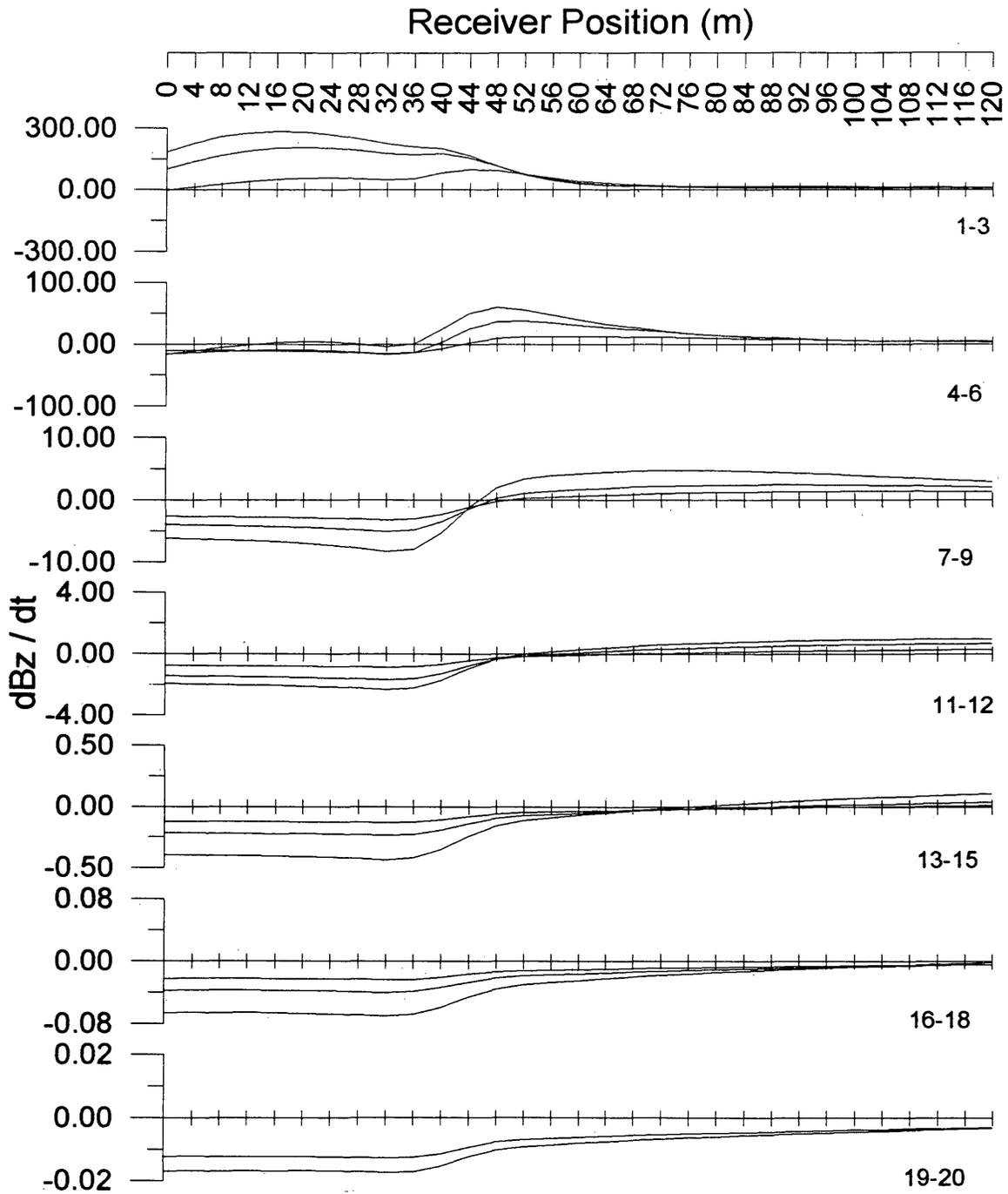


Figure 2.3 : Dike Model TEM data for 20 time points and 31 receivers

2.2 Fourier Transforms

Fourier transforms are known for their use in solutions to various differential equations and their ability to indicate the frequency structure of functions and discretely sampled data (Yilmaz, 1991). The transformation often leads to a collapse of the data into a summary representation or range of frequencies that is less than the original data or indicative of certain features of the data. It is the latter ability of the transform that is desired for pattern recognition. Simply, it is hoped that transforming the time and space data leads to a pattern that is unique and not readily visible in the original untransformed data.

Poulton et al. (1992) used Fourier transforms on their ellipticity data. From only the two frequencies and phase of the maximum amplitude of a two-dimensional data set, they were able to completely classify a set of patterns. Such a possibility was initially hoped for this project as well, but as will be shown, this amount of data reduction did not occur.

The result of the Fourier transform is a function that, for non-symmetric input functions, is complex. It is

generally most useful to examine the amplitude of the transform. The amplitudes corresponding to particular frequencies are primarily independent of the position of the structures in the original data, but rather represent the structures themselves. The phase of the transform, however, is affected by the original position of such structures. The ability of the transform to define structures regardless of initial position is useful in creating consistent patterns. For example, the actual spacing and locations of receiver stations may change, but the amplitude of the transform of the data received by them should be relatively consistent.

There is a multitude of ways to numerically perform the Fourier transform. One limitation of all discrete Fourier transforms (William et al. 1987), is the Nyquist frequency, which arises from the fact of having discretely sampled data, rather than continuous. The Nyquist frequency is proportional to one-half the reciprocal of the spacing between input data points. This places a limitation on the maximum frequency which provides useful data. Any frequency higher than Nyquist is subject to the possibility of an aliased signal. Frequencies below Nyquist may also suffer

from an erroneous increase in power due to the aliased higher frequencies.

The most common methods of Fourier transforms, both Discrete (DFT) and Fast Fourier Transforms (FFT) require regularly sampled data points. The inability to easily chose the desired frequencies, and occasional lack of sufficient data points are other problem with FFT's when trying to create consistent patterns. For TEM data, the spacing between sample points is not necessarily regular, and often increases exponentially. Besides ruling out the use of many discrete Fourier transforms, the notion of the Nyquist frequency becomes somewhat obscure.

2.3 Neural Networks

Neural networks have been referred to as the "second best solution to a problem". Their ability to classify a set of patterns or inputs, stems from the generality of their architecture. This generality, also makes the use of neural networks a matter of guesswork to some degree.

The basic building block of neural networks is the node. Each node accepts a number of inputs and multiplies each input by a different weight. All the products are then summed and become the argument of a transfer function which generates the output of the node; see figure 2.4.

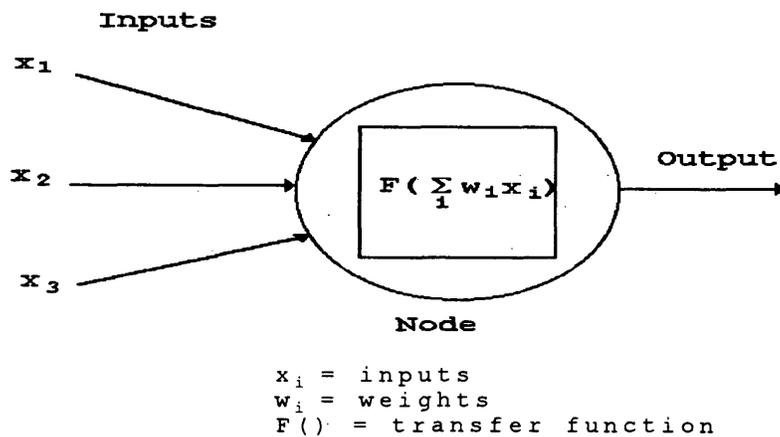


Figure 2.4: A sample node

A number of nodes are used in combination to create a network. A typical layout of a network, includes input nodes, which are just the input values, hidden nodes, and output nodes. This is shown in figure 2.5.

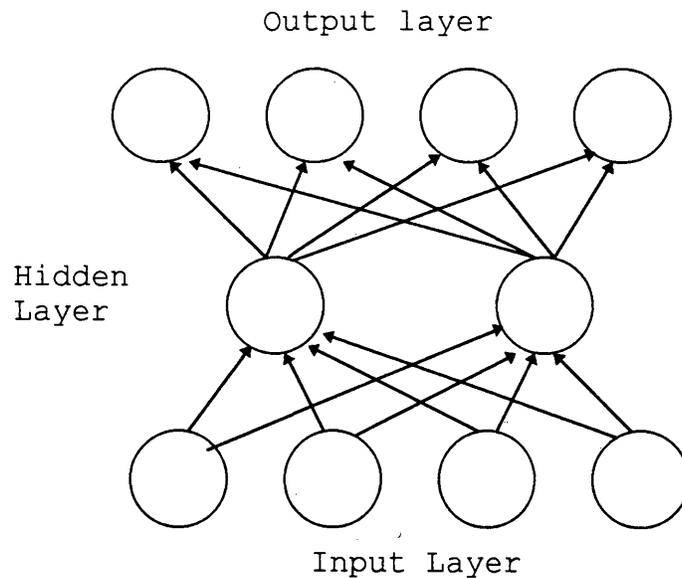


Figure 2.5: Typical Network Layout

Only the hidden, and output layers carry out the weighted sum. With the general topology of input, one or more hidden, and output layers, neural networks are capable of classifying a large number of given problems. This is achieved through adjusting the weights on the inputs, such that for given inputs, the desired output is attained. A

common method used to adjust the weights, and the one used for this project, is that of backpropagation. Other methods include genetic algorithms, and variants of backpropagation; see Leighton (1992) and Hertz, Krogh and Palmer (1991).

The basic principle of backpropagation is that the weights are adjusted in the direction of the gradient of the error. For the weight i on node j , w_{ji} , the change in weight is

$$\Delta w_{ji} = -\alpha \frac{\partial E}{\partial w_{ji}} \quad 2.1$$

where E is the total error, and α is a constant called the learning rate. The learning rate controls the amount the weights are changed with each pass. The total error is dependent upon whether the node is a hidden or output node and on the transfer function, for details see Elkington (1991), Leighton (1992), and Hertz et al. (1991). To smooth weight changes over time another term is added such that,

$$\Delta w_{ji}(t) = -\alpha \frac{\partial E(t)}{\partial w_{ji}} + \gamma \Delta w_{ji}(t-1) \quad 2.2$$

where γ is a constant called the inertia or momentum term. The momentum helps prevent wild oscillations in the weights.

The weights in the network are initially randomized. The network is then trained on a set of training data. After being shown each member of the training set, or the entire training set, the weights are updated according to method of backpropagation. When, for entire training set, the output of the net is within a specified error tolerance, the net is considered trained. There is no guarantee of the ability to train a network: backpropagation may become trapped in local minima on the error surface. The order in which the training data is fed into the network is also a factor when using backpropagation. To facilitate training a network, the learning rate, the momentum, the node layout, and transfer function may need to be altered. However, there are no standard ways of choosing the above features. Using networks is currently a task based on guesswork, and "rules-of-thumb".

The transfer function of the network can theoretically be any function. However, to use backpropagation the transfer function must be differentiable, in order to find the gradient of the error function. Functions that are bounded are usually chosen, because unbounded functions may not allow the network time to converge and become trained.

The most commonly used function, and the transfer function in the neural networks used for this project, is the sigmoidal function,

$$f(\text{net}) = \frac{1}{1 + e^{-\text{net}}} \quad 2.3$$

shown in figure 2.6.

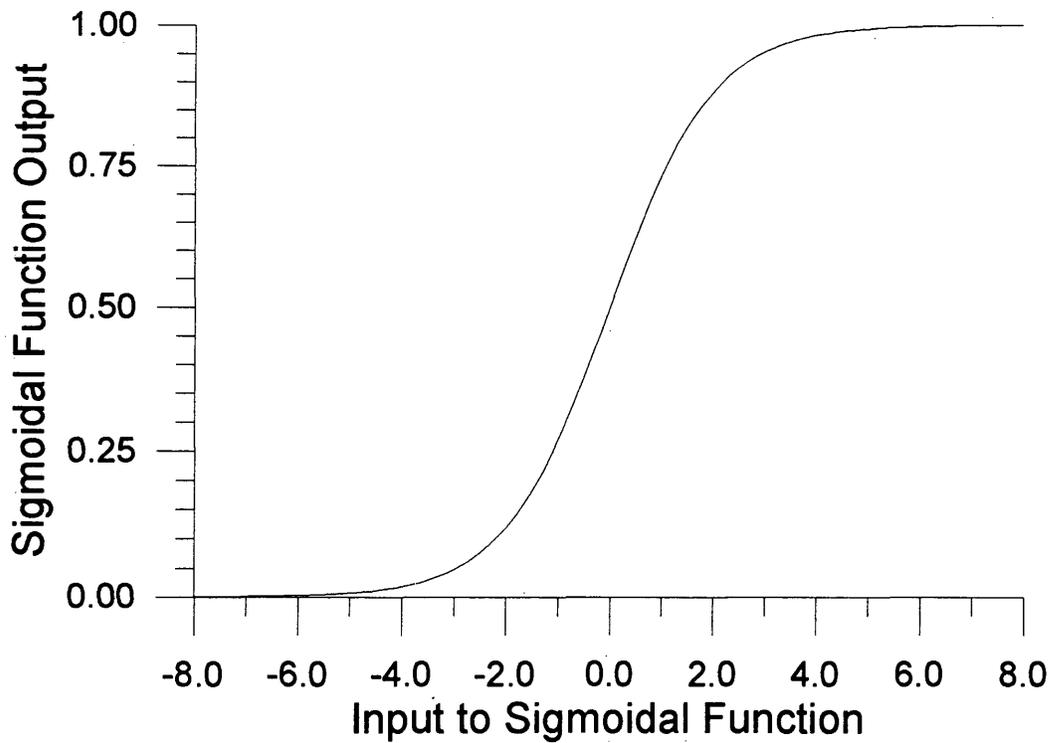


Figure 2.6: Sigmoidal Transfer Function

The sigmoidal function is a bounded function which is easily differentiable. Sometimes, offset is added to the output so it varies from -0.5 to 0.5 or scaled to -1 to 1 . The function is adept at reaching values near 0 or 1 in order to specify yes or no, a goal of many networks. The goal of this project, however, is to be able to interpolate between values in the training data. Presumably then for this purpose, the best area for output is within the region where the sigmoidal function is essentially linear. Inspection of figure 2.6 rapidly shows this area to be approximately located between 0.1 and 0.9 in the sigmoidal output.

One of the most important features of the network, is its topology. With too many or too few nodes, especially hidden nodes, the network will not converge or it may classify the training set, but not be able to generalize. The classic example of this is the classification of a two bit exclusive or, where at least one hidden node is required. The ability to change topology leads to an infinite range of possible networks, and there is not a mechanized method for choosing topology. One important topology, used in pattern recognition and in the networks

for this project, is tessellation. Tessellation occurs when the each node of one layer connects only to some of the nodes in the previous layer. This is contrasted to a fully connected network as shown in figure 2.5. Tessellated networks can also contain overlap, such that the nodes in a prior layer may still be connected to one or more nodes in the next layer. An excerpted example of tessellation is shown in figure 2.7.

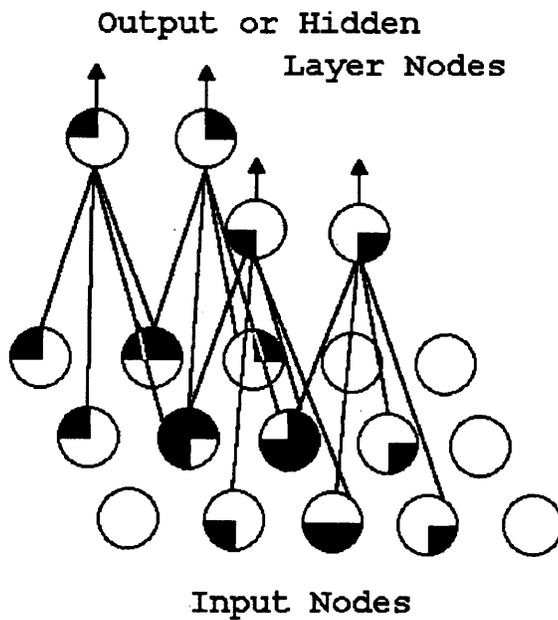


Figure 2.7: Example of An Excerpted Portion from a Tessellated Network

In the figure, input nodes are connected to the output nodes which share the same blackened quarters. Tessellated networks cut down on the total number of nodes, which can decrease learning time, or even the ability to learn at all. Through proper use of tessellation, areas of the pattern to be recognized can be emphasized; this might result in an increased ability of the network to learn.

The final factors affecting the learning ability of a network are the learning rate and momentum. These are to some degree chosen in tandem. A high momentum can decrease the effect of a high learning rate. In some cases a high learning rate is necessary to escape local minima in the error surface. Both values vary from 0 to 1. Momentum is generally around .9 and the learning rate .2 or less. With some insight and luck the features of a neural network can be chosen so that learning actually occurs.

Learning means successful classification of the training data. The network then classifies new data depending upon how it was trained. This classification may be to place new data in one of the classes of the training data, or as desired for this project, produce output that

has interpolated between the values in the training data and properly represents the new data point.

Chapter 3

Methods

This chapter describes the methods and details of the models created, Fourier transformations, and the neural network. The parameters used for the dike models are first described. Then the actual methodology of the Fourier transformation program is explained. Finally, the topology of the neural network and its parameters are given.

3.1 Dike Models

The aim was to create a geometric feature which could be described by a few parameters, but such that the parameters could not be casually determined from inspection of the data. The model chosen was that of a conductive dike, or in two dimensions a tabular feature. The depth and dip of the dike were the parameters to be determined through pattern recognition by the neural network.

To actually create the modeled data an explicit two-dimensional finite difference TEM modeling program based on the Dufort-Frankel scheme was used (Hohman, 1987), (Oristaglio and Hohmann, 1984). To create a model, a spatial finite difference grid of horizontal and vertical distances, and the time values for which calculation occurs are first defined. The program solves the secondary differential equations for a two-dimensional earth excited by a line source of current, by stepping through time and space, according to the user-defined grid. The finite differences are used to approximate differences between nodes and solutions are found through matrix inversion.

The spatial grid consisted of a central area containing 4 by 4 meter squares for 59 horizontal cells and 31 vertical cells. The intersection of the cells define each node on the grid. All dikes modeled were placed entirely within this area. Outside of the central area the cells increase in dimension horizontally away from the center and vertically below it. A representative schematic of the grid and cells is shown in figure 3.1. The actual parameters for the cells used are shown in table 3.1.

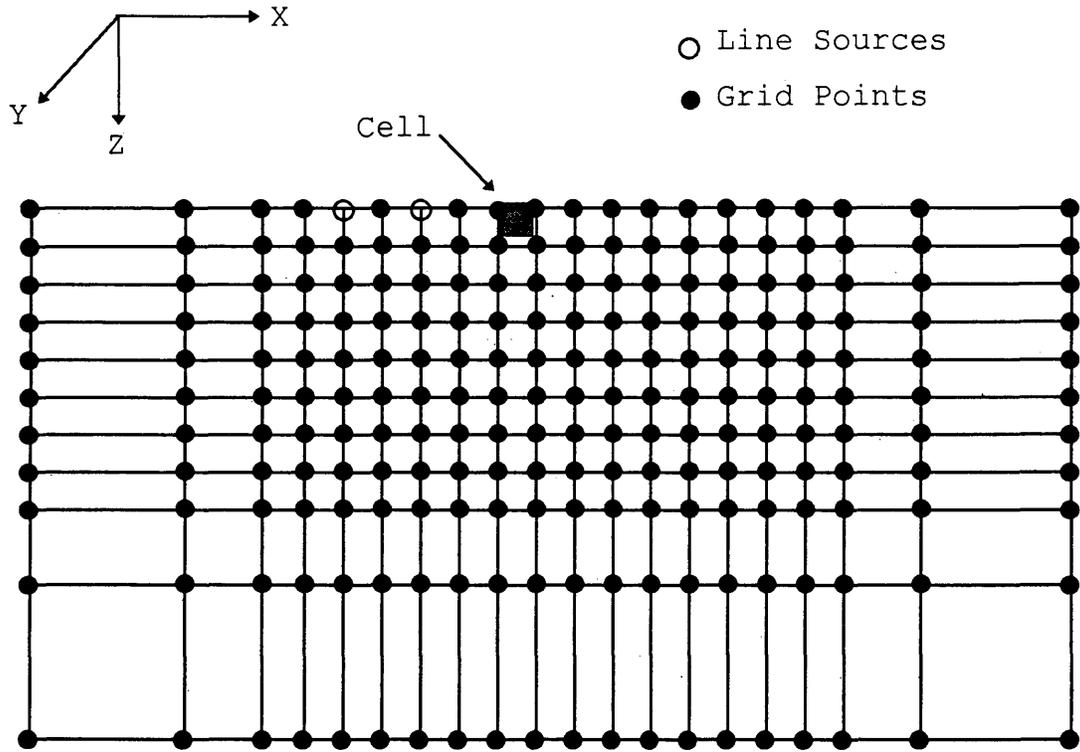


Figure 3.1 : Example finite difference grid

<u>Horizontally</u>		<u>Vertically</u>	
Number of Cells	Width of Cells (m)	Number of Cells	Height of Cells (m)
2	120	31	4
2	92	9	10
3	32	4	15
5	16	2	20
4	8	2	40
59	4	2	80
4	8	2	160
5	16	2	200
3	32		
2	92		
2	120		

Table 3.1: Parameters for finite difference grid

Twenty time values were chosen for output. Those values, the time increments and number of iterations at that increment for internal calculation are shown in table 3.2.

Time Increment	Number of Iterations	Output after # Iterations	Time at output
1 E-8	180	180	3.6 E-6
1 E-8	40	40	4.4 E-6
3 E-8	60	60	8.0 E-6
3 E-8	80	80	1.3 E-5
3 E-8	80	80	1.8 E-5
6 E-8	120	120	3.2 E-5
6 E-8	160	160	5.1 E-5
6 E-8	160	160	7.0 E-5
6 E-8	180	180	9.2 E-5
6 E-8	150	150	1.1 E-4
6 E-8	170	170	1.3 E-4
6 E-7	44	44	1.8 E-4
6 E-7	60	60	2.5 E-4
6 E-7	70	70	3.4 E-4
6 E-7	85	85	4.4 E-4
6 E-7	120	120	5.9 E-4
6 E-7	140	140	7.5 E-4
6 E-7	160	160	9.5 E-4
3 E-6	24	24	1.0 E-3
3 E-6	28	28	1.3 E-3

Table 3.2: Modeled time points

All models used the above described finite difference grid and time points. Also common to all models was the location of the source and receiver grid. A single source consisting of two line sources of opposite polarity, carrying 1 ampere of current, spaced 52 meters apart was used. The receiver array began 20 meters away from the nearest line source and contained 31 receivers spaced 4 meters apart or at each grid point. The source and receivers remained in a fixed location for each model. The receivers measured both vertical and horizontal components of changing magnetic field. The horizontal component is perpendicular to the strike of the grid, or in the Y-direction in figure 3.1. However, only the vertical component was used.

The targets for detection consisted of "dikes" created in the equally sized 4 by 4 meter element area previously described. The width of the dikes was approximately 8 meters, but this varied with the dip angle. The variation occurred because the angles can truly only be approximated with the given square element grid. All dikes were long enough so that the bottom edge of the dike reached the bottom or right edge of the equally sized element area. All

dikes began 40 meters into the receiver array or 60 meters from the nearest source line. Only the depth to the top of the dike and the dip angle, away from the source, were varied. This is shown in figure 3.2.

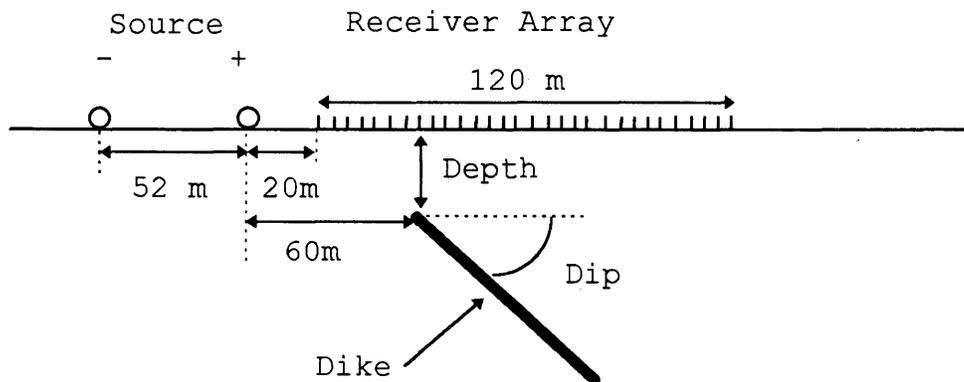


Figure 3.2: Diagram of dike model

Two sets of dike models were created: the training set, and the test set. The neural net was trained on the training set, and tested on the test set. Each set consisted of a series of dikes at different angles and different depths. The training set included the angles 15, 30, 45, 60, 75, and 90 degrees for depths of 0, 8, and 20 meters, and angle of 30, 60, 90 at a depth of 32 meters, for a total of 21

training models. The test set consisted of the angles 20, 40, and 55 degrees for depths of 0, 8, and 16 meters, for a total of 9 test models. All dikes had a resistivity of 10 ohm-meters and were the only feature in an otherwise homogenous earth with resistivity of 100 ohm-meters.

Two other models were created to test the response of the neural network. A slab with a rectangular cross section, and an object with a diamond shaped cross section were made. They were chosen because of the difference between the slab and a dike, and the similarity of the edges of a diamond to the other dike models. Also, given the constraints of the square finite-difference grid, they were shapes that could be approximated fairly well. The slab, see figure 3.3, extended 20 meters vertically and 24 meters horizontally, beginning at a depth of 8 meters and 20 meters into the receiver array. The diamond, see figure 3.4, began 32 meters into the receiver array at a depth of 4 meters, and had a maximum horizontal extent of 64 meters and a vertical extent of 28 meters. Both models had resistivities of 10 ohm-meters in the 100 ohm-meters background. After creating each model the next step was to perform Fourier transforms upon them.

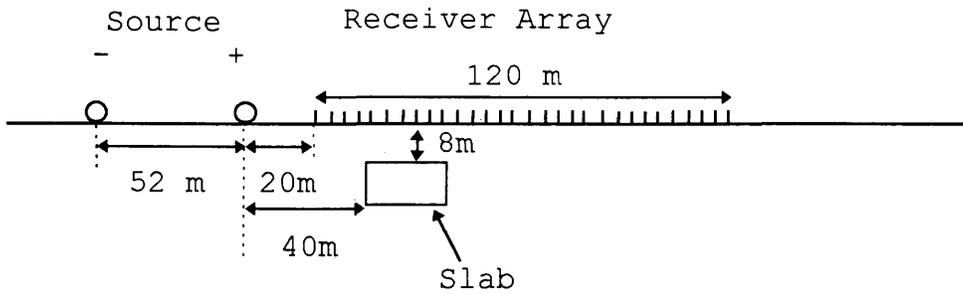


Figure 3.3: Schematic of Slab Model

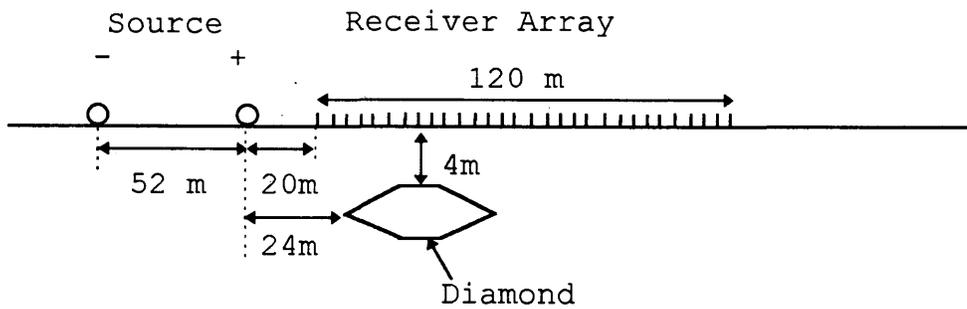


Figure 3.4: Schematic of Diamond Model

3.2 Fourier Transformation Routine

The data from the modeling program consisted of 20 irregularly, but approximately logarithmic spaced time points covering several orders of magnitude in time, and amplitude. This type of data is not readily transformed by the typical discrete Fourier Transform program, so a routine was written for this task. The program linearly interpolates between time points, and applies the analytical solution of the transform to the interpolation. The following describes this method.

Fourier transforms are done by use of the Fourier integral. The program used for this project used the following form of that integral:

$$\tilde{g}(f) = \int_{-\infty}^{\infty} G(t) e^{-2\pi i f t} dt \quad (3.1)$$

where $\tilde{g}(f)$ is the Fourier transform of $G(t)$. With equation 3.1 the transformed variable, t , becomes a frequency variable, f . For each square integrable function, there is a single Fourier transform, and the inverse of this transform is the original function. However, the inverse transform is not used in this paper.

The actual integral for sampled data is accomplished by parameterizing a linear equation between two data points for which the analytical solution for the Fourier transform is known. To accomplish this first equation 3.1 is rewritten using Euler's Identity.

$$\tilde{g}(f) = \int_{-\infty}^{\infty} [G(t) \cos(2\pi f t) - iG(t) \sin(2\pi f t)] dt \quad (3.2)$$

For each pair of data points $G(t)$ is approximated by a linear equation $Y(t)$, where

$$\begin{aligned} Y(t) &= mt + b \\ m &= \frac{G(t_2) - G(t_1)}{t_2 - t_1} \\ b &= \frac{G(t_1)t_2 - G(t_2)t_1}{t_2 - t_1} \end{aligned} \quad (3.3)$$

So for each data point pair the Fourier integral becomes

$$\tilde{g}(f) = \int_{t_1}^{t_2} \{mt[\cos(2\pi f t) - i \sin(2\pi f t)] + b[\cos(2\pi f t) - i \sin(2\pi f t)]\} dt \quad (3.4)$$

Now the solutions to each term in the integral between each two time points are analytic.

$$\begin{aligned}
\omega &= 2\pi f \\
\int_{t_1}^{t_2} mt \cos(\omega t) dt &= m \left(\frac{\cos(\omega t)}{\omega^2} + \frac{t \sin(\omega t)}{\omega} \right) \Big|_{t_1}^{t_2} \\
\int_{t_1}^{t_2} -imt \sin(\omega t) dt &= -im \left(\frac{\sin(\omega t)}{\omega^2} - \frac{t \cos(\omega t)}{\omega} \right) \Big|_{t_1}^{t_2} \quad (3.5) \\
\int_{t_1}^{t_2} b \cos(\omega t) dt &= b \frac{\sin(\omega t)}{\omega} \Big|_{t_1}^{t_2} \\
\int_{t_1}^{t_2} -ib \sin(\omega t) dt &= -ib \frac{\cos(\omega t)}{\omega} \Big|_{t_1}^{t_2}
\end{aligned}$$

Equations 3.5 are applied to each pair of time points in the data. The real and imaginary parts are then summed for the whole data set. This completes the Fourier transformation for data that was originally completely real numbers, as is the changing magnetic field from the TEM data. However after doing the time-frequency transformation, the data set is now complex. The above method is still applied, to the real and imaginary parts separately.

For a complex function, $G(x) = Y(x) + iZ(x)$, where $G(x)$ is complex, and $Y(x)$, $Z(x)$, and x are real, the Fourier transform is

$$\begin{aligned}
\tilde{g}(k) &= \int_{-\infty}^{\infty} [Y(x) + iZ(x)][\cos(\omega x) - i \sin(\omega x)] dx \\
\tilde{g}(k) &= \int_{-\infty}^{\infty} [Y(x) \cos(\omega x) - iY(x) \sin(\omega x) + iZ(x) \cos(\omega x) + Z(x) \sin(\omega x)] dx \quad (3.6)
\end{aligned}$$

Thus, one can apply the previously described routine to $Y(x)$ and $Z(x)$, then multiply the $Z(x)$ result by $-i$ and sum the two transform results.

Using the above methodology has the following advantages. First, the input data does not have to be equally spaced. In fact, the best placing of data samples is such that the function between the two points is assumed to be fairly linear. This is approximately the case with the TEM data sample points. Second, the user is allowed to choose any frequencies desired. Of course, care should be taken to observe the Nyquist frequency as described in chapter 2. The ability to chose frequencies is important in order to create consistent patterns. For example, a survey may have stations spaced between 2 to 4 meters. However, since any frequencies can be chosen, the amplitude of the Fourier transform output theoretically is the same for surveys over the same area with different spacings. The same holds relatively true for the TEM sample times. Consistent patterns are important for neural network recognition.

The Fourier transform result is a complex one. To achieve the consistent patterns as described above only the

amplitudes are used as input into the neural network. However, if the horizontal position of the dike, rather than just depth and angle, was desired, the phase of the transform might possibly hold necessary information. A sample of the Fourier time transformed modeled TEM data is shown in figure 3.5. The logarithm of the amplitude of the Fourier transform is contoured. The model is the same 60 degree dike from earlier figures. The frequency range chosen was found by trying to adhere to Nyquist frequency, and then simply by observing the patterns generated by different ranges. The total number of temporal frequencies used was 96. Figure 3.6 is the same 60 degree dike model used so far transformed in both time and space. From the 31 receiver positions, 32 spatial frequencies are made. These patterns, after normalization, were the data fed into the neural network. The pattern shown, at first glance, appears to be a difficult one to characterize, but with a proper network it is still possible. The layout of the neural network is described in the next section.

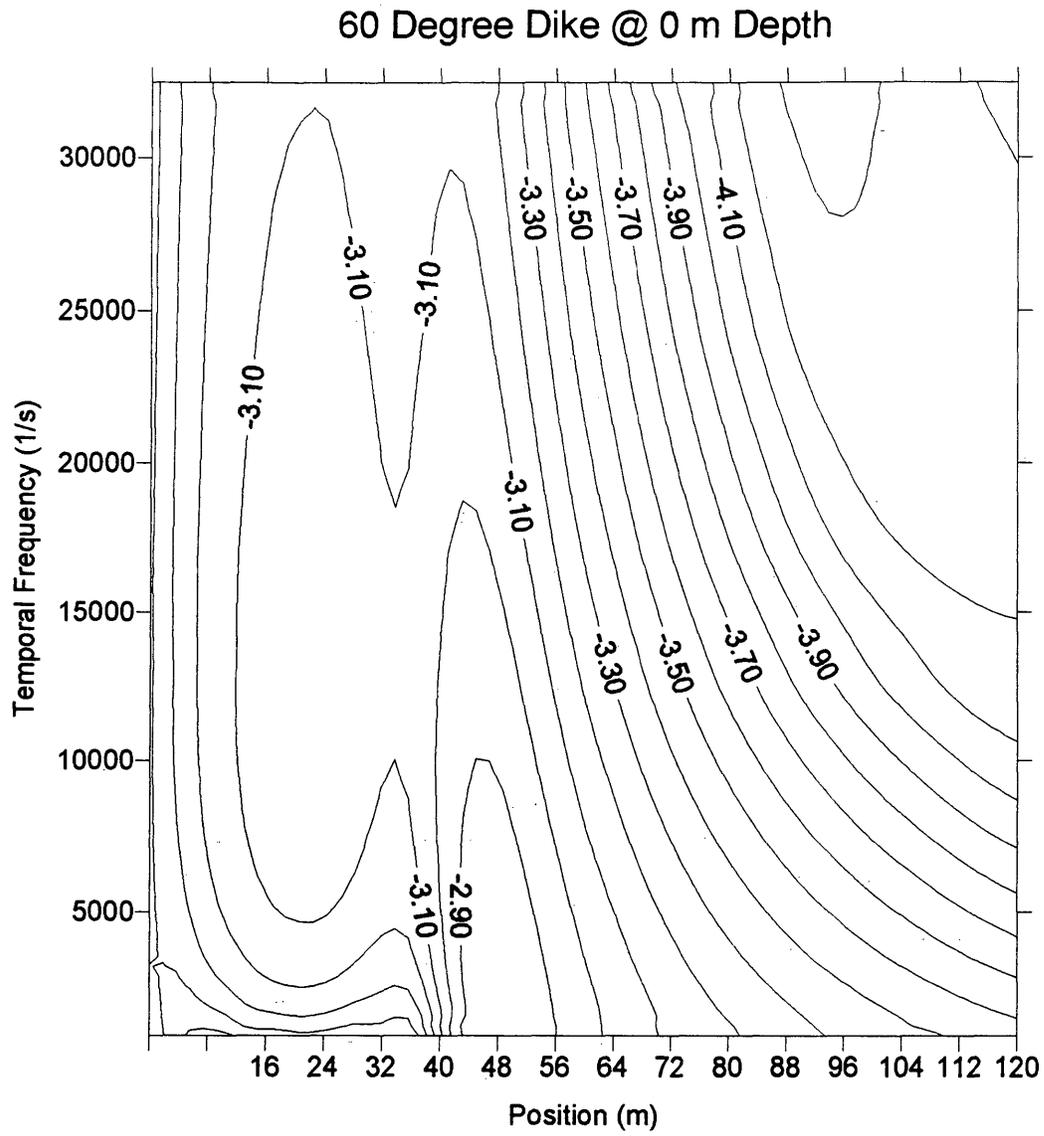


Figure 3.5 : Temporal Transform for 60 degree dike model

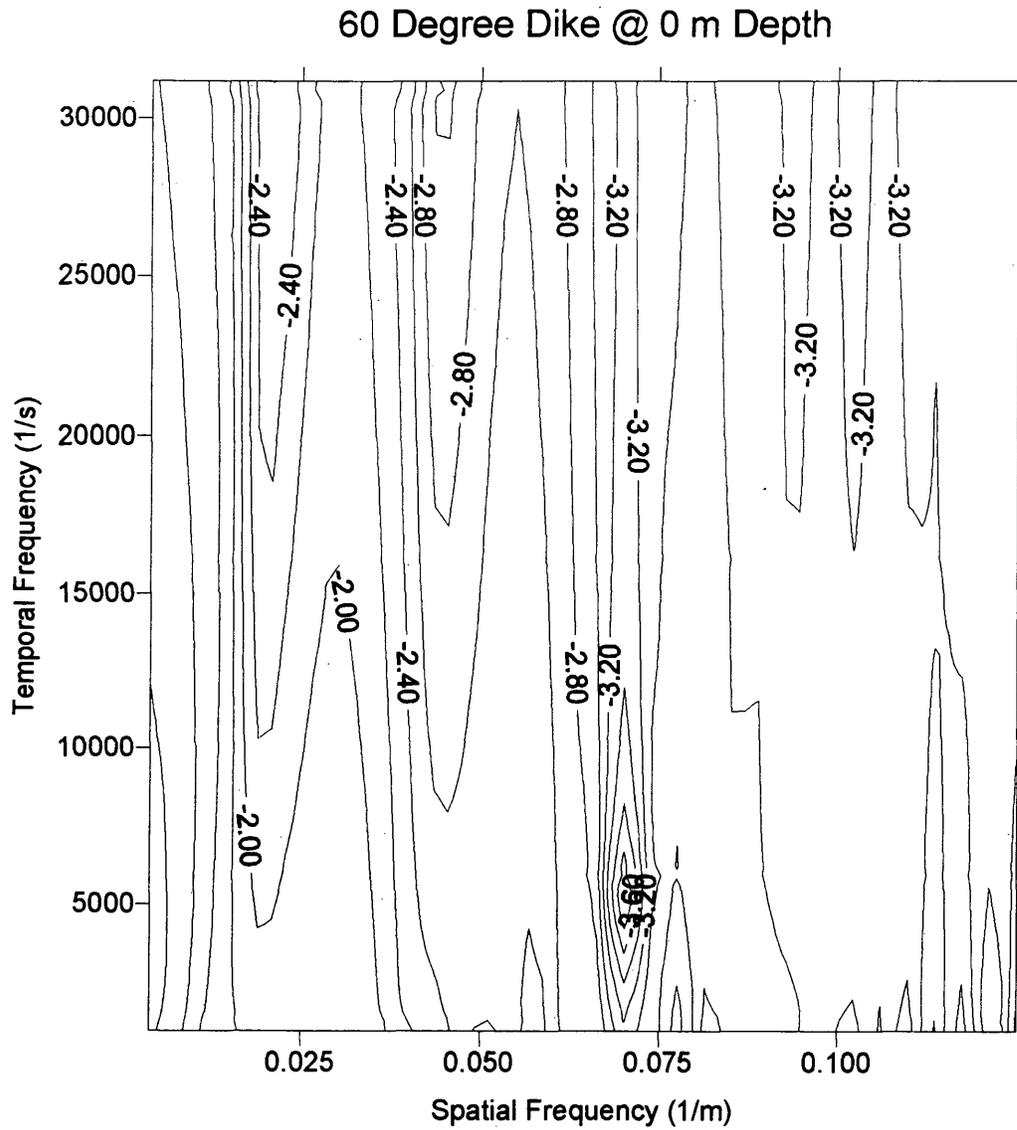


Figure 3.6 : Spatial and Temporal Transform for 60 degree dike model.

3.3 Neural Networks

As stated in the previous section, the design of neural networks can contain a substantial amount of guesswork. This proved to be the case with the dike models. The patterns derived in the spatial and temporal transforms appear to be primarily striking along the temporal axis. This fact led to the layout and the use of a tessellated network. The networks were created through use of the Aspirin/MIGRAINES Neural Network Software by Leighton (1992), which is available over the internet.

Any network used would consist of a layer 32 x 96 input nodes. It was quickly found using various hidden layers, that a fully connected network failed to converge. So a tessellated network was designed.

The final network used consisted of a 32 x 96 input layer, a hidden layer of 10 x 4 where each node covered a 5 x 30 area of the input nodes. The areas then had a 2 x 8 overlap with each other. This arrangement fully covers the input nodes. The tessellated network design was chosen because of the predominantly linearly up and down striking features in the Fourier transform patterns.

Finally to be chosen is the number of output nodes. Several attempts used a number of output nodes to binary encode the depths and angles of the dikes. The final method used consisted of two output nodes, one for angle and one for depth. All nodes in the network were of the sigmoidal type previously described.

In order for the output to be encoded, a range of the sigmoid output was used and divided linearly into the angle and depth ranges. The angle range went from .125 to .875 which corresponded to 0 to 90 degrees. The depth range went from .1 to .9 which corresponded to 0 to 32 meters.

The parameters which allowed for training consisted of a learning rate of .02, a momentum of .9 and a convergence test of .02. The convergence test means that all training patterns fell within a ± 0.02 range of the desired output. The training data was fed into the network in ascending angle order for lowest depth, and then for the next depth, and so on.

In order to be fed into the network, however, the input data needed to be normalized. A routine was written so that the data was normalized from 0 to 1 such that the original lowest value became 0, the highest value became 1,

and the rest linearly fit in between. An example of such input for the same 60 degree angle dike previously used is shown in figure 3.7.

An attempt was also made to test the effect of removing possible noise in the bottom right corner of the spectral patterns. The data values over the clipped range were averaged and replaced with that average value. An example of a clipped pattern for the same dike is shown in figure 3.8.

After the networks were successfully trained they were tested on the test data set, to determine their ability to interpolate. Since the data in the learning set was already correctly classified through learning, feeding the network any of the original training data set would result in successful classification to within the error tolerance.

To test the network's ability to tolerate noise, two noisy data sets were made. A noise of $\pm 5\%$ and $\pm 10\%$ was added to the output from the TEM modeling program. The data was transformed and input into the network. Further, the network was trained on noisy data sets and tested on both clean and noisy data sets. The results of these tests are discussed in the next chapter.

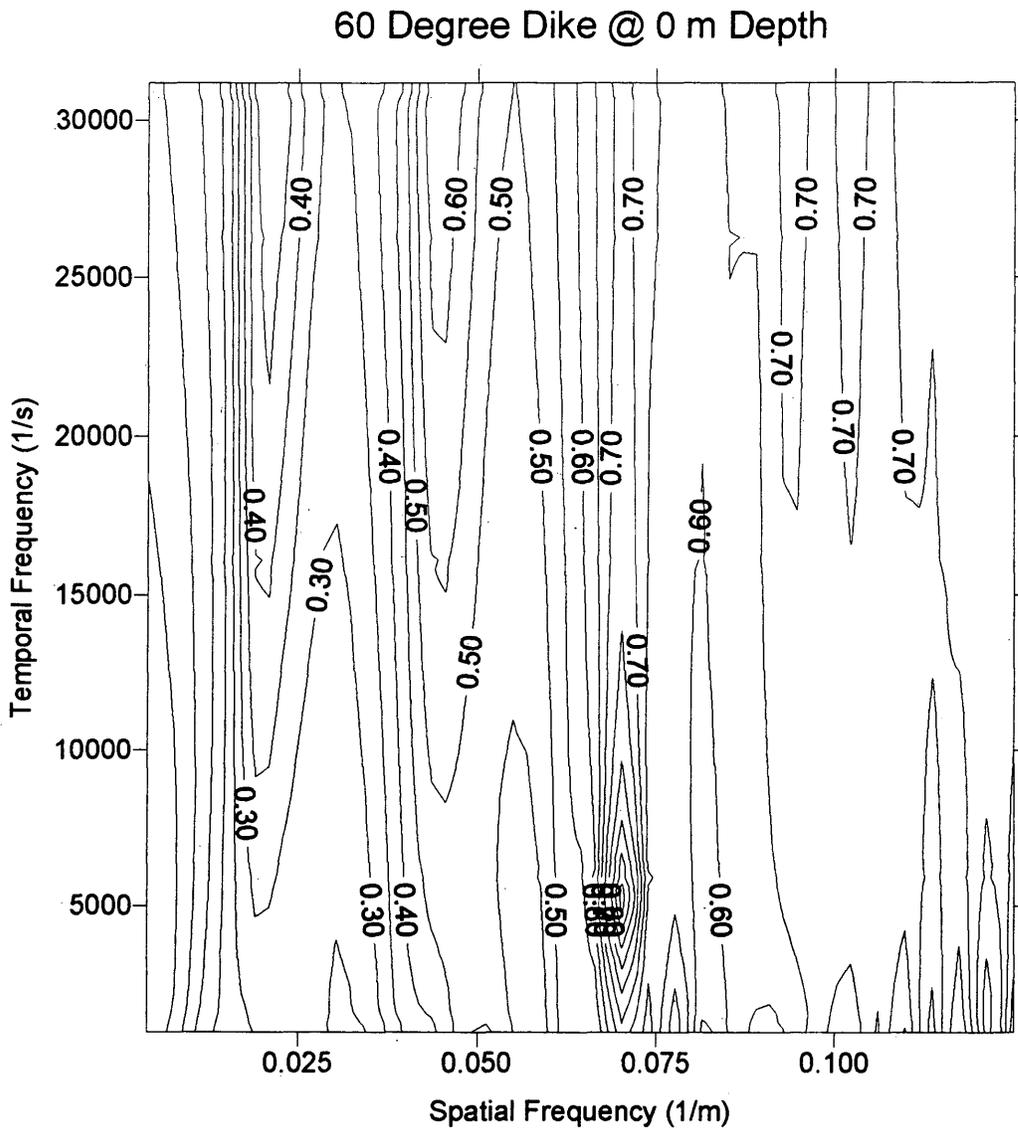


Figure 3.7 : Spatial and Temporal Transform Normalized for Neural Net Input for 60 degree dike model

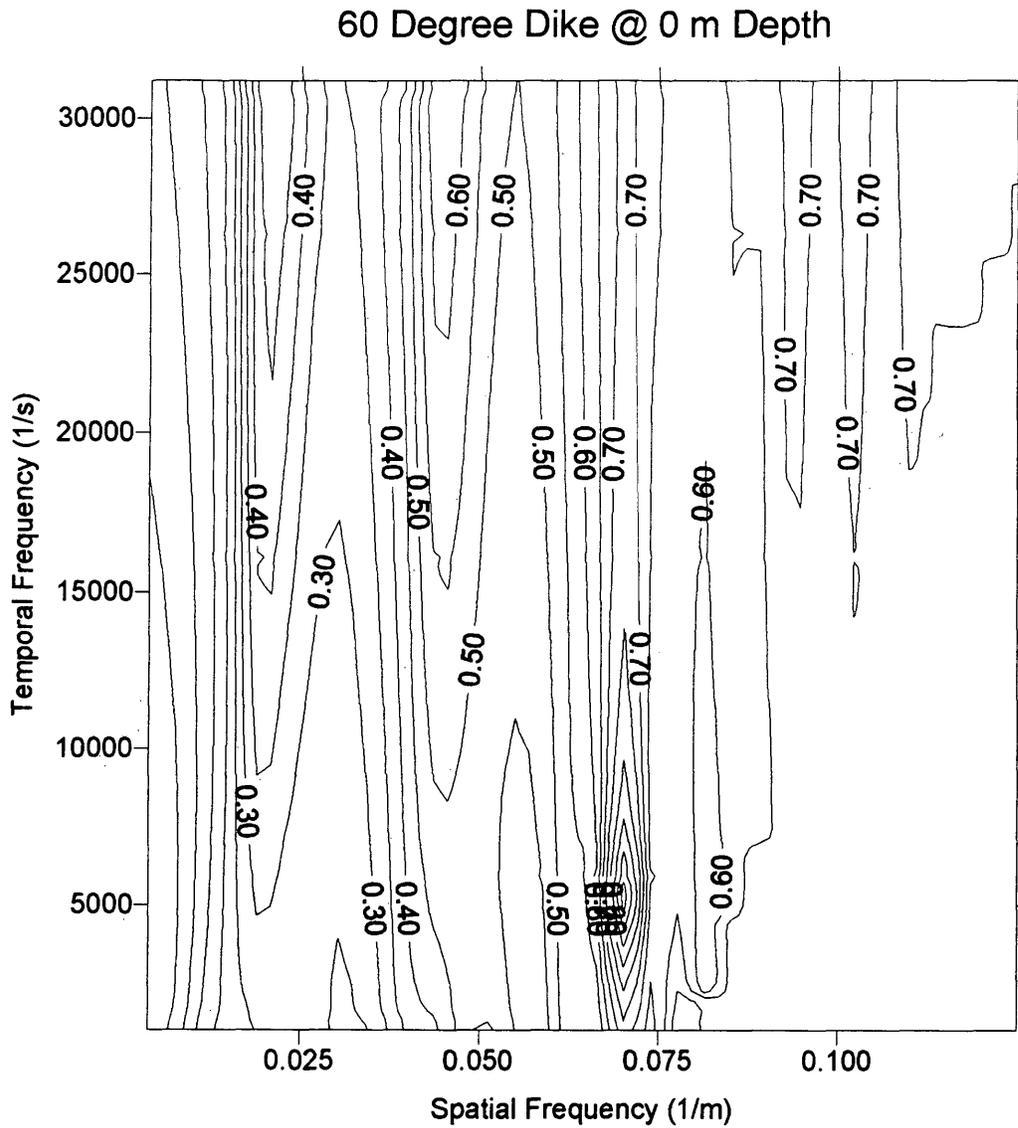


Figure 3.8 : Spatial and Temporal Transform
Normalized and Clipped
for Neural Net Input
for 60 degree dike model

Chapter 4

Results

After all the dike models were created, Fourier transformed, in both time and space, and normalized, they were used to train and test the capability of the neural network. The network was trained and tested on a normal data set, a clipped data set, and a series of noisy data set combinations. The results of these tests follow, along with a derivative analysis of some of the data.

4.1 Noise-free Results

The network was first trained on the unclipped noise-free training data set consisting of the 21 models described in section 3.1. The training for this and all training sets described hereafter was successful. This means all training patterns were classified to within the .02 raw output error, which correlates to 2.4 degrees and 0.8 meters. The network was then tested with the test data

set, also described in section 3.1. The results of this test are shown in table 4.1.

Overall the network showed a reasonable ability to interpolate the dike attribute values between those in the training data. The most spurious point, 20 degrees at 8 m depth, will be further analyzed in the next section. To try and improve the results, a clipped data set was created as described in section 3.3. Both the training and test sets were clipped. The results of this test are shown in table 4.2.

The clipping definitely improved performance. Except for the previously mentioned spurious data point, all but one of the values are within the neural net training error range. An analysis of the failure of the network at the one data point, and the benefits of clipping are done through Frechet derivatives in the next section.

Data Clipping	Training Noise	Test Noise	Data File
No	0%	0%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.2583	16.00	-11.44%	-20.02%	-4.00
2	0.4583	40.00	0.3979	32.75	-13.17%	-18.11%	-7.24
3	0.5833	55.00	0.6021	57.25	3.22%	4.09%	2.25
4	0.2917	20.00	0.1981	8.77	-32.10%	-56.17%	-11.24
5	0.4583	40.00	0.4475	38.70	-2.37%	-3.25%	-1.30
6	0.5833	55.00	0.5983	56.80	2.57%	3.28%	1.80
7	0.2917	20.00	0.2926	20.11	0.31%	0.54%	0.11
8	0.4583	40.00	0.4785	42.42	4.41%	6.07%	2.43
9	0.5833	55.00	0.5983	56.80	2.57%	3.28%	1.80
Square Average :					12.40%	21.00%	4.88

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.08571	-0.57	-14.29%	N/A	-0.57
2	0.1	0	0.07864	-0.85	-21.36%	N/A	-0.85
3	0.1	0	0.09074	-0.37	-9.27%	N/A	-0.37
4	0.3	8	0.34687	9.87	15.62%	23.44%	1.87
5	0.3	8	0.27983	7.19	-6.72%	-10.09%	-0.81
6	0.3	8	0.3057	8.23	1.90%	2.85%	0.23
7	0.5	16	0.50195	16.08	0.39%	0.49%	0.08
8	0.5	16	0.47074	14.83	-5.85%	-7.32%	-1.17
9	0.5	16	0.49024	15.61	-1.95%	-2.44%	-0.39
Square Average:					10.94%	10.95%	0.88

Table 4.1 : Results of noise-free non-clipped test data

Data Clipping	Training Noise	Test Noise	Data File
Yes	0%	0%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.2894	19.72	-0.80%	-1.40%	-0.28
2	0.4583	40.00	0.4668	41.02	1.86%	2.56%	1.03
3	0.5833	55.00	0.5832	54.98	-0.02%	-0.02%	-0.01
4	0.2917	20.00	0.2241	11.89	-23.19%	-40.58%	-8.12
5	0.4583	40.00	0.4630	40.56	1.02%	1.40%	0.56
6	0.5833	55.00	0.5908	55.89	1.28%	1.63%	0.90
7	0.2917	20.00	0.2991	20.89	2.54%	4.44%	0.89
8	0.4583	40.00	0.4591	40.09	0.17%	0.23%	0.09
9	0.5833	55.00	0.5870	55.44	0.63%	0.81%	0.44
Square Average :					7.83%	13.66%	2.77

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.0882	-0.47	-11.81%	N/A	-0.47
2	0.1	0	0.0882	-0.47	-11.81%	N/A	-0.47
3	0.1	0	0.1074	0.30	7.44%	N/A	0.30
4	0.3	8	0.3684	10.74	22.80%	34.20%	2.74
5	0.3	8	0.2926	7.70	-2.47%	-3.70%	-0.30
6	0.3	8	0.3057	8.23	1.90%	2.85%	0.23
7	0.5	16	0.4824	15.30	-3.51%	-4.39%	-0.70
8	0.5	16	0.4707	14.83	-5.85%	-7.32%	-1.17
9	0.5	16	0.5059	16.23	1.17%	1.46%	0.23
Square Average:					10.07%	14.53%	1.06

Table 4.2 : Results of noise-free clipped test data set

4.2 Frechet Derivative Analysis

The spurious data point at 20 degrees and 8 m depth, and the cause of the benefit from clipping, led to the use of Frechet derivative analysis of some of the data. The derivatives consist of the subtraction of one Fourier transformed image from another, and nominally dividing by the value of the difference between the changed attribute, all other attributes remaining the same. The purpose of the Frechet derivatives is to test the sensitivity of the network to changes in the Fourier transform patterns. The analysis was done on angle separations of 5 degrees. The Frechet derivative for the angles 20 and 15 degrees at 8 m depth is shown in figure 4.1. The derivatives for angles 20 and 15 at 0 m depth and angle 45 and 40 at 8 m depth are shown in figures 4.2 and 4.3 respectively.

The derivatives show that essentially the data at higher than .08 spatial frequency do not contribute to pattern recognition. This area was probably dominated by an aliased signal noise. Thus, the benefit of clipping, results from the removal of irrelevant data. However, what might have been even more beneficial would have been to

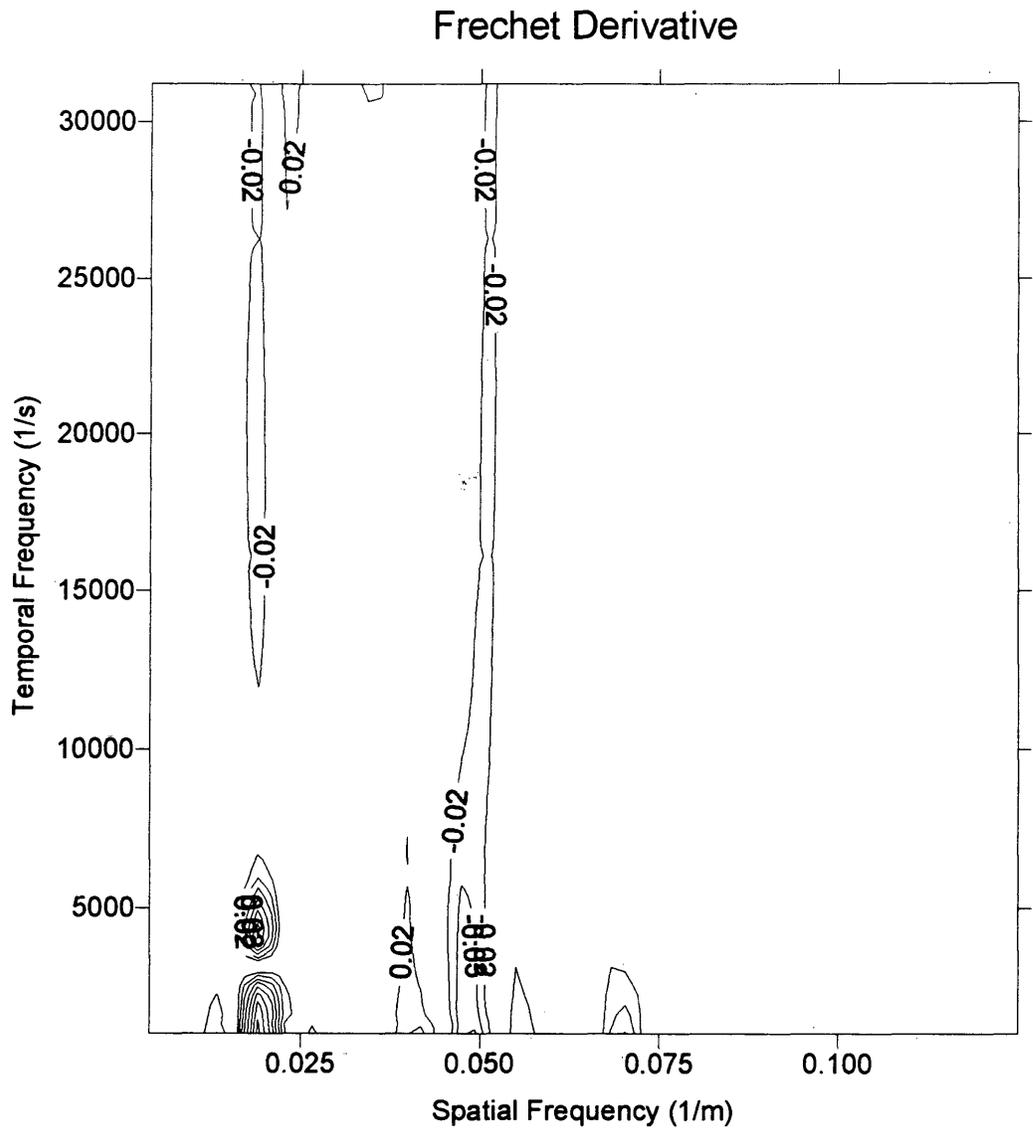


Figure 4.1 : Frechet Derivative for angles 20 and 15 @ 8 m depth

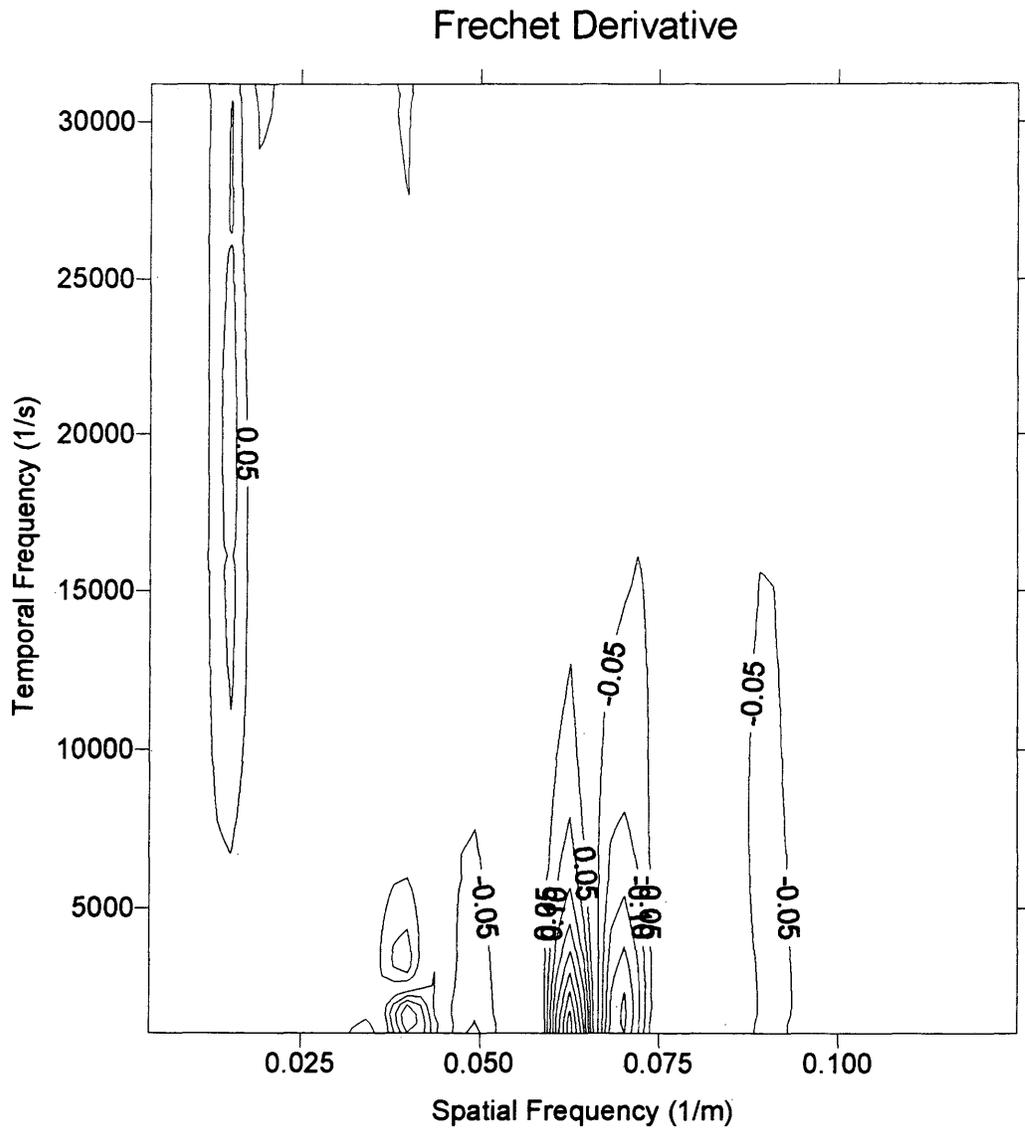


Figure 4.2 : Frechet Derivative for angles 20 and 15 @ 0 m depth

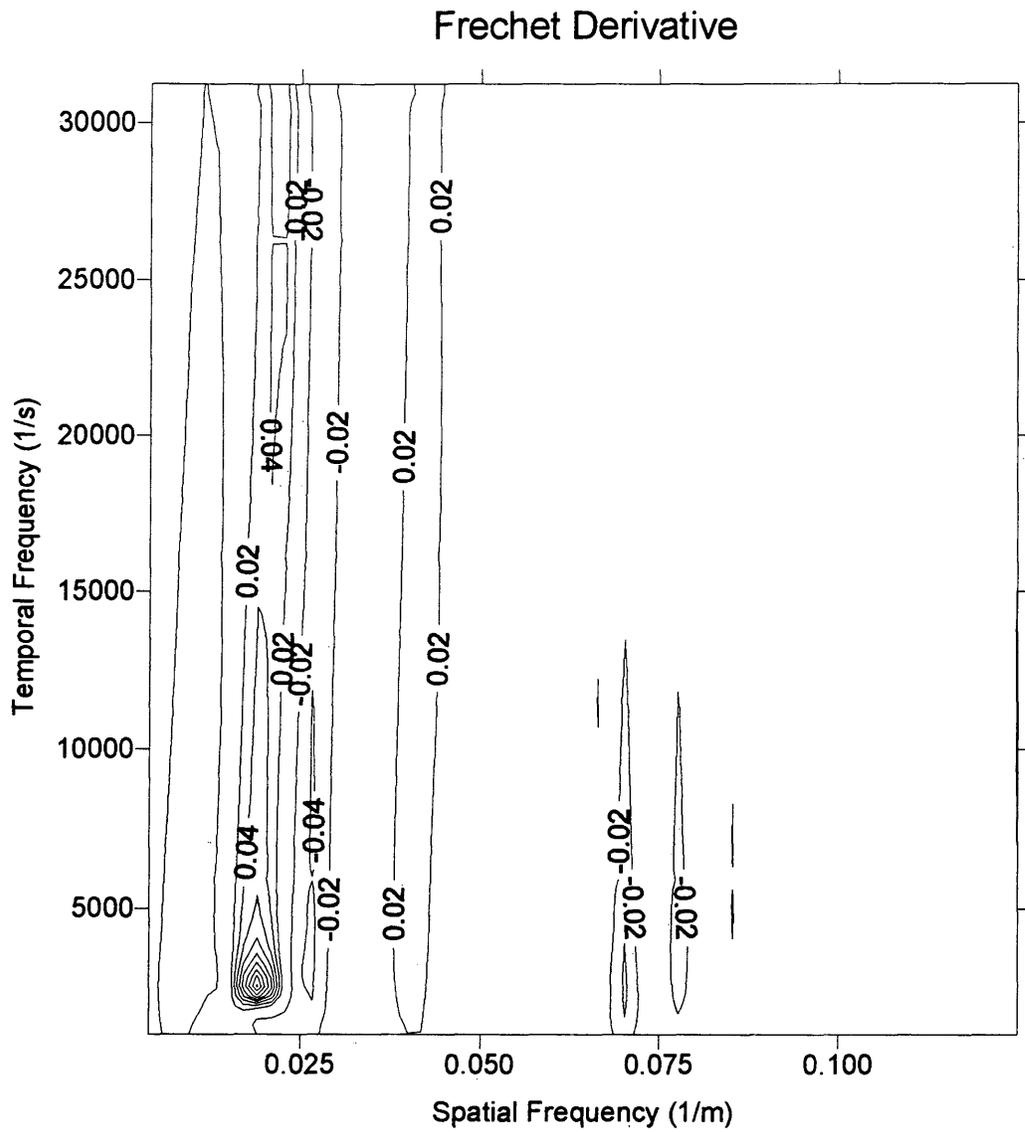


Figure 4.3 : Frechet Derivative for angles 45 and 40 @ 8 m depth

remove those points all together, and to increase the number of lower spatial frequencies. The derivatives at the lower depths show less intense change than those at higher depths. For the angles of 20 and 15 at 8 m depth, however, the magnitude of the change is lowest of all. This probably accounts for the failure in pattern recognition.

The reason for the small changes in patterns for the 20 degree angle at 8 m depth is probably due to poor coupling between source and receiver. This problem arises again when dealing with different models, as will be shown later. The single source multiple receiver layout may not be the best for resolving individualistic patterns.

4.3 Noisy Data Results

The network was also tested and trained on noisy data. This could simulate either having a model trained network, and attempting to resolve real data with it, or perhaps training a network on real data containing noise. In some cases the ability of neural networks has improved when trained on noisy data. That was not the case here however.

The test consist of several combinations of noisy training and test sets. Two amounts of white random noise were used, $\pm 5\%$ and $\pm 10\%$. The noise was added to the output of the modeling program, and then Fourier transformed and used with the neural network. The network was then trained on each of the three noise levels, noise-free, $\pm 5\%$, and $\pm 10\%$. For each one of the noise levels the network was trained on is was tested on the other two noise levels. Thus, the training sets of 21 models can be used as test sets for networks trained on training sets with a differing amount of noise.

The results for noise-free trained networks on $\pm 5\%$ test data for the non-clipped and clipped data sets are shown in

tables 4.3 and 4.4 respectively. The rest of the tests are contained in appendix A.

A random noise value of $\pm 5\%$ is fairly typical for real TEM surveys. The neural network, then, may actually be capable of resolving real data sets. However, for real data sets an important question is whether or not the object is dike-like at all. This subject is addressed in the next section.

Data Clipping	Training Noise	Test Noise	Data File
No	0%	5%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.2736	17.83	-6.21%	-10.87%	-2.18
2	0.4583	40.00	0.3942	32.30	-13.99%	-19.23%	-7.69
3	0.5833	55.00	0.6132	58.59	5.13%	6.53%	3.59
4	0.2917	20.00	0.2134	10.61	-26.85%	-46.98%	-9.40
5	0.4583	40.00	0.4552	39.62	-0.68%	-0.93%	-0.37
6	0.5833	55.00	0.5679	53.15	-2.63%	-3.35%	-1.84
7	0.2917	20.00	0.2767	18.20	-5.15%	-9.00%	-1.80
8	0.4583	40.00	0.4902	43.83	6.97%	9.58%	3.83
9	0.5833	55.00	0.6021	57.25	3.22%	4.09%	2.25
Square Average :					10.93%	18.07%	4.61

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.10302	0.12	3.02%	N/A	0.12
2	0.1	0	0.08946	-0.42	-10.55%	N/A	-0.42
3	0.1	0	0.10017	0.01	0.17%	N/A	0.01
4	0.3	8	0.33285	9.31	10.95%	16.43%	1.31
5	0.3	8	0.26741	6.70	-10.86%	-16.30%	-1.30
6	0.3	8	0.28299	7.32	-5.67%	-8.51%	-0.68
7	0.5	16	0.56794	18.72	13.59%	16.98%	2.72
8	0.5	16	0.42441	12.98	-15.12%	-18.90%	-3.02
9	0.5	16	0.51757	16.70	3.51%	4.39%	0.70
Square Average:					9.52%	14.56%	1.53

Table 4.3 : Results of ±5% noise non-clipped data

Data Clipping	Training Noise	Test Noise	Data File
Yes	0%	5%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.2524	15.29	-13.48%	-23.58%	-4.72
2	0.4583	40.00	0.4359	37.31	-4.89%	-6.72%	-2.69
3	0.5833	55.00	0.6095	58.14	4.50%	5.72%	3.15
4	0.2917	20.00	0.2241	11.89	-23.19%	-40.58%	-8.12
5	0.4583	40.00	0.4941	44.30	7.82%	10.75%	4.30
6	0.5833	55.00	0.6206	59.47	6.40%	8.14%	4.48
7	0.2917	20.00	0.3057	21.68	4.80%	8.40%	1.68
8	0.4583	40.00	0.4475	38.70	-2.37%	-3.25%	-1.30
9	0.5833	55.00	0.6496	62.95	11.36%	14.46%	7.95
Square Average :					10.66%	17.49%	4.85

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.11358	0.54	13.58%	N/A	0.54
2	0.1	0	0.10302	0.12	3.02%	N/A	0.12
3	0.1	0	0.09335	-0.27	-6.65%	N/A	-0.27
4	0.3	8	0.34334	9.73	14.45%	21.67%	1.73
5	0.3	8	0.27983	7.19	-6.72%	-10.09%	-0.81
6	0.3	8	0.31237	8.49	4.12%	6.19%	0.49
7	0.5	16	0.42823	13.13	-14.35%	-17.94%	-2.87
8	0.5	16	0.48243	15.30	-3.51%	-4.39%	-0.70
9	0.5	16	0.44746	13.90	-10.51%	-13.14%	-2.10
Square Average:					9.65%	13.68%	1.39

Table 4.4 : Results of $\pm 5\%$ noise clipped data

4.4 Results of Other Models

Two models, described in section 3.1: the diamond, and the slab, were also used as tests on the dike model trained neural network. The results, computed parameters for a dike-like structure, of the slab were a .998 for the angle, which is well out of the allowed range, and .372 or 10.8 m for the depth. The slab began at a depth of 8 meters. Essentially the .998 angle value would be rejected and one would assume the network did not recognize a dike. The results were not as good for the diamond.

The diamond received a .525 or 48 degrees for the angle and .576 or 19 m for the depth. The diamond began at a depth of four meters and continued down for 28 meters. Its far top side was approximately at a 45 degree angle. The diamond may have been recognized because the currents mainly concentrated in areas that resembled those of the dikes. This is a potential problem with the single source array. As was mentioned earlier this array may not be the best suited for characteristic pattern creation.

Chapter 5

Conclusions

The main objective of this thesis was achieved. Through the use of Fourier transforms, in both time and space, on TEM data, identifiable patterns were created and were classified by a neural network that was also capable of interpolation. There are still margins for improvement, however, and final analysis of the data suggests some methods of doing so. Further, there are a multitude of scenarios to which this paradigm may be applied.

While the patterns created appeared adequate for pattern recognition, final analysis suggests improvement in this area. More time points, more importantly at earlier times, could improve the initial time transformation. The range of frequencies was chosen to avoid what appeared as possible noise. The Frechet derivative analysis, though, points to a definite decrease in the maximum spatial frequency and probably the temporal maximum and minimum frequencies. An overall decrease in the number of temporal frequencies might have also been beneficial.

The network topology appeared to function rather well. With better patterns and fewer inputs, though, a fully connected network might have been able to perform as well. While only sigmoidal transfer functions were used, linear functions, possibly just on the output nodes, could increase distance between outputs and improve performance.

The fact that the network classified the diamond may have been due to the TEM array layout. Certainly different networks would have to be trained for both single fixed sources, and fixed offset source receiver pairs used in profiling. The latter array may lead to even more characteristic patterns.

Use of newer modeling programs, which use three-dimensional sources on a two-dimensional earth, while they may not improve performance, are probably necessary for the network to function well on real data. The ability of the network to interpret real data is, of course, the final goal. The network's performance on the noisy data suggests that it should be capable of such work.

Finally, the methods used to create the Fourier transforms and neural network are simple and general enough to be applied to a wide range of TEM problems. The final

trained network is fast and small enough to be run on almost any portable computer in use today. Such a system could be used in the field to rapidly identify possible targets. It may also be of use to suggest initial geometries for iterative modeling programs.

References Cited

- Aktarakci, Hasan Kamil. Principles and Application of Transient Electromagnetic Method. Thesis T-4331, Colorado School of Mines: Golden, CO, 1993.
- Cisar, Darrin, John Dickerson, and Timothy Novotny. "Electromagnetic Data Evaluation Using a Neural Network: Initial Investigation-Underground Storage Tanks," Chem-Nuclear Geotech, Inc, and Mesa State College: Grand Junction, CO.
- Elkington, Elizabeth A. The Role of Trace Attributes in the Implementation of Neural Networks for Seismic First Arrival Evaluation. Thesis T-4084, Colorado School of Mines: Golden, CO, 1991.
- Fitterman, David V. and Mark T. Stewart. "Transient Electromagnetic Sounding for Groundwater," Geophysics, Vol. 51., No. 4 (April 1986), pp. 995-1005.
- Gallagher, Peggie R., Stanley H. Ward, and G. W. Hohmann. "A Model Study of a Thin Plate in Free Space for the EM37 Transient Electromagnetic System," Geophysics, Vol. 50, No 6 (June 1985), pp. 1002-1019.
- Hansen, Don A. (ed.), et al. Society of Exploration Geophysicists' Mining Geophysics Volume II, Theory. Oklahoma: The Society of Exploration Geophysicists, 1967.
- Hertz, John, Anders Krogh, and Richard G. Palmer. Introduction to the Theory of Neural Computation. Redwood City, CA: Addison-Wesley, 1991.
- Hohmann, Gerald W. "Numerical Modeling for Electromagnetic Methods of Geophysics," from Misac N. Nabighian (ed.), Electromagnetic Methods in Applied Geophysics Vol. 1, Theory. United States: Society of Exploration Geophysicists, 1987, pp. 313-361.
- Kaufman, Alexander A. and George V. Keller. Frequency and Transient Soundings. New York: Elsevier, 1983.

- Kaufman, Alexander A. Geophysical Field Theory and Method Electromagnetic Fields I. San Diego: Academic Press, 1992.
- Keller, George V. and Frank C. Frischknecht. Electrical Methods in Geophysical Prospecting. New York: Pergamon Press Inc., 1966.
- Leighton, Russel R. The Aspirin/MIGRAINES Neural Network Software User's Manual Release V6.0. Russel Leighton and the MITRE Corporation, 1992.
- McCormack, Michael D., David E. Zaucha, and Dennis W. Dushek. "First-Break Refraction Event Picking and Seismic Data Trace Editing Using Neural Networks," Geophysics, Vol. 58, No. 1 (January 1993), pp. 67-78.
- Nabighian, M. N. "Quasi-Static Transient Response of a Conducting Half-space-an Approximate Representation," Geophysics, Vol. 44 (1979), pp. 1700-1705.
- Oristaglio, M. L. and G. W. Hohmann. "Diffusion of Electromagnetic Fields into a Two-Dimensional Earth: A Finite-Difference Approach," Geophysics, Vol. 49, No. 7 (July 1984), pp. 870-894.
- Poulton, Mary M., Ben K. Sternberg, and Charles E. Glass. "Location of Subsurface Targets in Geophysical Data Using Neural Networks," Geophysics, Vol. 57, No. 12 (December 1992), pp. 1534-1544.
- Press, William H., Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. Numerical Recipes The Art of Scientific Computing. New York: Cambridge University Press, 1987.
- Telford, W. M., L. P. Geldart, and R. E. Sheriff. Applied Geophysics Second Edition. New York: Cambridge University Press, 1990.

- Vanyan, L. L. and L. Z. Bobrovnikov. "Electrical Prospecting with the Transient Magnetic Field Method," from Vanyan, L. L. Electromagnetic Depth Soundings. Translated by George V. Keller. New York: Consultants Bureau, 1967.
- Ward, Stanley H., and Gerald W. Hohmann. "Electromagnetic Theory for Geophysical Applications," from Misac N. Nabighian (ed.), Electromagnetic Methods in Applied Geophysics Vol. 1, Theory. United States: Society of Exploration Geophysicists, 1987, pp. 131-308.
- Williams, Peter K. An Investigation into the Utility of Neural Networks for Recognizing Different Reflected Electromagnetic Spectra. Thesis T-4030, Colorado School of Mines: Golden, CO, 1991.
- Yilmaz, Özdoğan. Seismic Data Processing. Oklahoma: Society of Exploration Geophysicists, 1991.

Appendix A

Noisy Data Results

The neural network was trained on three training sets, with no noise, $\pm 5\%$ and $\pm 10\%$ noise. Each trained network was then tested on the other training and test data sets. Additional results to those not listed in the results chapter are given in this appendix. At the top of each data table the noise in the training data, the noise in the test data, and if clipping was used are listed. Also given is whether the test data was the 21 model training set or the 9 model training set.

Data Clipping	Training Noise	Test Noise	Data File
Yes	0%	5%	Training

Training Model #	esired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.250	15	0.2705	17.46	8.19%	16.38%	2.46
2	0.375	30	0.3648	28.77	-2.73%	-4.09%	-1.23
3	0.500	45	0.5059	45.70	1.17%	1.56%	0.70
4	0.625	60	0.5176	47.11	-17.19%	-21.49%	-12.89
5	0.750	75	0.7170	71.04	-4.40%	-5.28%	-3.96
6	0.875	90	0.9980	104.76	14.06%	16.40%	14.76
7	0.250	15	0.2583	16.00	3.33%	6.66%	1.00
8	0.375	30	0.3720	29.65	-0.79%	-1.18%	-0.35
9	0.500	45	0.5487	50.84	9.73%	12.98%	5.84
10	0.625	60	0.5870	55.44	-6.08%	-7.60%	-4.56
11	0.750	75	0.7447	74.36	-0.71%	-0.86%	-0.64
12	0.875	90	0.8470	86.63	-3.20%	-3.74%	-3.37
13	0.250	15	0.2705	17.46	8.19%	16.38%	2.46
14	0.375	30	0.3469	26.62	-7.50%	-11.25%	-3.38
15	0.500	45	0.5679	53.15	13.59%	18.12%	8.15
16	0.625	60	0.6943	68.32	11.09%	13.86%	8.32
17	0.750	75	0.7535	75.42	0.46%	0.55%	0.42
18	0.875	90	0.9236	95.83	5.55%	6.48%	5.83
19	0.375	30	0.4475	38.70	19.32%	28.98%	8.70
20	0.625	60	0.9093	94.11	45.48%	56.85%	34.11
21	0.875	90	0.9902	103.83	13.17%	15.36%	13.83
Square Average:					13.49%	17.73%	9.98

Training Model #	esired output		Actual output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.0933	-0.27	-6.65%	N/A	-0.27
2	0.1	0	0.0947	-0.21	-5.32%	N/A	-0.21
3	0.1	0	0.0960	-0.16	-3.98%	N/A	-0.16
4	0.1	0	0.1432	1.73	43.19%	N/A	1.73
5	0.1	0	0.1285	1.14	28.53%	N/A	1.14
6	0.1	0	0.0215	-3.14	-78.55%	N/A	-3.14
7	0.3	8	0.3057	8.23	1.90%	2.85%	0.23
8	0.3	8	0.2991	7.96	-0.30%	-0.45%	-0.04
9	0.3	8	0.2894	7.57	-3.54%	-5.31%	-0.43
10	0.3	8	0.3090	8.36	3.01%	4.51%	0.36
11	0.3	8	0.2958	7.83	-1.39%	-2.08%	-0.17
12	0.3	8	0.3504	10.02	16.81%	25.21%	2.02
13	0.6	20	0.5176	16.70	-13.74%	-16.49%	-3.30
14	0.6	20	0.6058	20.23	0.97%	1.16%	0.23
15	0.6	20	0.5641	18.56	-5.98%	-7.18%	-1.44
16	0.6	20	0.6206	20.82	3.44%	4.12%	0.82
17	0.6	20	0.6058	20.23	0.97%	1.16%	0.23
18	0.6	20	0.5487	17.95	-8.55%	-10.27%	-2.05
19	0.9	32	0.9131	32.52	1.45%	1.63%	0.52
20	0.9	32	0.8449	29.80	-6.12%	-6.88%	-2.20
21	0.9	32	0.8984	31.94	-0.18%	-0.20%	-0.06
Square Average:					21.38%	8.93%	1.41

Table A.1 : Results of $\pm 5\%$ noise clipped training data from noise-free trained network

Data Clipping	Training Noise	Test Noise	Data File
Yes	5%	5%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.3124	22.48	7.09%	12.40%	2.48
2	0.4583	40.00	0.5370	49.44	17.18%	23.62%	9.45
3	0.5833	55.00	0.7447	74.36	27.66%	35.21%	19.36
4	0.2917	20.00	0.2214	11.56	-24.12%	-42.20%	-8.44
5	0.4583	40.00	0.3757	30.08	-18.02%	-24.78%	-9.91
6	0.5833	55.00	0.6388	61.66	9.52%	12.12%	6.66
7	0.2917	20.00	0.3057	21.68	4.80%	8.40%	1.68
8	0.4583	40.00	0.4785	42.42	4.41%	6.07%	2.43
9	0.5833	55.00	0.7138	70.66	22.38%	28.48%	15.66
Square Average :					17.16%	24.52%	10.21

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.10448	0.18	4.48%	N/A	0.18
2	0.1	0	0.07978	-0.81	-20.22%	N/A	-0.81
3	0.1	0	0.07531	-0.99	-24.69%	N/A	-0.99
4	0.3	8	0.31237	8.49	4.12%	6.19%	0.49
5	0.3	8	0.26741	6.70	-10.86%	-16.30%	-1.30
6	0.3	8	0.3057	8.23	1.90%	2.85%	0.23
7	0.5	16	0.50976	16.39	1.95%	2.44%	0.39
8	0.5	16	0.47853	15.14	-4.29%	-5.37%	-0.86
9	0.5	16	0.4168	12.67	-16.64%	-20.80%	-3.33
Square Average:					12.81%	11.40%	1.32

Table A.2 : Results of $\pm 5\%$ noise clipped test data from $\pm 5\%$ noise trained network

Data Clipping	Training Noise	Test Noise	Data File
Yes	0%	10%	Training

Training Model #	esired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.250	15	0.3024	21.29	20.96%	41.91%	6.29
2	0.375	30	0.4054	33.65	8.12%	12.18%	3.65
3	0.500	45	0.5020	45.23	0.39%	0.52%	0.23
4	0.625	60	0.6316	60.79	1.06%	1.32%	0.79
5	0.750	75	0.7476	74.71	-0.32%	-0.38%	-0.29
6	0.875	90	0.9225	95.70	5.43%	6.33%	5.70
7	0.250	15	0.2379	13.55	-4.83%	-9.66%	-1.45
8	0.375	30	0.3868	31.41	3.14%	4.71%	1.41
9	0.500	45	0.5832	54.98	16.64%	22.19%	9.98
10	0.625	60	0.7042	69.50	12.67%	15.83%	9.50
11	0.750	75	0.9179	95.15	22.39%	26.86%	20.15
12	0.875	90	0.8429	86.14	-3.67%	-4.28%	-3.86
13	0.250	15	0.3090	22.08	23.61%	47.22%	7.08
14	0.375	30	0.3090	22.08	-17.59%	-26.39%	-7.92
15	0.500	45	0.5525	51.30	10.51%	14.01%	6.30
16	0.625	60	0.6775	66.30	8.40%	10.49%	6.30
17	0.750	75	0.8019	81.23	6.92%	8.31%	6.23
18	0.875	90	0.9898	103.77	13.11%	15.30%	13.77
19	0.375	30	0.9792	102.50	161.12%	241.68%	72.50
20	0.625	60	0.4980	44.77	-20.31%	-25.39%	-15.23
21	0.875	90	0.9795	102.54	11.94%	13.93%	12.54
Square Average:					37.35%	56.20%	17.95

Training Model #	esired output		Actual output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.09335	-0.27	-6.65%	N/A	-0.27
2	0.1	0	0.09468	-0.21	-5.32%	N/A	-0.21
3	0.1	0	0.09603	-0.16	-3.98%	N/A	-0.16
4	0.1	0	0.14319	1.73	43.19%	N/A	1.73
5	0.1	0	0.12853	1.14	28.53%	N/A	1.14
6	0.1	0	0.02145	-3.14	-78.55%	N/A	-3.14
7	0.3	8	0.3057	8.23	1.90%	2.85%	0.23
8	0.3	8	0.29911	7.96	-0.30%	-0.45%	-0.04
9	0.3	8	0.28937	7.57	-3.54%	-5.31%	-0.43
10	0.3	8	0.30902	8.36	3.01%	4.51%	0.36
11	0.3	8	0.29584	7.83	-1.39%	-2.08%	-0.17
12	0.3	8	0.35042	10.02	16.81%	25.21%	2.02
13	0.6	20	0.51757	16.70	-13.74%	-16.49%	-3.30
14	0.6	20	0.6058	20.23	0.97%	1.16%	0.23
15	0.6	20	0.5641	18.56	-5.98%	-7.18%	-1.44
16	0.6	20	0.62062	20.82	3.44%	4.12%	0.82
17	0.6	20	0.6058	20.23	0.97%	1.16%	0.23
18	0.6	20	0.54867	17.95	-8.55%	-10.27%	-2.05
19	0.9	32	0.91306	32.52	1.45%	1.63%	0.52
20	0.9	32	0.84492	29.80	-6.12%	-6.88%	-2.20
21	0.9	32	0.89841	31.94	-0.18%	-0.20%	-0.06
Square Average:					21.38%	8.93%	1.41

Figure A.3 : Results of ± 10% noise clipped training data from noise-free trained network

Data Clipping	Training Noise	Test Noise	Data File
Yes	0%	10%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.1656	4.87	-43.23%	-75.65%	-15.13
2	0.4583	40.00	0.5370	49.44	17.18%	23.62%	9.45
3	0.5833	55.00	0.4980	44.77	-14.62%	-18.60%	-10.23
4	0.2917	20.00	0.1700	5.39	-41.74%	-73.03%	-14.61
5	0.4583	40.00	0.9563	99.76	108.66%	149.42%	59.76
6	0.5833	55.00	0.5137	46.64	-11.94%	-15.19%	-8.36
7	0.2917	20.00	0.3157	22.89	8.24%	14.42%	2.88
8	0.4583	40.00	0.6943	68.32	51.50%	70.81%	28.32
9	0.5833	55.00	0.7202	71.42	23.47%	29.87%	16.42
Square Average :					46.36%	67.19%	24.40

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.19806	3.92	98.06%	N/A	3.92
2	0.1	0	0.07864	-0.85	-21.36%	N/A	-0.85
3	0.1	0	0.15922	2.37	59.22%	N/A	2.37
4	0.3	8	0.53315	17.33	77.72%	116.58%	9.33
5	0.3	8	0.14707	1.88	-50.98%	-76.47%	-6.12
6	0.3	8	0.32595	9.04	8.65%	12.98%	1.04
7	0.5	16	0.587	19.48	17.40%	21.75%	3.48
8	0.5	16	0.46685	14.67	-6.63%	-8.29%	-1.33
9	0.5	16	0.54092	17.64	8.18%	10.23%	1.64
Square Average:					50.23%	58.10%	4.27

Figure A.4 : Results of ± 10% noise clipped test data from noise-free trained network

Data Clipping	Training Noise	Test Noise	Data File
Yes	5%	10%	Training

Training Model #	esired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.250	15	0.2958	20.50	18.34%	36.67%	5.50
2	0.375	30	0.4707	41.49	25.53%	38.29%	11.49
3	0.500	45	0.3868	31.41	-22.65%	-30.20%	-13.59
4	0.625	60	0.7447	74.36	19.14%	23.93%	14.36
5	0.750	75	0.8278	84.34	10.38%	12.45%	9.34
6	0.875	90	0.6775	66.30	-22.57%	-26.34%	-23.70
7	0.250	15	0.2295	12.54	-8.19%	-16.37%	-2.46
8	0.375	30	0.3905	31.86	4.13%	6.19%	1.86
9	0.500	45	0.5641	52.69	12.82%	17.09%	7.69
10	0.625	60	0.7592	76.11	21.48%	26.85%	16.11
11	0.750	75	0.8256	84.07	10.08%	12.09%	9.07
12	0.875	90	0.8093	82.11	-7.51%	-8.76%	-7.89
13	0.250	15	0.2830	18.96	13.20%	26.40%	3.96
14	0.375	30	0.4630	40.56	23.46%	35.18%	10.56
15	0.500	45	0.4475	38.70	-10.51%	-14.01%	-6.30
16	0.625	60	0.8117	82.40	29.87%	37.34%	22.40
17	0.750	75	0.7417	74.00	-1.11%	-1.33%	-1.00
18	0.875	90	0.9478	98.73	8.32%	9.70%	8.73
19	0.375	30	0.8510	87.12	126.92%	190.39%	57.12
20	0.625	60	0.3329	24.94	-46.74%	-58.43%	-35.06
21	0.875	90	0.6775	66.30	-22.57%	-26.34%	-23.70
Square Average:					33.77%	49.12%	18.87

Training Model #	esired output		Actual output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.10595	0.24	5.95%	N/A	0.24
2	0.1	0	0.08946	-0.42	-10.55%	N/A	-0.42
3	0.1	0	0.10744	0.30	7.44%	N/A	0.30
4	0.1	0	0.08694	-0.52	-13.06%	N/A	-0.52
5	0.1	0	0.10895	0.36	8.94%	N/A	0.36
6	0.1	0	0.10895	0.36	8.94%	N/A	0.36
7	0.3	8	0.31912	8.76	6.37%	9.56%	0.76
8	0.3	8	0.28299	7.32	-5.67%	-8.51%	-0.68
9	0.3	8	0.28937	7.57	-3.54%	-5.31%	-0.43
10	0.3	8	0.30902	8.36	3.01%	4.51%	0.36
11	0.3	8	0.26741	6.70	-10.86%	-16.30%	-1.30
12	0.3	8	0.28617	7.45	-4.61%	-6.92%	-0.55
13	0.6	20	0.51757	16.70	-13.74%	-16.49%	-3.30
14	0.6	20	0.59078	19.63	-1.54%	-1.84%	-0.37
15	0.6	20	0.60952	20.38	1.59%	1.90%	0.38
16	0.6	20	0.59831	19.93	-0.28%	-0.34%	-0.07
17	0.6	20	0.62795	21.12	4.66%	5.59%	1.12
18	0.6	20	0.49236	15.69	-17.94%	-21.53%	-4.31
19	0.9	32	0.80928	28.37	-10.08%	-11.34%	-3.63
20	0.9	32	0.92359	32.94	2.62%	2.95%	0.94
21	0.9	32	0.84492	29.80	-6.12%	-6.88%	-2.20
Square Average:					8.31%	9.95%	1.60

Table A.5 : Results of ± 10% noise clipped training data from ± 5% noise trained network

Data Clipping	Training Noise	Test Noise	Data File
Yes	5%	10%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.3398	25.78	16.50%	28.87%	5.78
2	0.4583	40.00	0.7233	71.80	57.82%	79.51%	31.80
3	0.5833	55.00	0.7649	76.79	31.13%	39.62%	21.79
4	0.2917	20.00	0.1981	8.77	-32.10%	-56.17%	-11.24
5	0.4583	40.00	0.8344	85.13	82.07%	112.84%	45.13
6	0.5833	55.00	0.5525	51.30	-5.27%	-6.71%	-3.69
7	0.2917	20.00	0.3191	23.29	9.40%	16.45%	3.29
8	0.4583	40.00	0.7944	80.33	73.33%	100.84%	40.33
9	0.5833	55.00	0.7356	73.28	26.12%	33.24%	18.28
Square Average :					45.37%	63.42%	25.10

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.11047	0.42	10.47%	N/A	0.42
2	0.1	0	0.05223	-1.91	-47.77%	N/A	-1.91
3	0.1	0	0.08694	-0.52	-13.06%	N/A	-0.52
4	0.3	8	0.32595	9.04	8.65%	12.98%	1.04
5	0.3	8	0.14904	1.96	-50.32%	-75.48%	-6.04
6	0.3	8	0.40169	12.07	33.90%	50.84%	4.07
7	0.5	16	0.5641	18.56	12.82%	16.02%	2.56
8	0.5	16	0.46685	14.67	-6.63%	-8.29%	-1.33
9	0.5	16	0.47853	15.14	-4.29%	-5.37%	-0.86
Square Average:					26.97%	38.31%	2.73

Table A.6 : Results of $\pm 10\%$ noise clipped test data from $\pm 5\%$ noise trained network

Data Clipping	Training Noise	Test Noise	Data File
Yes	10%	10%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.1251	0.01	-57.13%	-99.96%	-20.00
2	0.4583	40.00	0.4244	35.93	-7.39%	-10.17%	-4.07
3	0.5833	55.00	0.5370	49.44	-7.93%	-10.09%	-5.55
4	0.2917	20.00	0.3398	25.78	16.50%	28.87%	5.78
5	0.4583	40.00	0.5718	53.61	24.76%	34.04%	13.62
6	0.5833	55.00	0.6567	63.80	12.58%	16.01%	8.80
7	0.2917	20.00	0.2436	14.24	-16.48%	-28.83%	-5.77
8	0.4583	40.00	0.5487	50.84	19.72%	27.11%	10.84
9	0.5833	55.00	0.7705	77.46	32.09%	40.84%	22.46
Square Average :					26.06%	41.74%	12.47

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.19806	3.92	98.06%	N/A	3.92
2	0.1	0	0.10448	0.18	4.48%	N/A	0.18
3	0.1	0	0.10895	0.36	8.94%	N/A	0.36
4	0.3	8	0.3057	8.23	1.90%	2.85%	0.23
5	0.3	8	0.25238	6.10	-15.87%	-23.81%	-1.90
6	0.3	8	0.26133	6.45	-12.89%	-19.33%	-1.55
7	0.5	16	0.62795	21.12	25.59%	31.99%	5.12
8	0.5	16	0.56794	18.72	13.59%	16.98%	2.72
9	0.5	16	0.50976	16.39	1.95%	2.44%	0.39
Square Average:					34.93%	19.44%	2.48

Table A.7 : Results of ± 10% noise clipped test data from ± 10% noise trained network

Data Clipping	Training Noise	Test Noise	Data File
Yes	10%	5%	Training

Training Model #	esired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.250	15	0.1656	4.87	-33.76%	-67.53%	-10.13
2	0.375	30	0.3612	28.34	-3.69%	-5.53%	-1.66
3	0.500	45	0.4824	42.89	-3.51%	-4.69%	-2.11
4	0.625	60	0.5564	51.77	-10.98%	-13.72%	-8.23
5	0.750	75	0.7649	76.79	1.99%	2.38%	1.79
6	0.875	90	0.8697	89.37	-0.60%	-0.70%	-0.63
7	0.250	15	0.2134	10.61	-14.65%	-29.30%	-4.39
8	0.375	30	0.4092	34.11	9.13%	13.69%	4.11
9	0.500	45	0.4785	42.42	-4.29%	-5.73%	-2.58
10	0.625	60	0.6531	63.38	4.50%	5.63%	3.38
11	0.750	75	0.6879	67.55	-8.28%	-9.93%	-7.45
12	0.875	90	0.7138	70.66	-18.42%	-21.49%	-19.34
13	0.250	15	0.2894	19.72	15.75%	31.50%	4.72
14	0.375	30	0.3540	27.48	-5.60%	-8.41%	-2.52
15	0.500	45	0.5409	49.91	8.18%	10.91%	4.91
16	0.625	60	0.5908	55.89	-5.48%	-6.84%	-4.11
17	0.750	75	0.7233	71.80	-3.56%	-4.27%	-3.20
18	0.875	90	0.7621	76.45	-12.91%	-15.06%	-13.55
19	0.375	30	0.6206	59.47	65.50%	98.25%	29.47
20	0.625	60	0.7264	72.17	16.23%	20.29%	12.17
21	0.875	90	0.7202	71.42	-17.69%	-20.64%	-18.58
Square Average:					18.84%	29.63%	10.44

Training Model #	esired output		Actual output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.12507	1.00	25.07%	N/A	1.00
2	0.1	0	0.08211	-0.72	-17.89%	N/A	-0.72
3	0.1	0	0.08694	-0.52	-13.06%	N/A	-0.52
4	0.1	0	0.11676	0.67	16.76%	N/A	0.67
5	0.1	0	0.11358	0.54	13.58%	N/A	0.54
6	0.1	0	0.08946	-0.42	-10.55%	N/A	-0.42
7	0.3	8	0.37205	10.88	24.02%	36.02%	2.88
8	0.3	8	0.24653	5.86	-17.82%	-26.73%	-2.14
9	0.3	8	0.31912	8.76	6.37%	9.56%	0.76
10	0.3	8	0.27048	6.82	-9.84%	-14.76%	-1.18
11	0.3	8	0.30902	8.36	3.01%	4.51%	0.36
12	0.3	8	0.34334	9.73	14.45%	21.67%	1.73
13	0.6	20	0.67747	23.10	12.91%	15.49%	3.10
14	0.6	20	0.62429	20.97	4.05%	4.86%	0.97
15	0.6	20	0.68426	23.37	14.04%	16.85%	3.37
16	0.6	20	0.60952	20.38	1.59%	1.90%	0.38
17	0.6	20	0.59831	19.93	-0.28%	-0.34%	-0.07
18	0.6	20	0.6058	20.23	0.97%	1.16%	0.23
19	0.9	32	0.86431	30.57	-3.97%	-4.46%	-1.43
20	0.9	32	0.86247	30.50	-4.17%	-4.69%	-1.50
21	0.9	32	0.89256	31.70	-0.83%	-0.93%	-0.30
Square Average:					12.65%	15.07%	1.50

Table A.8 : Results of ± 5% noise clipped training data from ± 10% noise trained network

Data Clipping	Training Noise	Test Noise	Data File
Yes	10%	5%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.1136	-1.37	-61.06%	-106.85%	-21.37
2	0.4583	40.00	0.4902	43.83	6.97%	9.58%	3.83
3	0.5833	55.00	0.5332	48.98	-8.60%	-10.94%	-6.02
4	0.2917	20.00	0.1530	3.37	-47.53%	-83.18%	-16.64
5	0.4583	40.00	0.5603	52.23	22.25%	30.59%	12.23
6	0.5833	55.00	0.5946	56.35	1.93%	2.45%	1.35
7	0.2917	20.00	0.2082	9.98	-28.63%	-50.10%	-10.02
8	0.4583	40.00	0.4168	35.02	-9.06%	-12.45%	-4.98
9	0.5833	55.00	0.6671	65.06	14.37%	18.30%	10.06
Square Average :					29.28%	49.99%	11.37

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.15922	2.37	59.22%	N/A	2.37
2	0.1	0	0.08094	-0.76	-19.06%	N/A	-0.76
3	0.1	0	0.10744	0.30	7.44%	N/A	0.30
4	0.3	8	0.49805	15.92	66.02%	99.02%	7.92
5	0.3	8	0.16132	2.45	-46.23%	-69.34%	-5.55
6	0.3	8	0.27983	7.19	-6.72%	-10.09%	-0.81
7	0.5	16	0.62062	20.82	24.12%	30.16%	4.82
8	0.5	16	0.59455	19.78	18.91%	23.64%	3.78
9	0.5	16	0.40545	12.22	-18.91%	-23.64%	-3.78
Square Average:					36.15%	52.82%	4.11

Table A.9 : Results of $\pm 5\%$ noise clipped test data from $\pm 10\%$ noise trained network

Data Clipping	Training Noise	Test Noise	Data File
Yes	10%	0%	Training

Training Model #	esired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.250	15	0.1656	4.87	-33.76%	-67.53%	-10.13
2	0.375	30	0.3612	28.34	-3.69%	-5.53%	-1.66
3	0.500	45	0.4824	42.89	-3.51%	-4.69%	-2.11
4	0.625	60	0.5564	51.77	-10.98%	-13.72%	-8.23
5	0.750	75	0.7649	76.79	1.99%	2.38%	1.79
6	0.875	90	0.8697	89.37	-0.60%	-0.70%	-0.63
7	0.250	15	0.2134	10.61	-14.65%	-29.30%	-4.39
8	0.375	30	0.4092	34.11	9.13%	13.69%	4.11
9	0.500	45	0.4785	42.42	-4.29%	-5.73%	-2.58
10	0.625	60	0.6531	63.38	4.50%	5.63%	3.38
11	0.750	75	0.6876	67.52	-8.32%	-9.98%	-7.48
12	0.875	90	0.7138	70.66	-18.42%	-21.49%	-19.34
13	0.250	15	0.2894	19.72	15.75%	31.50%	4.72
14	0.375	30	0.3540	27.48	-5.60%	-8.41%	-2.52
15	0.500	45	0.5409	49.91	8.18%	10.91%	4.91
16	0.625	60	0.5908	55.89	-5.48%	-6.84%	-4.11
17	0.750	75	0.7233	71.80	-3.56%	-4.27%	-3.20
18	0.875	90	0.7621	76.45	-12.91%	-15.06%	-13.55
19	0.375	30	0.6206	59.47	65.50%	98.25%	29.47
20	0.625	60	0.7264	72.17	16.23%	20.29%	12.17
21	0.875	90	0.7202	71.42	-17.69%	-20.64%	-18.58
Square Average:					18.85%	29.63%	10.45

Training Model #	esired output		Actual output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.12507	1.00	25.07%	N/A	1.00
2	0.1	0	0.08211	-0.72	-17.89%	N/A	-0.72
3	0.1	0	0.06894	-1.24	-31.06%	N/A	-1.24
4	0.1	0	0.11676	0.67	16.76%	N/A	0.67
5	0.1	0	0.11358	0.54	13.58%	N/A	0.54
6	0.1	0	0.08946	-0.42	-10.55%	N/A	-0.42
7	0.3	8	0.37205	10.88	24.02%	36.02%	2.88
8	0.3	8	0.24653	5.86	-17.82%	-26.73%	-2.14
9	0.3	8	0.31912	8.76	6.37%	9.56%	0.76
10	0.3	8	0.27048	6.82	-9.84%	-14.76%	-1.18
11	0.3	8	0.30902	8.36	3.01%	4.51%	0.36
12	0.3	8	0.34334	9.73	14.45%	21.67%	1.73
13	0.6	20	0.67747	23.10	12.91%	15.49%	3.10
14	0.6	20	0.62429	20.97	4.05%	4.86%	0.97
15	0.6	20	0.68426	23.37	14.04%	16.85%	3.37
16	0.6	20	0.60952	20.38	1.59%	1.90%	0.38
17	0.6	20	0.59831	19.93	-0.28%	-0.34%	-0.07
18	0.6	20	0.6058	20.23	0.97%	1.16%	0.23
19	0.9	32	0.86431	30.57	-3.97%	-4.46%	-1.43
20	0.9	32	0.86247	30.50	-4.17%	-4.69%	-1.50
21	0.9	32	0.89256	31.70	-0.83%	-0.93%	-0.30
Square Average:					14.07%	15.07%	1.52

Table A.10 : Results of noise-free clipped training data from ± 10% noise trained network

Data Clipping	Training Noise	Test Noise	Data File
Yes	10%	0%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.1136	-1.37	-61.06%	-106.85%	-21.37
2	0.4583	40.00	0.4902	43.83	6.97%	9.58%	3.83
3	0.5833	55.00	0.5332	48.98	-8.60%	-10.94%	-6.02
4	0.2917	20.00	0.1530	3.37	-47.53%	-83.18%	-16.64
5	0.4583	40.00	0.5603	52.23	22.25%	30.59%	12.23
6	0.5833	55.00	0.5946	56.35	1.93%	2.45%	1.35
7	0.2917	20.00	0.2082	9.98	-28.63%	-50.10%	-10.02
8	0.4583	40.00	0.4168	35.02	-9.06%	-12.45%	-4.98
9	0.5833	55.00	0.6671	65.06	14.37%	18.30%	10.06
Square Average :					29.28%	49.99%	11.37

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.15922	2.37	59.22%	N/A	2.37
2	0.1	0	0.08094	-0.76	-19.06%	N/A	-0.76
3	0.1	0	0.10744	0.30	7.44%	N/A	0.30
4	0.3	8	0.49805	15.92	66.02%	99.02%	7.92
5	0.3	8	0.16132	2.45	-46.23%	-69.34%	-5.55
6	0.3	8	0.27983	7.19	-6.72%	-10.09%	-0.81
7	0.5	16	0.62062	20.82	24.12%	30.16%	4.82
8	0.5	16	0.59455	19.78	18.91%	23.64%	3.78
9	0.5	16	0.40545	12.22	-18.91%	-23.64%	-3.78
Square Average:					36.15%	52.82%	4.11

Table A.11 : Results of noise-free clipped test data from ± 10% noise trained network

Data Clipping	Training Noise	Test Noise	Data File
No	0%	5%	Training

Training Model #	esired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.250	15	0.2644	16.72	5.74%	11.49%	1.72
2	0.375	30	0.3504	27.05	-6.55%	-9.83%	-2.95
3	0.500	45	0.4980	44.77	-0.39%	-0.52%	-0.23
4	0.625	60	0.6706	65.47	7.30%	9.12%	5.47
5	0.750	75	0.9358	97.30	24.78%	29.73%	22.30
6	0.875	90	0.9470	98.64	8.23%	9.60%	8.64
7	0.250	15	0.2613	16.36	4.53%	9.06%	1.36
8	0.375	30	0.4017	33.20	7.12%	10.68%	3.20
9	0.500	45	0.5215	47.58	4.29%	5.73%	2.58
10	0.625	60	0.6809	66.71	8.94%	11.18%	6.71
11	0.750	75	0.8141	82.69	8.54%	10.25%	7.69
12	0.875	90	0.8816	90.79	0.76%	0.88%	0.79
13	0.250	15	0.2958	20.50	18.34%	36.67%	5.50
14	0.375	30	0.3057	21.68	-18.48%	-27.72%	-8.32
15	0.500	45	0.7264	72.17	45.29%	60.38%	27.17
16	0.625	60	0.5718	53.61	-8.52%	-10.65%	-6.39
17	0.750	75	0.6531	63.38	-12.92%	-15.50%	-11.62
18	0.875	90	0.7969	80.63	-8.92%	-10.41%	-9.37
19	0.375	30	0.3469	26.62	-7.50%	-11.25%	-3.38
20	0.625	60	0.6777	66.33	8.44%	10.55%	6.33
21	0.875	90	0.9768	102.22	11.64%	13.58%	12.22
Square Average:					14.42%	19.99%	9.85

Training Model #	esired output		Actual output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.12337	0.93	23.37%	N/A	0.93
2	0.1	0	0.09739	-0.10	-2.61%	N/A	-0.10
3	0.1	0	0.08329	-0.67	-16.71%	N/A	-0.67
4	0.1	0	0.07864	-0.85	-21.36%	N/A	-0.85
5	0.1	0	0.08946	-0.42	-10.55%	N/A	-0.42
6	0.1	0	0.0546	-1.82	-45.40%	N/A	-1.82
7	0.3	8	0.30239	8.10	0.80%	1.20%	0.10
8	0.3	8	0.27983	7.19	-6.72%	-10.09%	-0.81
9	0.3	8	0.28617	7.45	-4.61%	-6.92%	-0.55
10	0.3	8	0.30224	8.09	0.75%	1.12%	0.09
11	0.3	8	0.28937	7.57	-3.54%	-5.31%	-0.43
12	0.3	8	0.28299	7.32	-5.67%	-8.51%	-0.68
13	0.6	20	0.57559	19.02	-4.07%	-4.88%	-0.98
14	0.6	20	0.64958	21.98	8.26%	9.92%	1.98
15	0.6	20	0.56794	18.72	-5.34%	-6.41%	-1.28
16	0.6	20	0.57177	18.87	-4.71%	-5.65%	-1.13
17	0.6	20	0.5448	17.79	-9.20%	-11.04%	-2.21
18	0.6	20	0.61324	20.53	2.21%	2.65%	0.53
19	0.9	32	0.88642	31.46	-1.51%	-1.70%	-0.54
20	0.9	32	0.86613	30.65	-3.76%	-4.23%	-1.35
21	0.9	32	0.88162	31.26	-2.04%	-2.30%	-0.74
Square Average:					13.48%	6.34%	1.04

Table A.12 : Results of ± 5% noise non-clipped training data from noise-free trained network

Data Clipping	Training Noise	Test Noise	Data File
No	5%	5%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.2991	20.89	2.54%	4.44%	0.89
2	0.4583	40.00	0.3979	32.75	-13.17%	-18.11%	-7.24
3	0.5833	55.00	0.5137	46.64	-11.94%	-15.19%	-8.36
4	0.2917	20.00	0.1981	8.77	-32.10%	-56.17%	-11.24
5	0.4583	40.00	0.5215	47.58	13.78%	18.95%	7.58
6	0.5833	55.00	0.6458	62.49	10.71%	13.64%	7.50
7	0.2917	20.00	0.2160	10.92	-25.95%	-45.40%	-9.08
8	0.4583	40.00	0.4630	40.56	1.02%	1.40%	0.56
9	0.5833	55.00	0.6976	68.71	19.60%	24.94%	13.72
Square Average :					17.37%	27.82%	8.39

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.10448	0.18	4.48%	N/A	0.18
2	0.1	0	0.09877	-0.05	-1.23%	N/A	-0.05
3	0.1	0	0.12853	1.14	28.53%	N/A	1.14
4	0.3	8	0.33983	9.59	13.28%	19.91%	1.59
5	0.3	8	0.28937	7.57	-3.54%	-5.31%	-0.43
6	0.3	8	0.30239	8.10	0.80%	1.20%	0.10
7	0.5	16	0.50195	16.08	0.39%	0.49%	0.08
8	0.5	16	0.49805	15.92	-0.39%	-0.49%	-0.08
9	0.5	16	0.49024	15.61	-1.95%	-2.44%	-0.39
Square Average:					10.69%	8.49%	0.69

Table A.13 : Results of $\pm 5\%$ noise non-clipped test data from $\pm 5\%$ noise trained network

Data Clipping	Training Noise	Test Noise	Data File
No	0%	10%	Training

Training Model #	esired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.250	15	0.2494	14.93	-0.22%	-0.44%	-0.07
2	0.375	30	0.3024	21.29	-19.36%	-29.04%	-8.71
3	0.500	45	0.4206	35.47	-15.88%	-21.17%	-9.53
4	0.625	60	0.7813	78.76	25.01%	31.27%	18.76
5	0.750	75	0.9040	93.48	20.53%	24.64%	18.48
6	0.875	90	0.9026	93.31	3.16%	3.68%	3.31
7	0.250	15	0.2108	10.29	-15.69%	-31.39%	-4.71
8	0.375	30	0.4168	35.02	11.15%	16.72%	5.02
9	0.500	45	0.4552	39.62	-8.96%	-11.95%	-5.38
10	0.625	60	0.5679	53.15	-9.13%	-11.41%	-6.85
11	0.750	75	0.7840	79.08	4.53%	5.44%	4.08
12	0.875	90	0.9775	102.31	11.72%	13.67%	12.31
13	0.250	15	0.5020	45.23	100.78%	201.56%	30.23
14	0.375	30	0.8661	88.94	130.97%	196.45%	58.94
15	0.500	45	0.4863	43.36	-2.73%	-3.64%	-1.64
16	0.625	60	0.8019	81.23	28.31%	35.39%	21.23
17	0.750	75	0.9536	99.43	27.15%	32.58%	24.43
18	0.875	90	0.8117	82.40	-7.24%	-8.44%	-7.60
19	0.375	30	0.3469	26.62	-7.50%	-11.25%	-3.38
20	0.625	60	0.9515	99.18	52.24%	65.30%	39.18
21	0.875	90	0.8322	84.87	-4.89%	-5.70%	-5.13
Square Average:					40.29%	65.71%	19.80

Training Model #	esired output		Actual output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.1413	1.65	41.28%	N/A	1.65
2	0.1	0	0.1074	0.30	7.44%	N/A	0.30
3	0.1	0	0.1152	0.61	15.16%	N/A	0.61
4	0.1	0	0.1375	1.50	37.53%	N/A	1.50
5	0.1	0	0.1030	0.12	3.02%	N/A	0.12
6	0.1	0	0.1184	0.74	18.39%	N/A	0.74
7	0.3	8	0.3757	11.03	25.24%	37.85%	3.03
8	0.3	8	0.2295	5.18	-23.49%	-35.23%	-2.82
9	0.3	8	0.3024	8.10	0.80%	1.20%	0.10
10	0.3	8	0.2644	6.57	-11.88%	-17.82%	-1.43
11	0.3	8	0.2736	6.94	-8.81%	-13.21%	-1.06
12	0.3	8	0.1981	3.92	-33.98%	-50.97%	-4.08
13	0.6	20	0.5908	19.63	-1.54%	-1.84%	-0.37
14	0.6	20	0.5983	19.93	-0.28%	-0.34%	-0.07
15	0.6	20	0.6169	20.68	2.82%	3.39%	0.68
16	0.6	20	0.6388	21.55	6.47%	7.77%	1.55
17	0.6	20	0.5332	17.33	-11.14%	-13.37%	-2.67
18	0.6	20	0.5832	19.33	-2.80%	-3.36%	-0.67
19	0.9	32	0.9067	32.27	0.74%	0.83%	0.27
20	0.9	32	0.7759	27.04	-13.78%	-15.51%	-4.96
21	0.9	32	0.9268	33.07	2.98%	3.35%	1.07
Square Average:					17.85%	20.47%	1.94

Table A.14 : Results of ± 10% noise non-clipped training data from noise-free trained network

Data Clipping	Training Noise	Test Noise	Data File
No	0%	10%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.2214	11.56	-24.12%	-42.20%	-8.44
2	0.4583	40.00	0.4824	42.89	5.26%	7.24%	2.90
3	0.5833	55.00	0.6775	66.30	16.15%	20.55%	11.30
4	0.2917	20.00	0.2862	19.34	-1.90%	-3.32%	-0.66
5	0.4583	40.00	0.6132	58.59	33.81%	46.49%	18.59
6	0.5833	55.00	0.5718	53.61	-1.98%	-2.52%	-1.38
7	0.2917	20.00	0.2830	18.96	-2.99%	-5.23%	-1.05
8	0.4583	40.00	0.5254	48.04	14.63%	20.12%	8.05
9	0.5833	55.00	0.9830	102.95	68.52%	87.20%	47.96
Square Average :					27.76%	37.22%	18.02

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.11839	0.74	18.39%	N/A	0.74
2	0.1	0	0.10159	0.06	1.59%	N/A	0.06
3	0.1	0	0.08094	-0.76	-19.06%	N/A	-0.76
4	0.3	8	0.33983	9.59	13.28%	19.91%	1.59
5	0.3	8	0.28937	7.57	-3.54%	-5.31%	-0.43
6	0.3	8	0.27983	7.19	-6.72%	-10.09%	-0.81
7	0.5	16	0.5448	17.79	8.96%	11.20%	1.79
8	0.5	16	0.48243	15.30	-3.51%	-4.39%	-0.70
9	0.5	16	0.32939	9.18	-34.12%	-42.65%	-6.82
Square Average:					15.62%	20.37%	2.47

Table A.15 : Results of $\pm 10\%$ noise non-clipped test data from noise-free trained network

Data Clipping	Training Noise	Test Noise	Data File
No	5%	10%	Training

Training Model #	esired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.250	15	0.2241	11.89	-10.38%	-20.76%	-3.11
2	0.375	30	0.3191	23.29	-14.90%	-22.35%	-6.71
3	0.500	45	0.3329	24.94	-33.43%	-44.57%	-20.06
4	0.625	60	0.4707	41.49	-24.68%	-30.85%	-18.51
5	0.750	75	0.7840	79.08	4.53%	5.44%	4.08
6	0.875	90	0.7107	70.28	-18.78%	-21.91%	-19.72
7	0.250	15	0.2553	15.64	2.14%	4.28%	0.64
8	0.375	30	0.6352	61.23	69.39%	104.09%	31.23
9	0.500	45	0.4513	39.16	-9.73%	-12.98%	-5.84
10	0.625	60	0.7233	71.80	15.73%	19.66%	11.80
11	0.750	75	0.8941	92.29	19.21%	23.05%	17.29
12	0.875	90	0.8366	85.39	-4.39%	-5.13%	-4.61
13	0.250	15	0.2644	16.72	5.74%	11.49%	1.72
14	0.375	30	0.4359	37.31	16.24%	24.36%	7.31
15	0.500	45	0.5487	50.84	9.73%	12.98%	5.84
16	0.625	60	0.5448	50.38	-12.83%	-16.04%	-9.62
17	0.750	75	0.9493	98.92	26.57%	31.89%	23.92
18	0.875	90	0.9080	93.96	3.77%	4.40%	3.96
19	0.375	30	0.4321	36.85	15.22%	22.83%	6.85
20	0.625	60	0.7476	74.71	19.62%	24.52%	14.71
21	0.875	90	0.8187	83.25	-6.43%	-7.50%	-6.75
Square Average:					21.73%	30.64%	13.37

Training Model #	esired output		Actual output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.1303	1.21	30.29%	N/A	1.21
2	0.1	0	0.1184	0.74	18.39%	N/A	0.74
3	0.1	0	0.1030	0.12	3.02%	N/A	0.12
4	0.1	0	0.1303	1.21	30.29%	N/A	1.21
5	0.1	0	0.0988	-0.05	-1.23%	N/A	-0.05
6	0.1	0	0.1268	1.07	26.79%	N/A	1.07
7	0.3	8	0.2524	6.10	-15.87%	-23.81%	-1.90
8	0.3	8	0.1859	3.44	-38.02%	-57.03%	-4.56
9	0.3	8	0.2958	7.83	-1.39%	-2.08%	-0.17
10	0.3	8	0.2926	7.70	-2.47%	-3.70%	-0.30
11	0.3	8	0.2767	7.07	-7.77%	-11.65%	-0.93
12	0.3	8	0.2830	7.32	-5.67%	-8.51%	-0.68
13	0.6	20	0.5254	17.01	-12.44%	-14.93%	-2.99
14	0.6	20	0.5409	17.64	-9.85%	-11.82%	-2.36
15	0.6	20	0.6169	20.68	2.82%	3.39%	0.68
16	0.6	20	0.6206	20.82	3.44%	4.12%	0.82
17	0.6	20	0.4746	14.99	-20.89%	-25.07%	-5.01
18	0.6	20	0.5979	19.92	-0.34%	-0.41%	-0.08
19	0.9	32	0.8783	31.13	-2.41%	-2.71%	-0.87
20	0.9	32	0.8587	30.35	-4.59%	-5.16%	-1.65
21	0.9	32	0.9310	33.24	3.44%	3.87%	1.24
Square Average:					16.10%	18.46%	1.91

Table A.16 : Results of ± 10% noise non-clipped training data from ± 5% noise trained network

Data Clipping	Training Noise	Test Noise	Data File
No	5%	10%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.2038	9.45	-30.15%	-52.76%	-10.55
2	0.4583	40.00	0.4824	42.89	5.26%	7.24%	2.90
3	0.5833	55.00	0.6095	58.14	4.50%	5.72%	3.15
4	0.2917	20.00	0.2214	11.56	-24.12%	-42.20%	-8.44
5	0.4583	40.00	0.5176	47.11	12.93%	17.78%	7.11
6	0.5833	55.00	0.6280	60.35	7.65%	9.74%	5.36
7	0.2917	20.00	0.3090	22.08	5.94%	10.39%	2.08
8	0.4583	40.00	0.3831	30.97	-16.42%	-22.57%	-9.03
9	0.5833	55.00	0.8490	86.88	45.55%	57.97%	31.88
Square Average :					21.46%	31.69%	12.39

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.12853	1.14	28.53%	N/A	1.14
2	0.1	0	0.08449	-0.62	-15.51%	N/A	-0.62
3	0.1	0	0.09603	-0.16	-3.98%	N/A	-0.16
4	0.3	8	0.34342	9.74	14.47%	21.71%	1.74
5	0.3	8	0.29584	7.83	-1.39%	-2.08%	-0.17
6	0.3	8	0.31237	8.49	4.12%	6.19%	0.49
7	0.5	16	0.47074	14.83	-5.85%	-7.32%	-1.17
8	0.5	16	0.51757	16.70	3.51%	4.39%	0.70
9	0.5	16	0.37938	11.18	-24.12%	-30.16%	-4.82
Square Average:					14.63%	15.79%	1.83

Table A.17 : Results of $\pm 10\%$ noise non-clipped test data from $\pm 5\%$ noise trained network

Data Clipping	Training Noise	Test Noise	Data File
No	10%	10%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.2268	12.21	-22.25%	-38.94%	-7.79
2	0.4583	40.00	0.3434	26.21	-25.07%	-34.47%	-13.79
3	0.5833	55.00	0.7009	69.11	20.16%	25.66%	14.11
4	0.2917	20.00	0.4054	33.65	39.00%	68.24%	13.65
5	0.4583	40.00	0.5641	52.69	23.08%	31.74%	12.70
6	0.5833	55.00	0.6876	67.52	17.89%	22.76%	12.52
7	0.2917	20.00	0.5409	49.91	85.44%	149.50%	29.91
8	0.4583	40.00	0.7009	69.11	52.93%	72.79%	29.11
9	0.5833	55.00	0.7621	76.45	30.65%	39.01%	21.45
Square Average :					40.75%	65.59%	18.73

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.12679	1.07	26.79%	N/A	1.07
2	0.1	0	0.09468	-0.21	-5.32%	N/A	-0.21
3	0.1	0	0.09468	-0.21	-5.32%	N/A	-0.21
4	0.3	8	0.3057	8.23	1.90%	2.85%	0.23
5	0.3	8	0.30239	8.10	0.80%	1.20%	0.10
6	0.3	8	0.23231	5.29	-22.56%	-33.85%	-2.71
7	0.5	16	0.4359	13.44	-12.82%	-16.02%	-2.56
8	0.5	16	0.42441	12.98	-15.12%	-18.90%	-3.02
9	0.5	16	0.4206	12.82	-15.88%	-19.85%	-3.18
Square Average:					14.65%	18.99%	1.96

Table A.18 : Results of $\pm 10\%$ noise non-clipped test data from $\pm 10\%$ noise trained network

Data Clipping	Training Noise	Test Noise	Data File
No	10%	5%	Training

Training Model #	esired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.250	15	0.2958	20.50	18.34%	36.67%	5.50
2	0.375	30	0.3540	27.48	-5.60%	-8.41%	-2.52
3	0.500	45	0.4902	43.83	-1.95%	-2.60%	-1.17
4	0.625	60	0.5293	48.51	-15.32%	-19.15%	-11.49
5	0.750	75	0.8164	82.97	8.85%	10.62%	7.97
6	0.875	90	0.8429	86.14	-3.67%	-4.28%	-3.86
7	0.250	15	0.3831	30.97	53.23%	106.45%	15.97
8	0.375	30	0.4513	39.16	20.35%	30.53%	9.16
9	0.500	45	0.4980	44.77	-0.39%	-0.52%	-0.23
10	0.625	60	0.7042	69.50	12.67%	15.83%	9.50
11	0.750	75	0.8366	85.39	11.54%	13.85%	10.39
12	0.875	90	0.7564	75.76	-13.56%	-15.82%	-14.24
13	0.250	15	0.5679	53.15	127.17%	254.35%	38.15
14	0.375	30	0.6496	62.95	73.22%	109.83%	32.95
15	0.500	45	0.5756	54.07	15.12%	20.16%	9.07
16	0.625	60	0.6316	60.79	1.06%	1.32%	0.79
17	0.750	75	0.7295	72.54	-2.73%	-3.28%	-2.46
18	0.875	90	0.8164	82.97	-6.70%	-7.81%	-7.03
19	0.375	30	0.6424	62.09	71.32%	106.97%	32.09
20	0.625	60	0.5908	55.89	-5.48%	-6.84%	-4.11
21	0.875	90	0.6352	61.23	-27.40%	-31.97%	-28.77
Square Average:					39.10%	70.58%	16.25

Training Model #	esired output		Actual output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.1334	1.34	33.39%	N/A	1.34
2	0.1	0	0.1217	0.87	21.69%	N/A	0.87
3	0.1	0	0.1002	0.01	0.17%	N/A	0.01
4	0.1	0	0.1200	0.80	20.03%	N/A	0.80
5	0.1	0	0.0845	-0.62	-15.51%	N/A	-0.62
6	0.1	0	0.0974	-0.10	-2.61%	N/A	-0.10
7	0.3	8	0.3576	10.30	19.19%	28.78%	2.30
8	0.3	8	0.3398	9.59	13.28%	19.91%	1.59
9	0.3	8	0.3090	8.36	3.01%	4.51%	0.36
10	0.3	8	0.2494	5.98	-16.85%	-25.28%	-2.02
11	0.3	8	0.2613	6.45	-12.89%	-19.33%	-1.55
12	0.3	8	0.3363	9.45	12.11%	18.17%	1.45
13	0.6	20	0.5603	18.41	-6.62%	-7.95%	-1.59
14	0.6	20	0.5293	17.17	-11.79%	-14.15%	-2.83
15	0.6	20	0.6388	21.55	6.47%	7.77%	1.55
16	0.6	20	0.6058	20.23	0.97%	1.16%	0.23
17	0.6	20	0.6316	21.26	5.27%	6.32%	1.26
18	0.6	20	0.6843	23.37	14.04%	16.85%	3.37
19	0.9	32	0.8732	30.93	-2.98%	-3.35%	-1.07
20	0.9	32	0.8864	31.46	-1.51%	-1.70%	-0.54
21	0.9	32	0.8800	31.20	-2.23%	-2.50%	-0.80
Square Average:					13.54%	14.74%	1.51

Table A.19 : Results of $\pm 5\%$ noise non-clipped training data from $\pm 10\%$ noise trained network

Data Clipping	Training Noise	Test Noise	Data File
No	10%	5%	Test

Test Model #	Desired output		Actual output		Angle error		
	raw	real degrees	raw	real degrees	raw %	real %	absolute degrees
1	0.2917	20.00	0.2436	14.24	-16.48%	-28.83%	-5.77
2	0.4583	40.00	0.4017	33.20	-12.35%	-16.99%	-6.79
3	0.5833	55.00	0.6388	61.66	9.52%	12.12%	6.66
4	0.2917	20.00	0.2323	12.88	-20.36%	-35.63%	-7.13
5	0.4583	40.00	0.3794	30.53	-17.22%	-23.68%	-9.47
6	0.5833	55.00	0.6460	62.52	10.75%	13.68%	7.53
7	0.2917	20.00	0.4785	42.42	64.05%	112.07%	22.42
8	0.4583	40.00	0.5718	53.61	24.76%	34.04%	13.62
9	0.5833	55.00	0.7476	74.71	28.17%	35.85%	19.72
Square Average :					27.59%	45.08%	12.46

Test Model #	Desired Output		Actual Output		Depth error		
	raw	real meters	raw	real meters	raw %	real %	absolute meters
1	0.1	0	0.10895	0.36	8.94%	N/A	0.36
2	0.1	0	0.10017	0.01	0.17%	N/A	0.01
3	0.1	0	0.12337	0.93	23.37%	N/A	0.93
4	0.3	8	0.35399	10.16	18.00%	26.99%	2.16
5	0.3	8	0.32939	9.18	9.80%	14.70%	1.18
6	0.3	8	0.25833	6.33	-13.89%	-20.84%	-1.67
7	0.5	16	0.45133	14.05	-9.73%	-12.17%	-1.95
8	0.5	16	0.4436	13.74	-11.28%	-14.10%	-2.26
9	0.5	16	0.413	12.52	-17.40%	-21.75%	-3.48
Square Average:					14.00%	19.14%	1.85

Table A.20 : Results of $\pm 5\%$ noise non-clipped test data from $\pm 10\%$ noise trained network

Appendix B

Fourier Time Transforms

In this appendix contour plots of the logarithm of the amplitude of the Fourier time transform of the dike models are included. Only a representative sample is given.

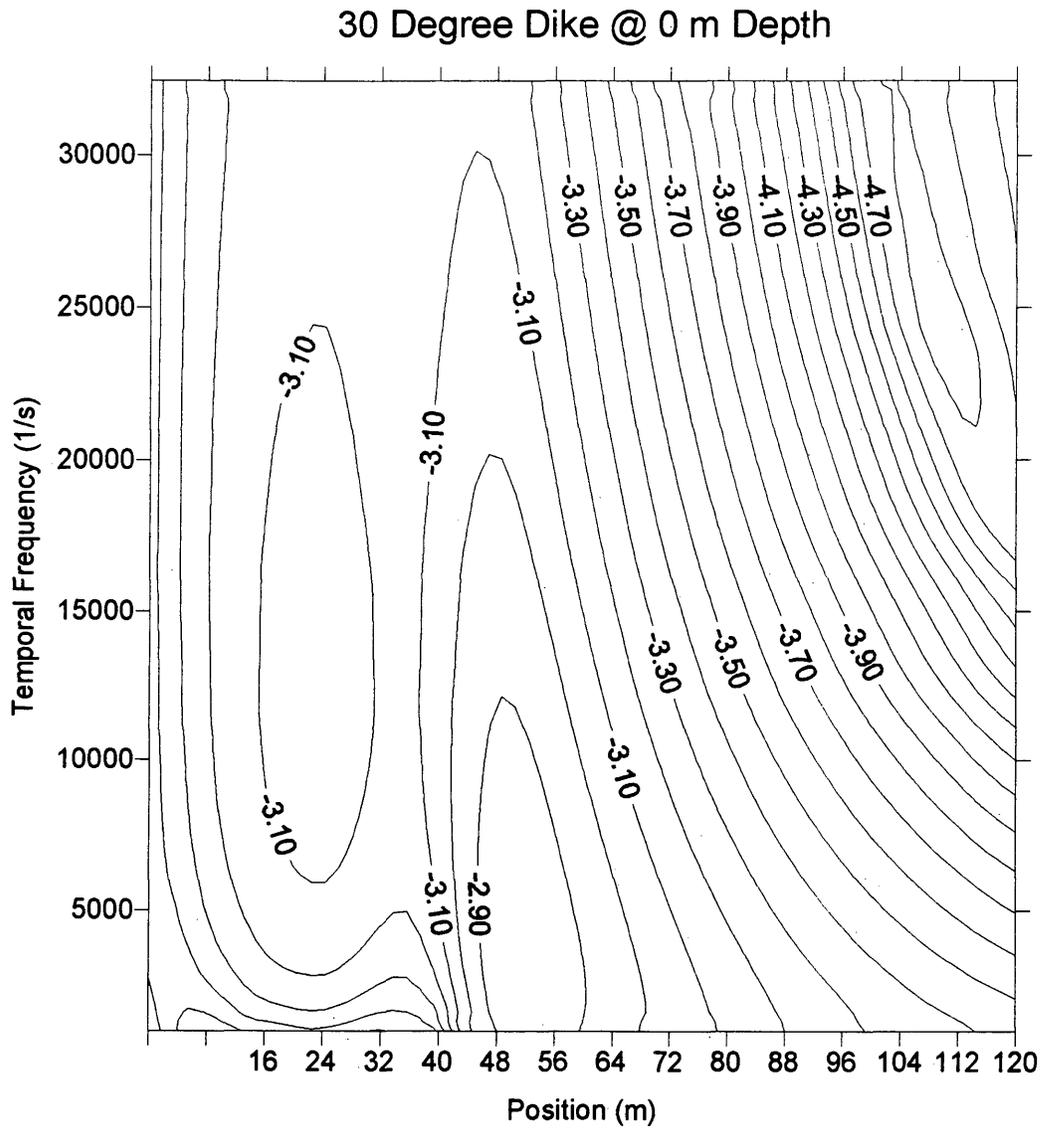


Figure B.1 : Temporal Transform for 30 degree dike model @ 0 m depth

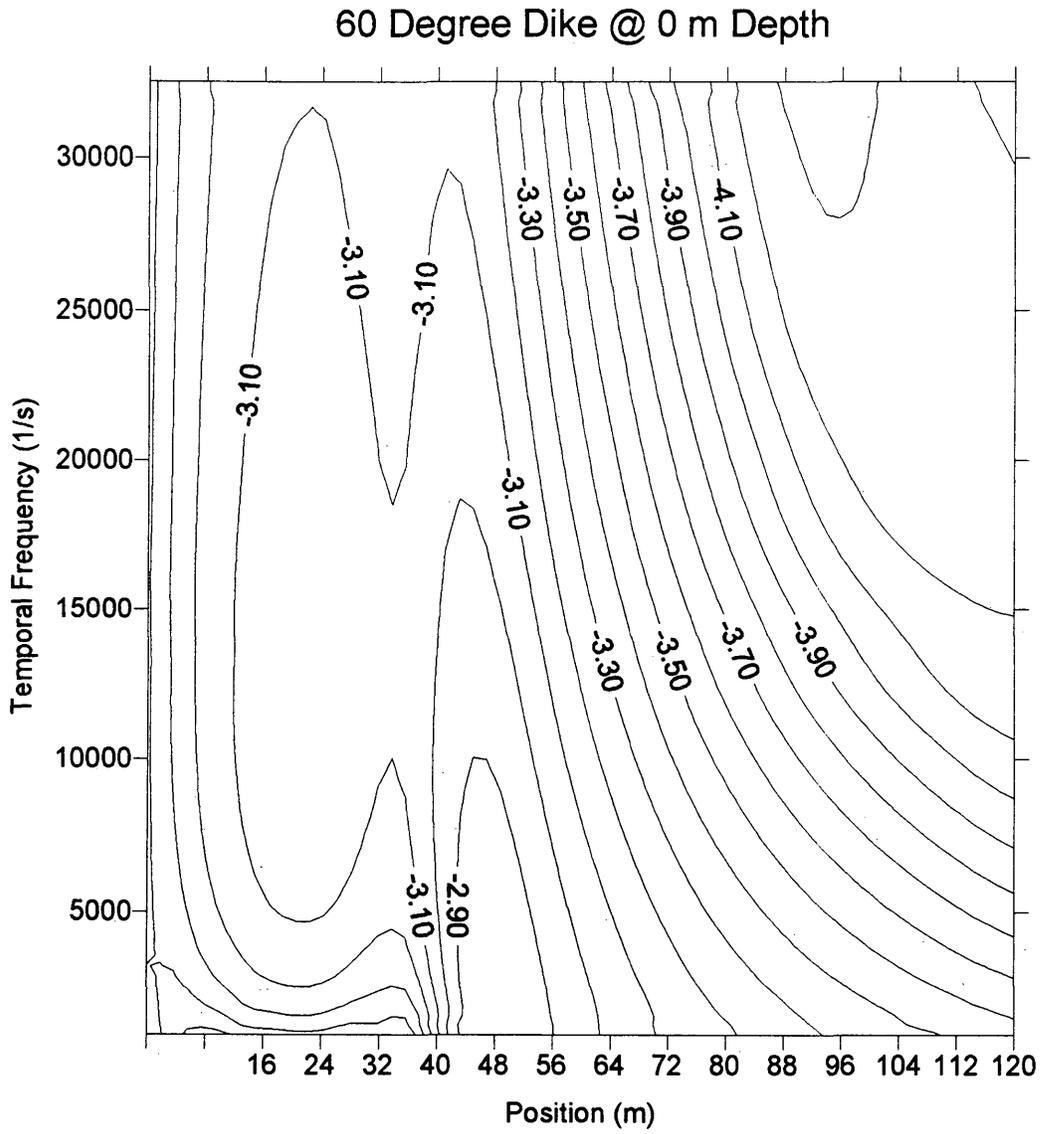


Figure B.2 : Temporal Transform for 60 degree dike model @ 0 depth

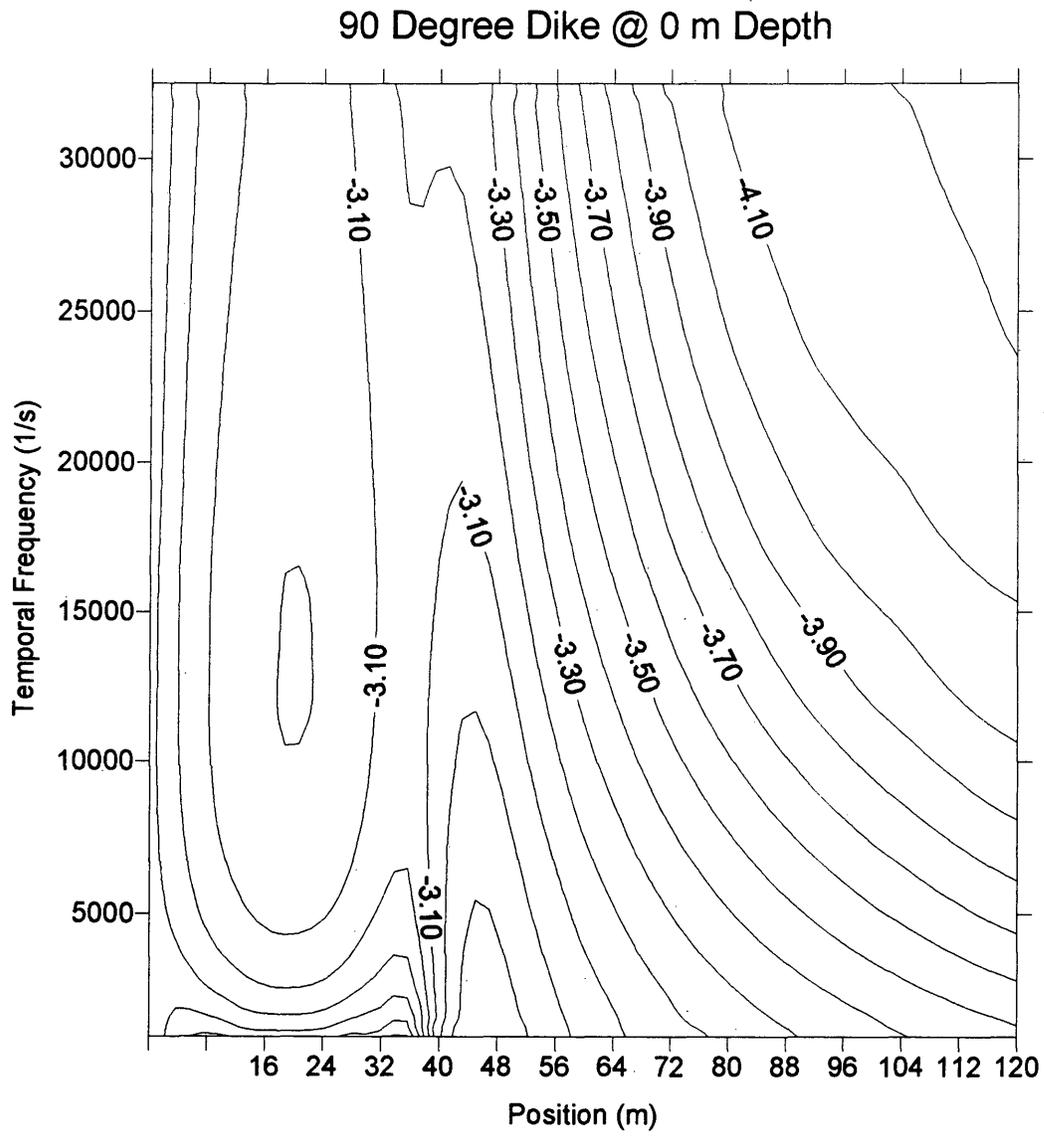


Figure B.3 : Temporal Transform for 90 degree dike model @ 0 m depth

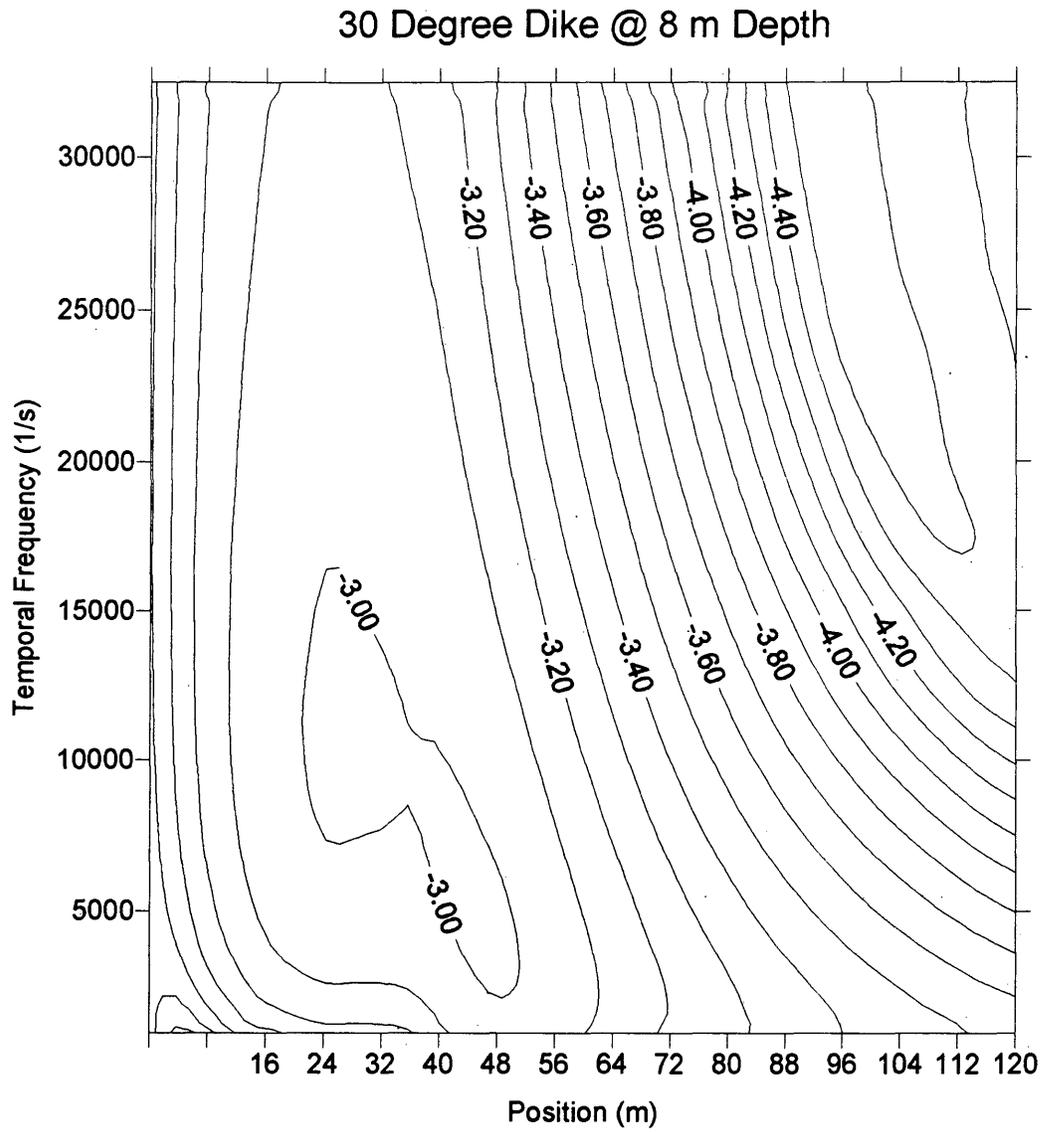


Figure B.4 : Temporal Transform for 30 degree dike model @ 8 m depth

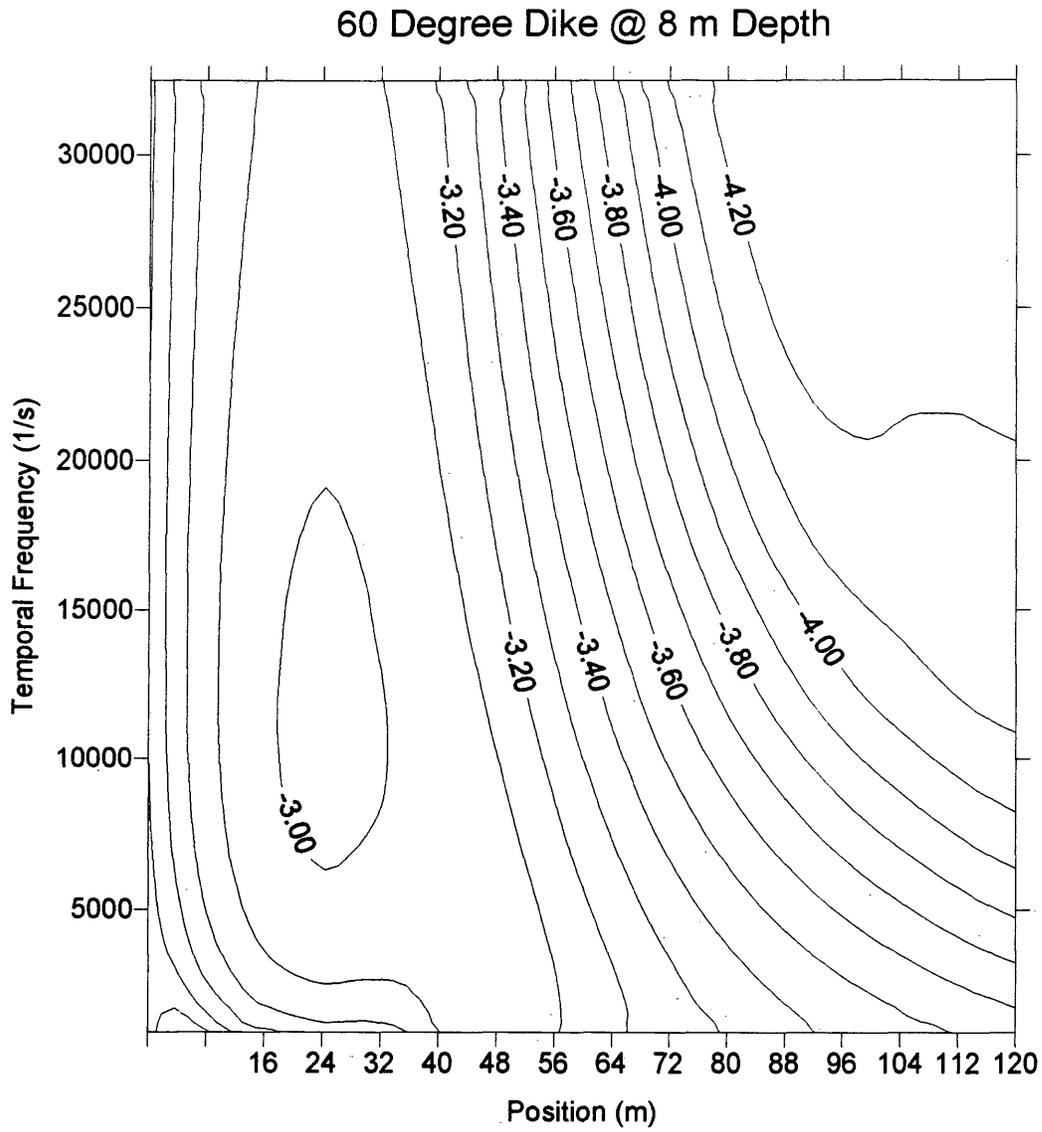


Figure B.5 : Temporal Transform for 60 degree dike model @ 8 m depth

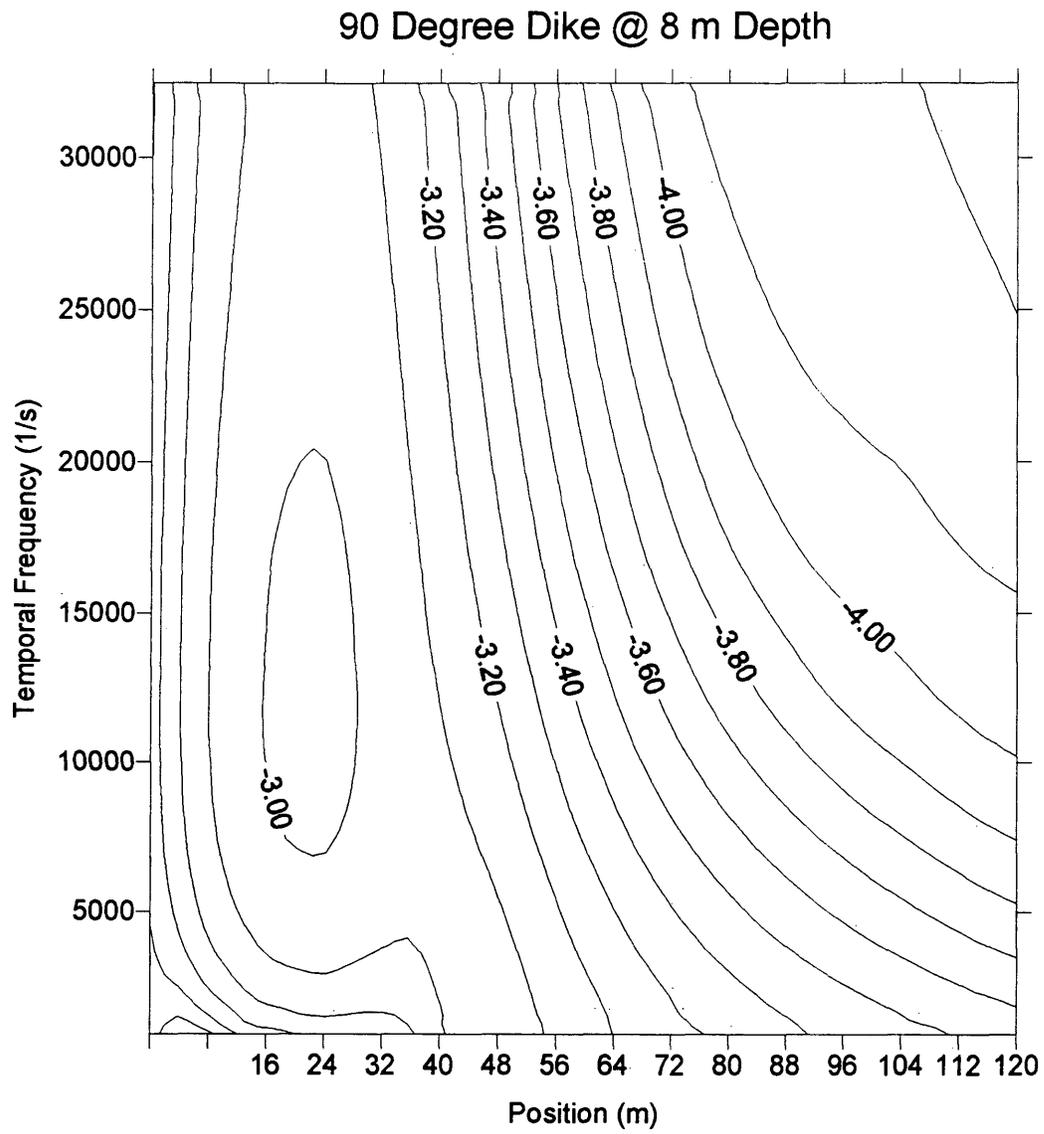


Figure B.6 : Temporal Transform for 90 degree dike model @ 8 m depth

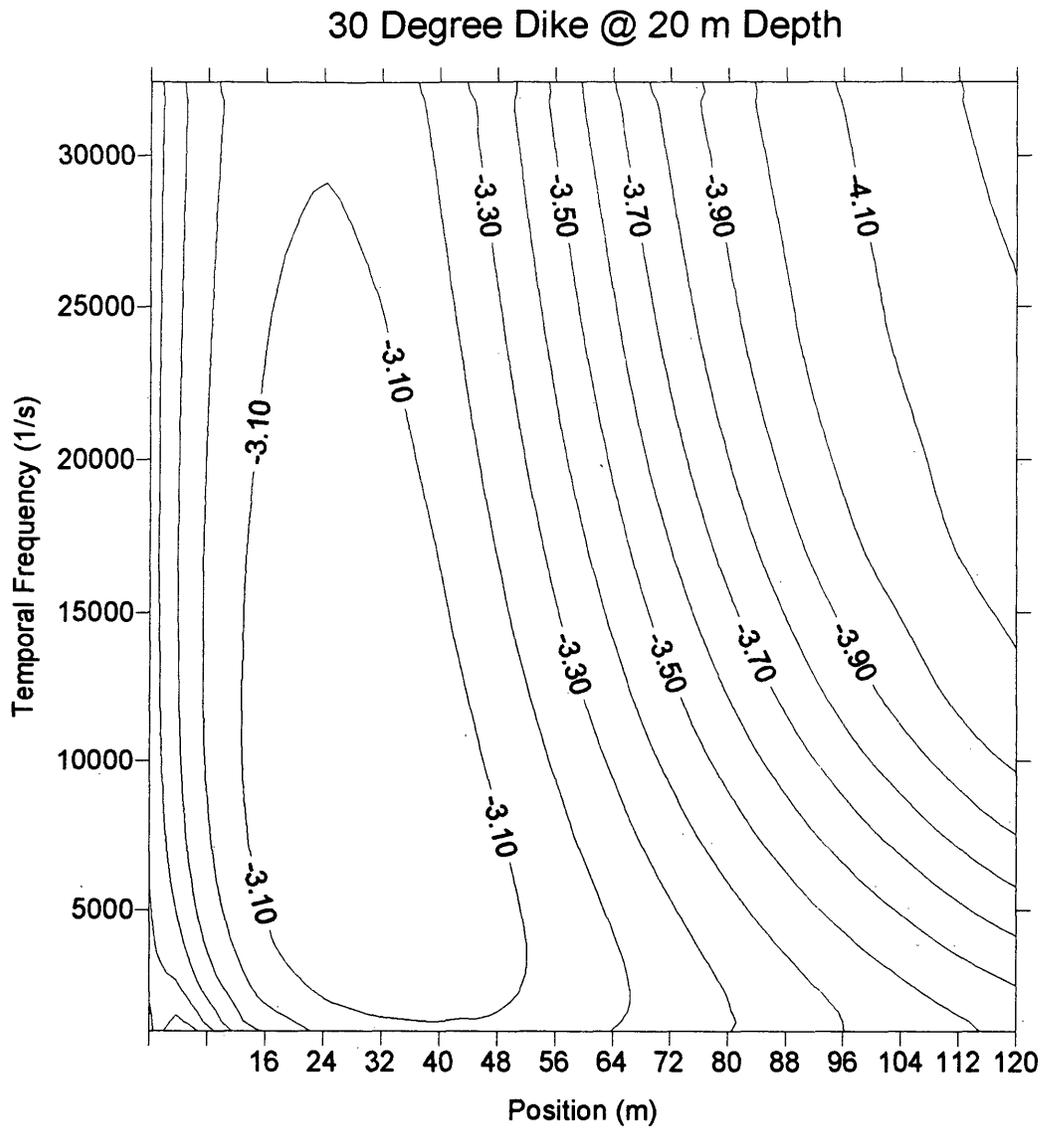


Figure B.7 : Temporal Transform for 30 degree dike model @ 20 m depth

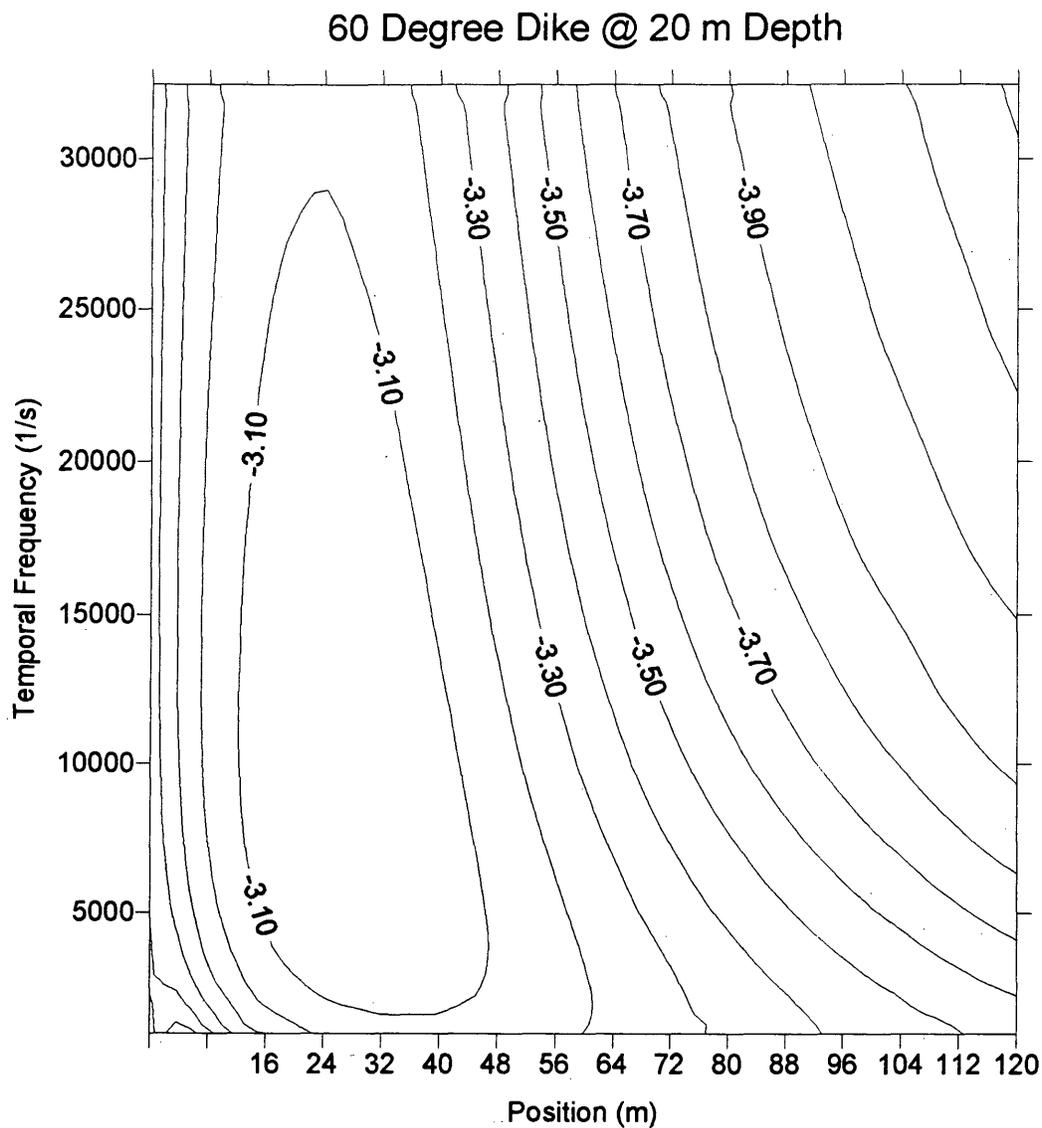


Figure B.8 : Temporal Transform for 60 degree dike model @ 20 m depth

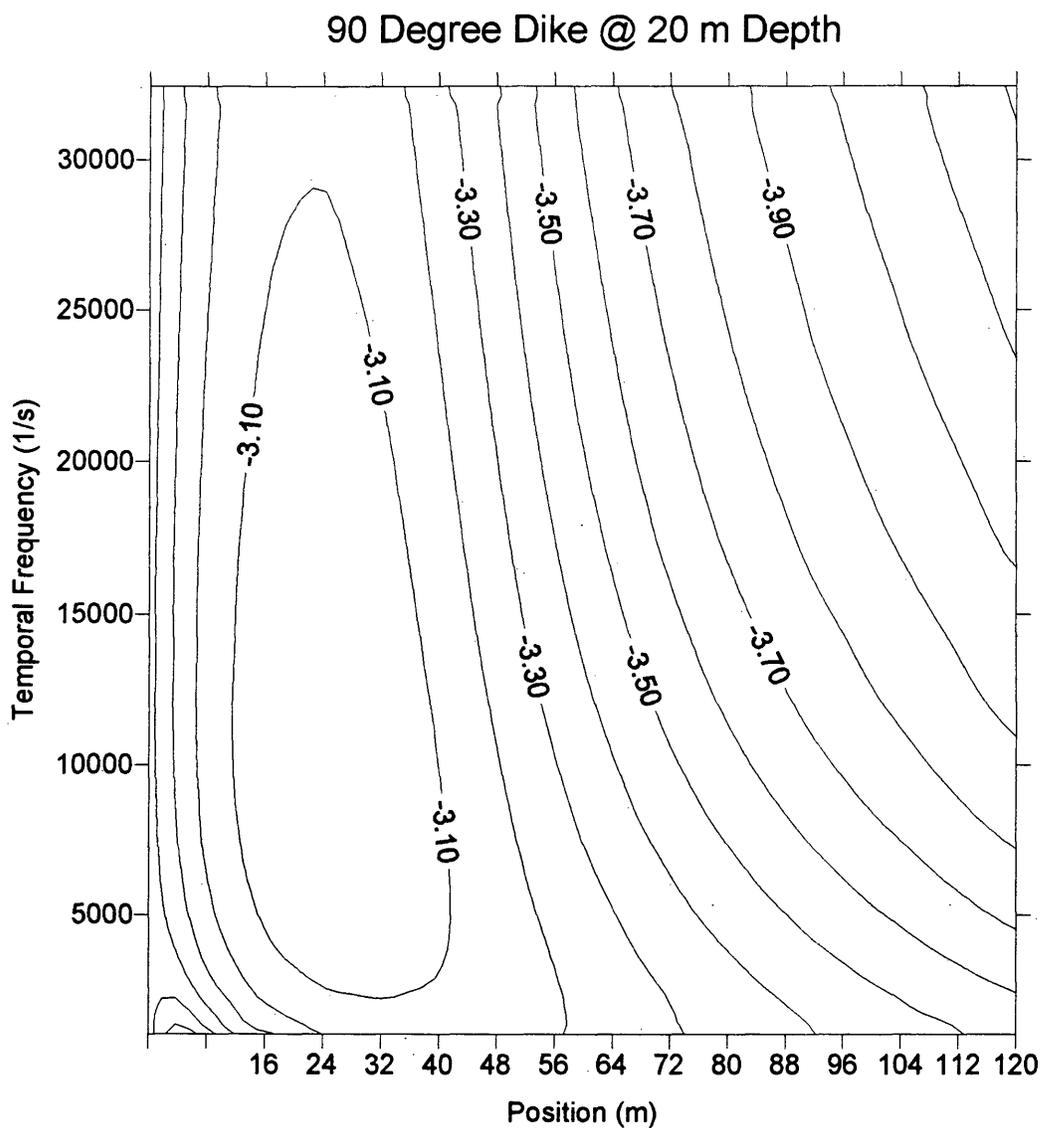


Figure B.9 : Temporal Transform for 90 degree dike model @ 20 m depth

Appendix C

Fourier Time and Space Transforms

In this appendix contour plots of the logarithm of the amplitude of the Fourier time and space transforms of the dike models are included. Only a representative sample is given.

30 Degree Dike @ 0 m Depth

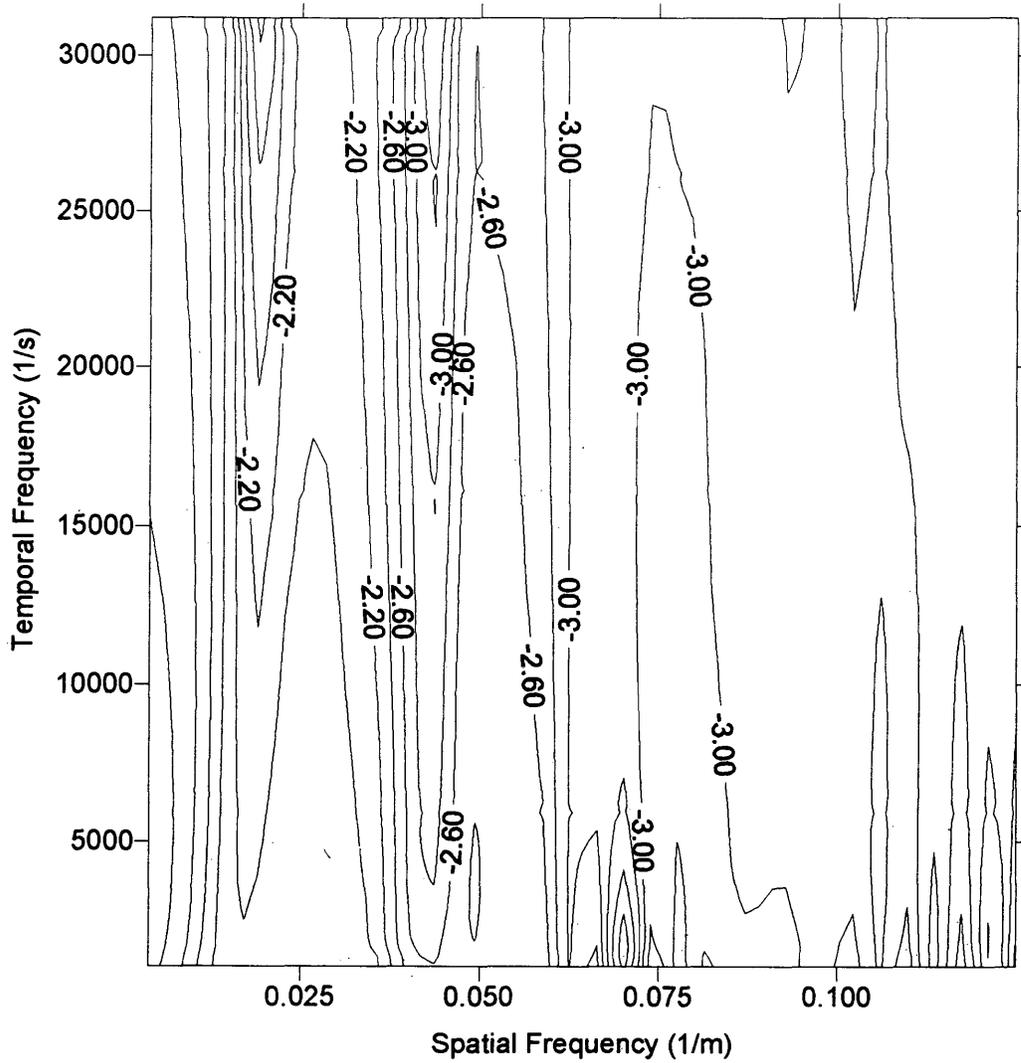


Figure C.1 : Spatial and Temporal Transform for 30 degree dike model @ 0 m depth

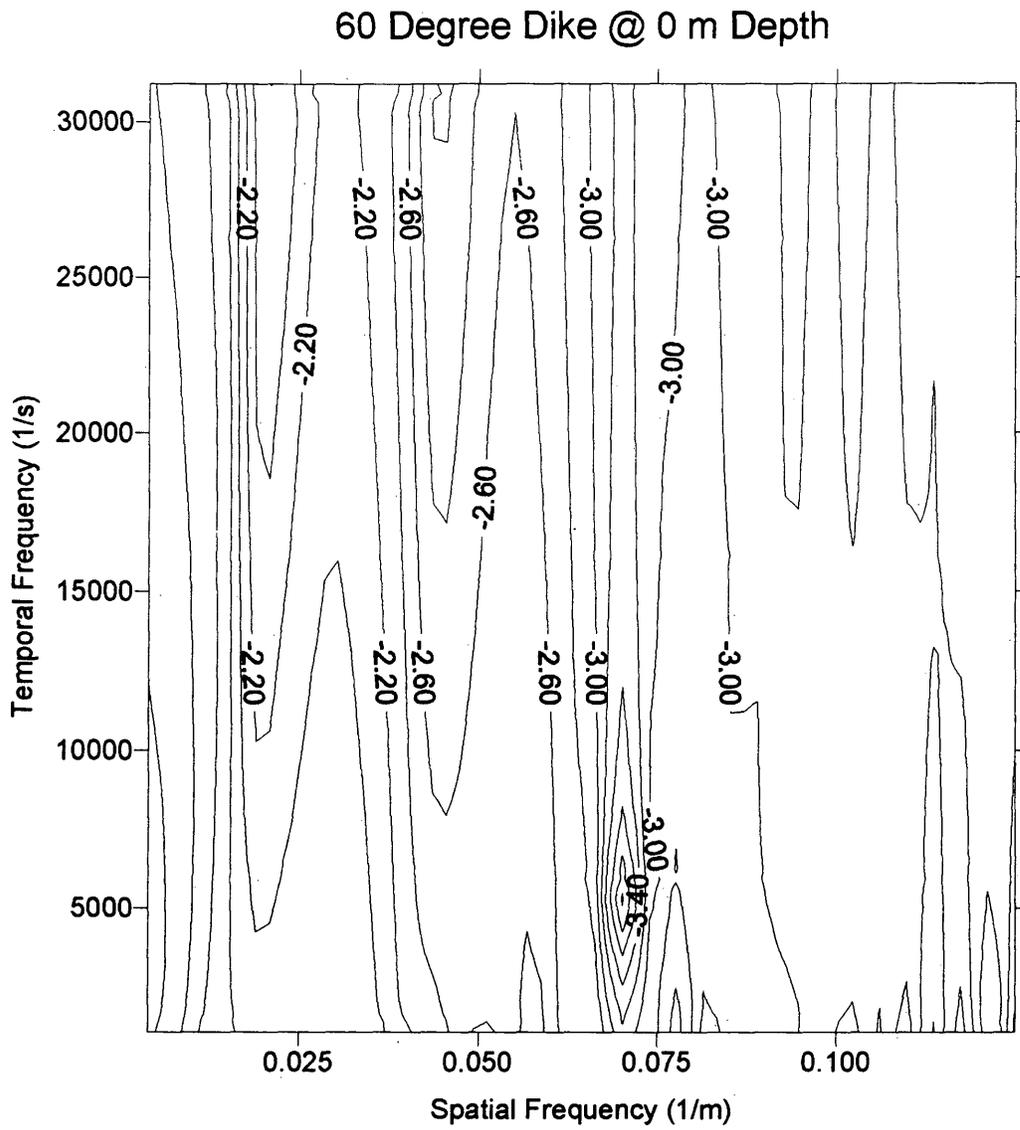


Figure C.2 : Spatial and Temporal Transform for 60 degree dike model @ 0 m depth

90 Degree Dike @ 0 m Depth

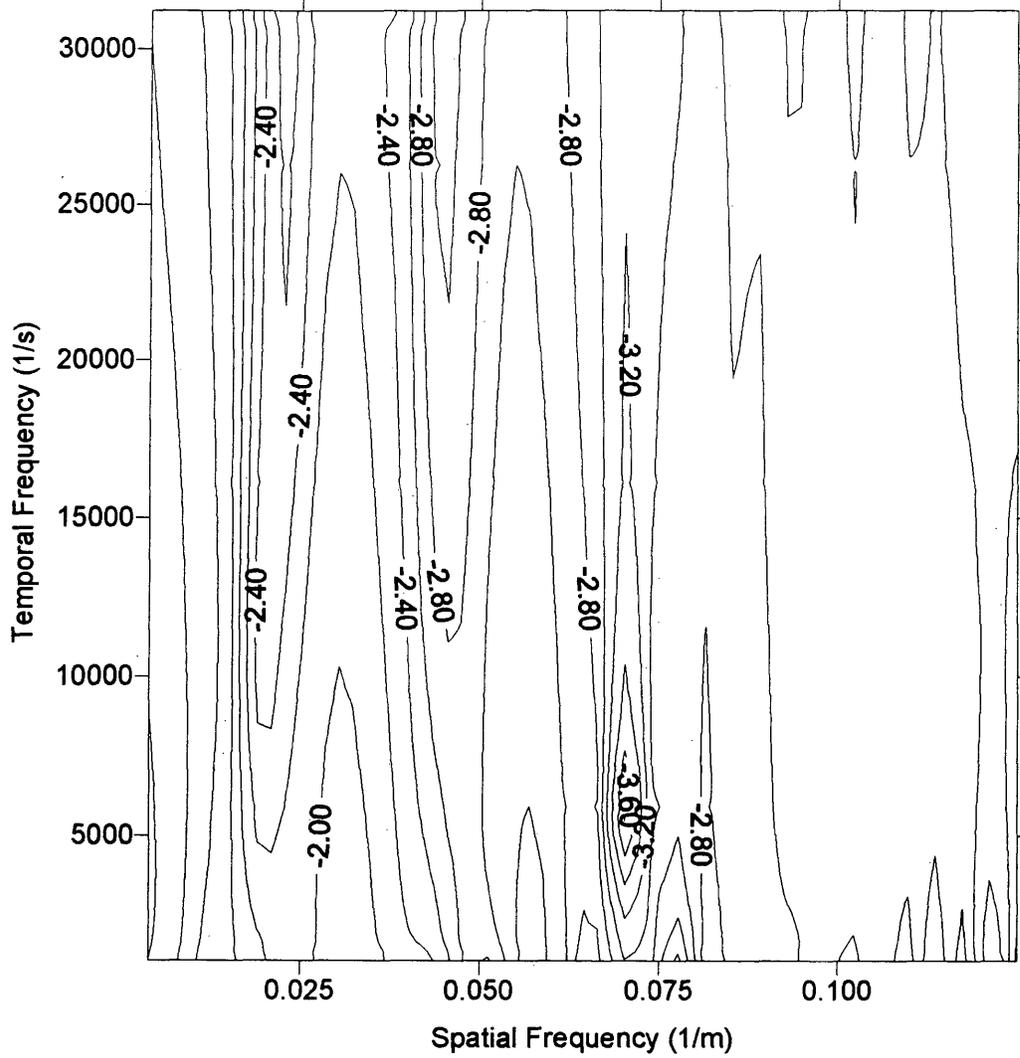


Figure C.3 : Spatial and Temporal Transform for 90 degree dike model @ 0 m depth

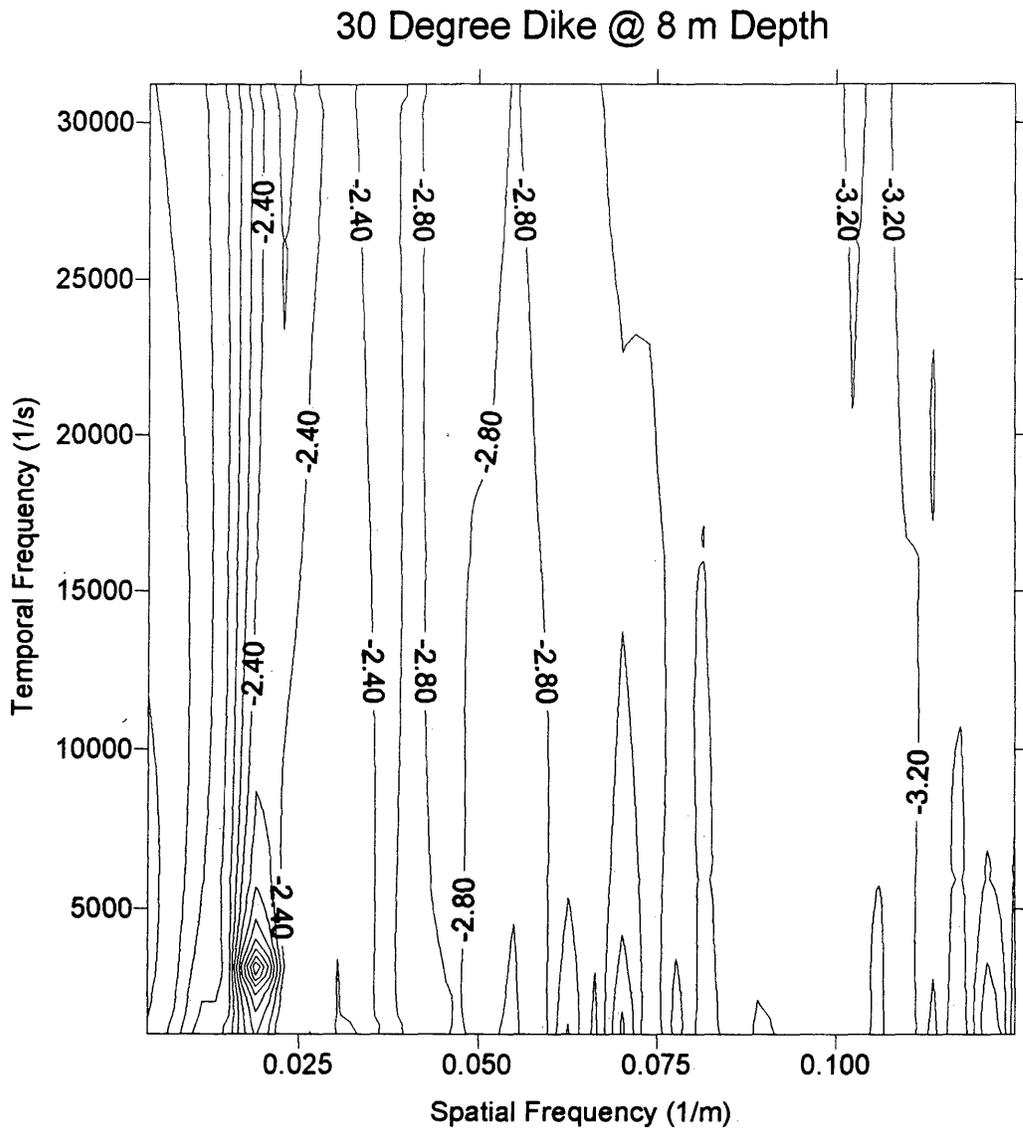


Figure C.4 : Spatial and Temporal Transform for 30 degree dike model @ 8 m depth

60 Degree Dike @ 8 m Depth

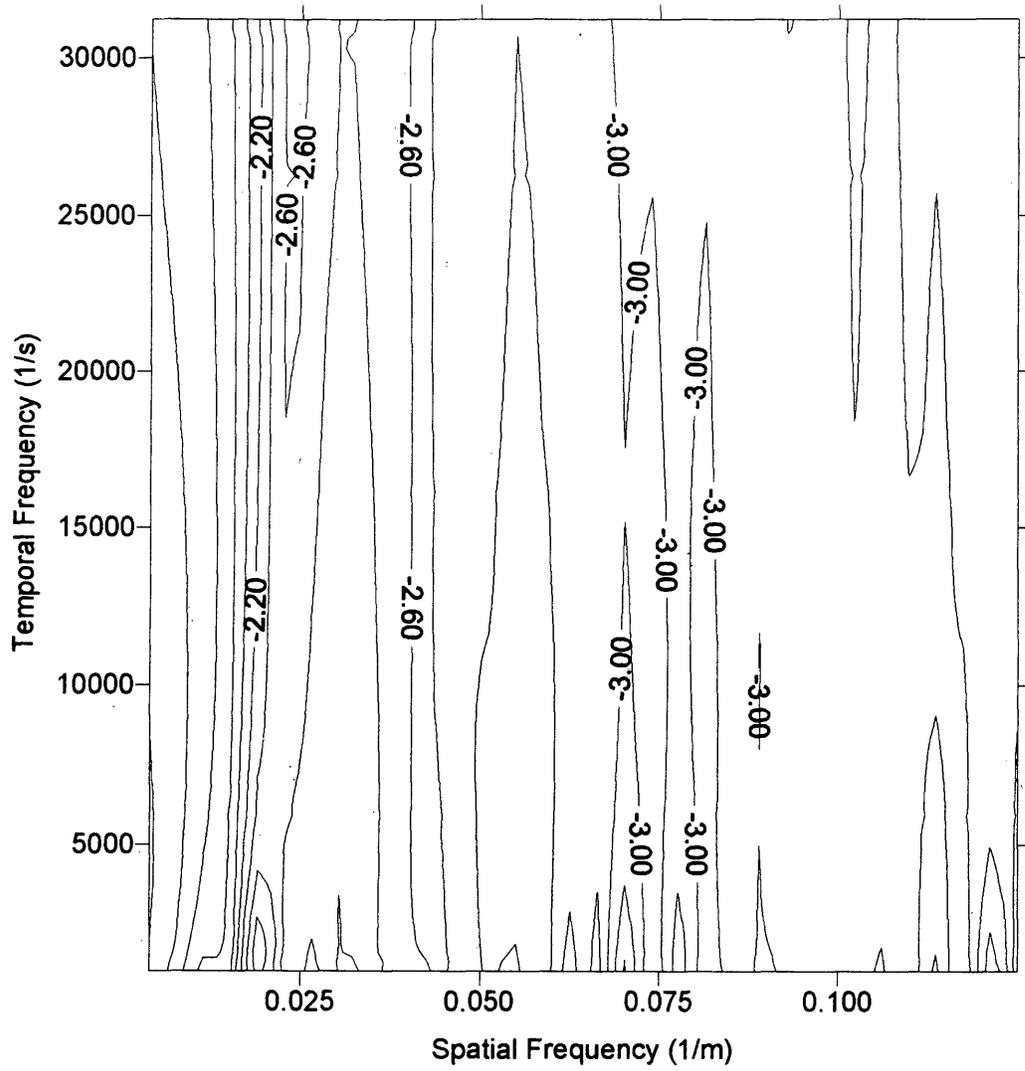


Figure C.5 : Spatial and Temporal Transform for 60 degree dike model @ 8 m depth

90 Degree Dike @ 8 m Depth

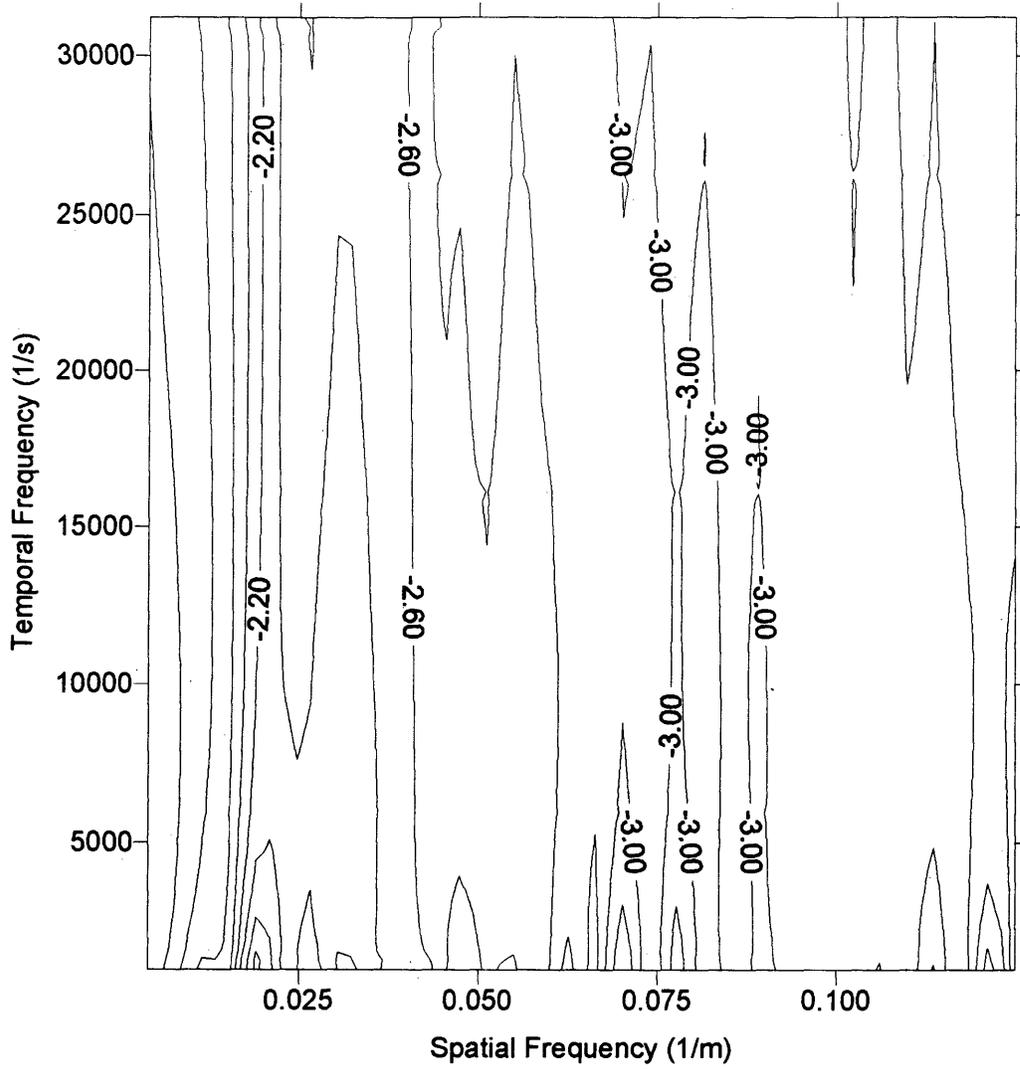


Figure C.6 : Spatial and Temporal Transform for 90 degree dike model @ 8 m depth

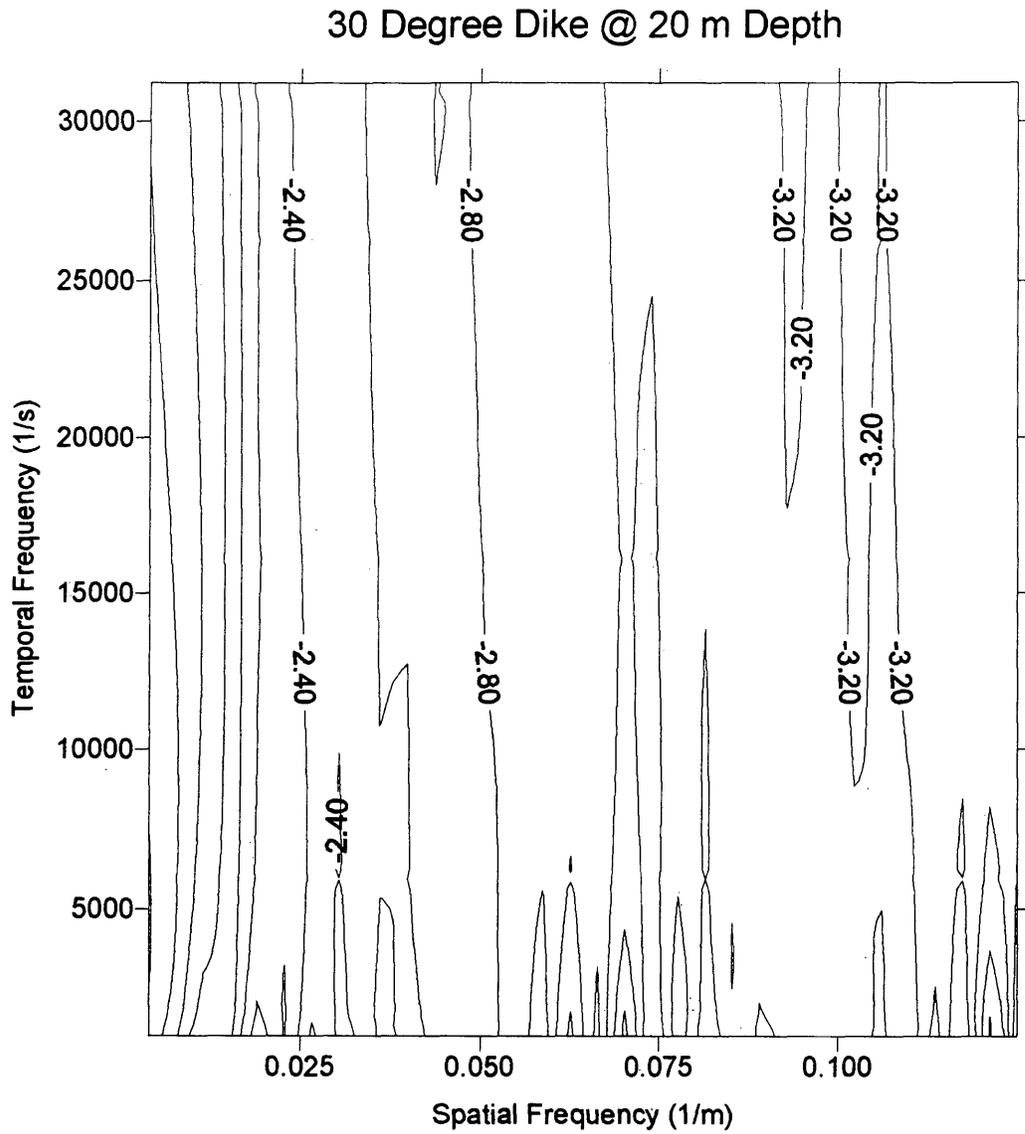


Figure C.7 : Spatial and Temporal Transform for 30 degree dike model @ 20 m depth

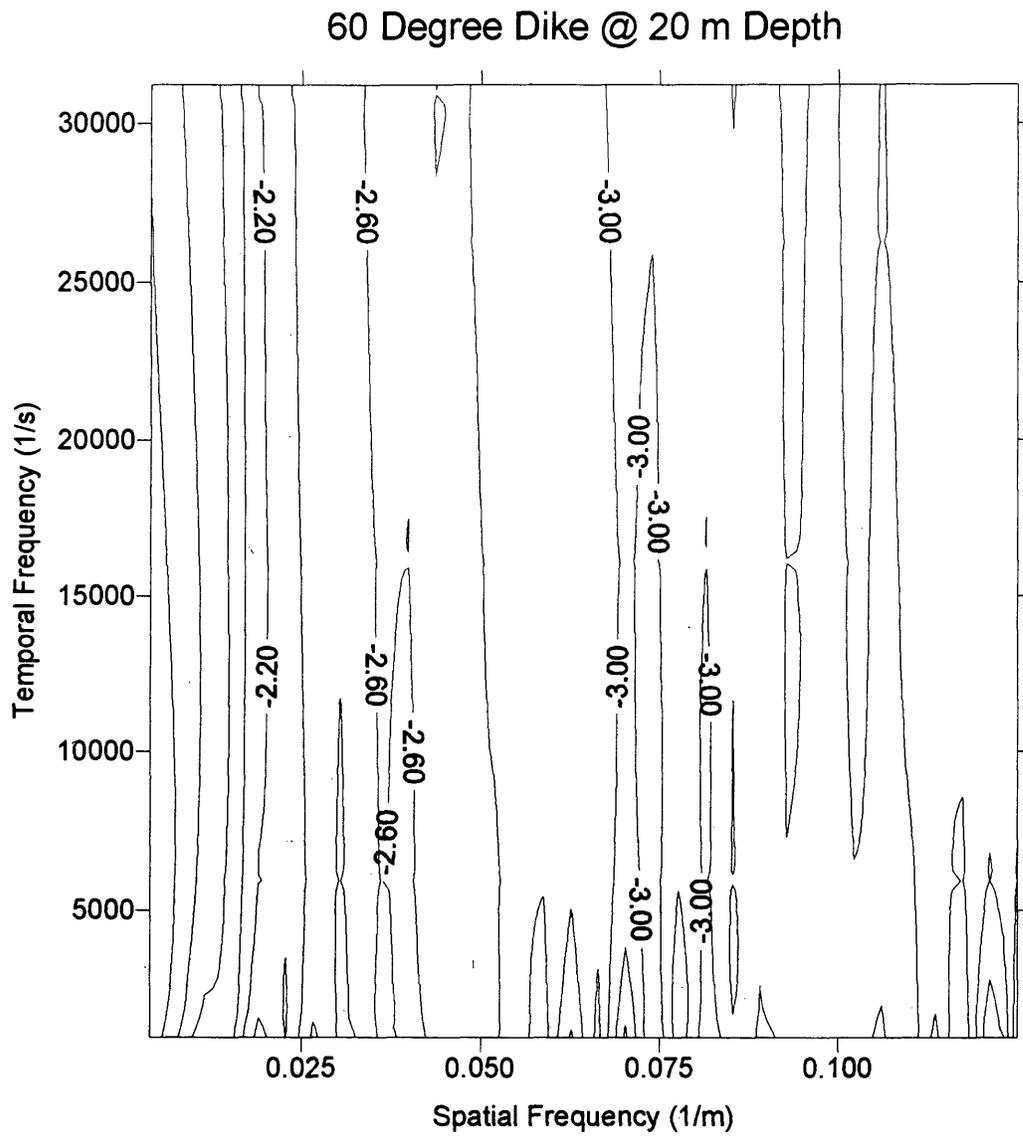


Figure C.8 : Spatial and Temporal Transform for 60 degree dike model @ 20 m depth

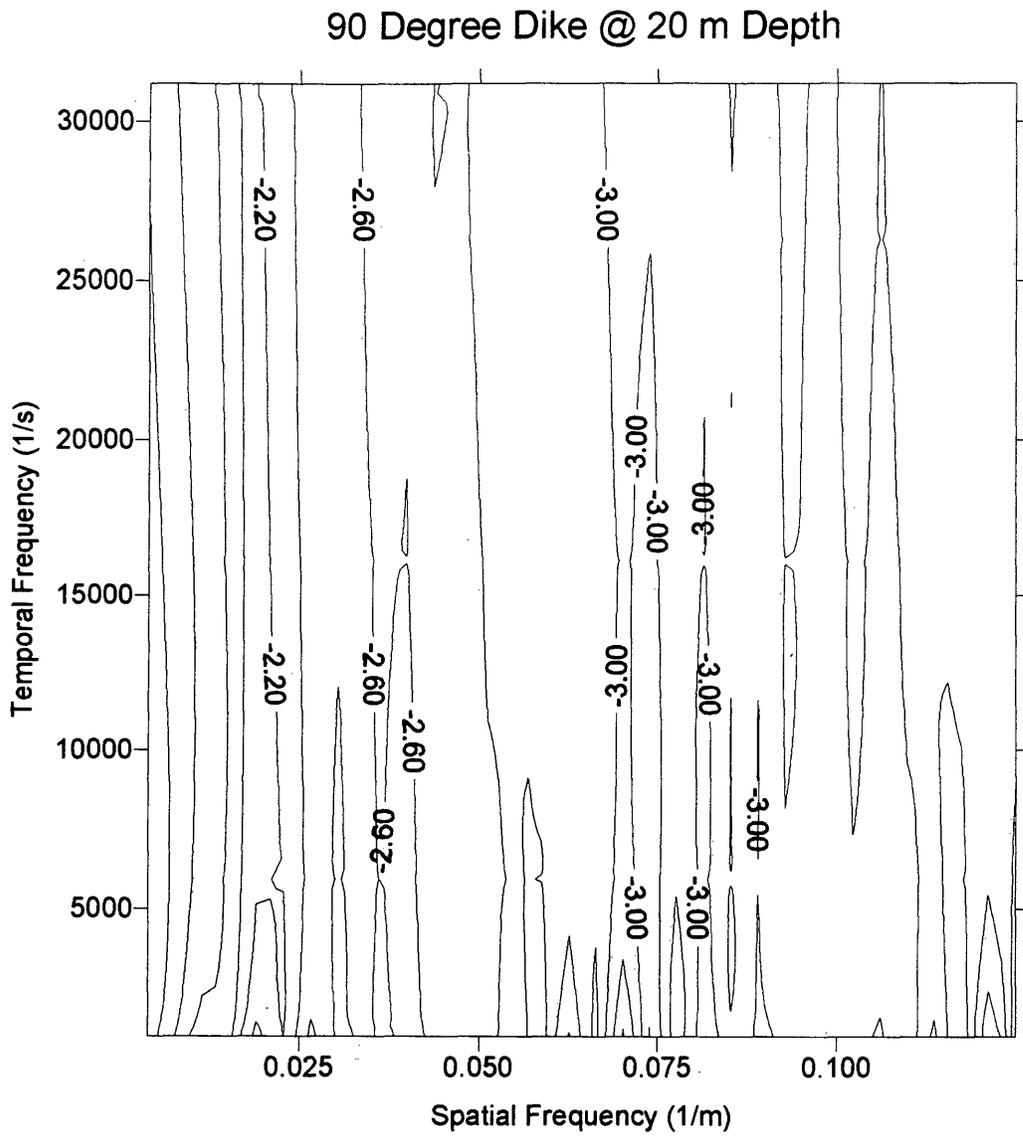


Figure C.9 : Spatial and Temporal Transform for 90 degree dike model @ 20 m depth

Appendix D

Program Listings

Included in this appendix are the programs written for Fourier transformation, neural net normalization and the file used by Aspirin to create the neural network.

```

*****
*           This program calculates the temporal
*           Fourier transform on twenty points of data
*****

      PROGRAM BTTT

      real time(20)
*****
*   time(20) contains original 20 time points for one
*   station
*****
      real hmat(20,128)
*****
*   hmat holds readings for array of stations
*****
      real itt(20)
      real frq(100)
      real xint,fttr,ftti,delkx,pi2
      real ampl,phase,kx
      complex hf(128)
*****
*   hf holds time transform for one station
*****

      integer nft,nf,tp,tpcnt,stcnt,nstn
      integer frcnt,frtot,kxcnt
      character*30 dfile,ofile
      data tp,nf,nft/20,1,128/
      data frtot/100/
      data pi2,xint/6.283185,4.0/

      delkx=pi2/(float(nft)*xint)

      write(6,*)'enter data file name?'
      read(5,*)dfile
      write(6,*)'enter output file name?'
      read(5,*)ofile

      open(unit=1,file=dfile,status='old')
      open(unit=2,file=ofile,status='new')

      call prefreq(frq,frtot)

      stcnt=1

```

```

10    read(1,*,end=20)
      read(1,*)
      do 15 tpcnt=1,tp
*****
*    read in 20 time values and input
*    for each station in array
*****
      read(1,17)time(tpcnt),hmat(tpcnt,stcnt)
17    format(2x,2(1p11.4,2x))
15    continue
      stcnt=stcnt+1
      goto 10

20    nstn=stcnt-1

      do 50 frcnt=1,frtot
      do 60 stcnt=1,nstn
      do 70 tpcnt=1,tp
          itt(tpcnt)=hmat(tpcnt,stcnt)
70    continue
*****
*    polft actually performs the transform
*****

      call polft(tp,time,itt,nf,frq(frcnt),fttr,ftti)
      hf(stcnt)=cplx(fttr,ftti)
60    continue
      do 90 kxcnt=1,nstn
          kx=float(kxcnt-1)*xint
          ampl=cabs(hf(kxcnt))
          phase=atan(aimag(hf(kxcnt))/real(hf(kxcnt)))
          write(2,95) frq(frcnt),kx,real(hf(kxcnt))
          & ,aimag(hf(kxcnt)),
          &          ampl,phase
95    format(6(e13.5,1x))
90    continue
50    continue

      close(unit=1)
      close(unit=2)

      end

```

```
*****
*   This subroutine calculates the frequencies
*   for which the transform is calculated
*   the frequencies used were linear, while
*   The commented line choses an exponential range
*****
      subroutine prefreq(frq,frtot)

      real frq(100)
      real frstep,frinit
      integer frtot,frcnt
      data frinit,frstep/10.0,30.0/

      do 10 frcnt=1,frtot
c10  frq(frcnt)=(frinit)**(float(frcnt-1)/frstep+2.5)
10   frq(frcnt)=1000+318*float(frcnt-1)
      return
      end
```

```
*****
```

```
* This subroutine actually performs the
* Fourier transform on a one-dimensional data
* input set
```

```
*****
```

```
subroutine polft(np,t,x,nf,frq,fr,fi)
```

```
implicit none.
```

```
real t(128),x(128)
```

```
real frq,fr,fi
```

```
double precision pi2,m,b,mdiff,bdiff,w,t2,t1,x1,x2
```

```
double precision cs1,sn1,cs2,sn2
```

```
integer count,np,nf
```

```
data pi2/6.238185d0/
```

```
fr=0.0
```

```
fi=0.0
```

```
w=pi2*dble(frq)
```

```
do 10 count=1,np-1
```

```
  t2=dble(t(count+1))
```

```
  t1=dble(t(count))
```

```
  x2=dble(x(count+1))
```

```
  x1=dble(x(count))
```

```
  cs1=dcos(w*t1)
```

```
  sn1=dsin(w*t1)
```

```
  cs2=dcos(w*t2)
```

```
  sn2=dsin(w*t2)
```

```
  m=(x2-x1)/(t2-t1)
```

```
  b=(t2*x1-t1*x2)/(t2-t1)
```

```
  mdiff=m*((cs2/w**2+t2*sn2/w)-(cs1/w**2+
```

```
& t1*sn1/w))
```

```
  bdiff=b*(sn2/w-sn1/w)
```

```
  fr=fr+sngl(mdiff+bdiff)
```

```
  mdiff=m*((sn2/w**2-t2*cs2/w)-(sn1/w**2-
```

```
& t1*cs1/w))
```

```
  bdiff=b*(-cs2/w+cs1/w)
```

```
  fi=fi+sngl(mdiff+bdiff)
```

```
10 continue
```

```
return
```

```
end
```

```
*****
*   This Program performs the Fourier Transform
*   in both space and time
*****
```

PROGRAM B2DFK

```
*****
*   same variables as bttt
*****
  real time(20)
  real hmat(20,128)
  real itt(20)
  real frq(100)
  real xint,fttr,ftti,delkx,pi2
  real ampl,phase,kx
  complex hf(128)
  integer nft,nf,tp,tpcnt,stcnt,nstn
  integer frcnt,frtot,kxcnt
  character*30 dfile,ofile
  data tp,nf,nft/20,1,32/
  data frtot/96/
  data pi2,xint/6.283185,4.000/

  delkx=1.0/(float(nft)*xint*2.0)

  write(6,*)'enter data file name?'
  read(5,*)dfile
  write(6,*)'enter output file name?'
  read(5,*)ofile

  open(unit=1,file=dfile,status='old')
  open(unit=2,file=ofile,status='new')

  call prefreq(frq,frtot)

  stcnt=1

10  read(1,*,end=20)
    read(1,*)
    do 15 tpcnt=1,tp
      read(1,17)time(tpcnt),hmat(tpcnt,stcnt)
17    format(2x,2(1pE11.4,2x))
15  continue
    stcnt=stcnt+1
```

```

        goto 10

20     nstn=stcnt-1

        do 50 frcnt=1,frtot
        do 60 stcnt=1,nstn
        do 70 tpcnt=1,tp
            itt(tpcnt)=hmat(tpcnt,stcnt)
70     continue
*****
*     perform time transform for one frequency
*     for each station in array
*****
        call polft(tp,time,itt,nf,frq(frcnt),fttr,ftti)
        hf(stcnt)=cplx(fttr,ftti)
60     continue

*****
*     bcon2 performs the actual
*     2d Fourier transform, hf holds time
*     transformed data for each station and
*     space transformed data is returned in
*     it as well
*****
        call bcon2(nstn,nft,xint,hf)
        do 90 kxcnt=1,nft
            kx=float(kxcnt)*delkx
            ampl=cabs(hf(kxcnt))
            phase=atan(aimag(hf(kxcnt))/real(hf(kxcnt)))
            write(2,95)frq(frcnt),kx,real(hf(kxcnt)),
&    aimag(hf(kxcnt)),
&    ampl,phase
95     format(6(e13.5,1x))
90     continue
50     continue

        close(unit=1)
        close(unit=2)

        end

        subroutine prefreq(frq,frtot)
        real frq(100)
        real frstep,frinit

```

```
integer frtot, frcnt
data frinit, frstep/10.0, 31.0/
do 10 frcnt=1, frtot
10   frq(frcnt)=1000+318*float(frcnt-1)
return
end
```

```

*****
*   This subroutine actually calculates
*   the 2d Fourier transform by
*   making two calls to bpol and using
*   the proper multiplicative factors
*   and adding the proper terms
*****

      SUBROUTINE BCON2 (NX,NFT,XINT,HF)
      implicit none
      external polft
      COMPLEX HF(128)
      real x(128),hr(128),hi(128),kx,fr,fi,frtemp,fitemp
      real pi,pi2,dx,xint
      integer count,nf,i,nxpl,nx,nft
C
      PI=3.141592
      PI2=2.0*PI
      NXP1=NX+1
*****
*   split time transform into real
*   and imaginary parts and put
*   each in real array for bpol
*****
      do 60 i=1,nx
          hr(i)=real(hf(i))
          hi(i)=imag(hf(i))
          x(i)=float(i-1)*xint
60      continue
*****
*   define frequencies according
*   to Nyquist limit
*****
      dx=1.0/(float(nft)*xint*2.0)
      do 50 count=1,nft
          kx=float(count)*dx
          call polft(nx,x,hr,nf,kx,fr,fi)
          frtemp=fr
          fitemp=fi
          call polft(nx,x,hi,nf,kx,fr,fi)
          hf(count)=cmlpx(frtemp+fi,fr-fitemp)
50      continue

      RETURN
      END

```

```

*****
*   This program takes the amplitude data
*   from Fourier transform program and
*   scales it from 0 to 1 and clips the
*   data
*****

      implicit none

      real frq(32,100),kx(32,100),val(32,100),min,max
      real sum,ave,ang,dep,x1,y1,x2,y2,m,b
      integer loop,fnum,count,scnt
      character*30 ofile,nfile
*****
*   define clip region in frequencies
*****
      data x1,y1,x2,y2/.125,28000,.080,1000/

      m=(y2-y1)/(x2-x1)
      b=y2-m*x2

      write(6,*)'enter old file?'
      read(5,*)ofile
      write(6,*)'enter new file?'
      read(5,*)nfile

      open(unit=1,file=ofile,status='old')
      open(unit=2,file=nfile,status='new')
c     open(unit=3,file='avtest',status='new')

      fnum=1
      sum=0.0
      ave=0.0
      scnt=0
      max=-9000.0
      min=9000.0
5     do 10 loop=1,32
      read(1,15,end=20)kx(loop,fnum),frq(loop,fnum)
      & ,val(loop,fnum)
15    format(3(e13.5,1x))
*****
*   if value is in clip region then add to total sum
*****
      if (frq(loop,fnum).lt.m*kx(loop,fnum)+b) then
          sum=sum+val(loop,fnum)

```

```

c          write(6,*)'sum=',sum
          scnt=scnt+1
          else
*****
*   see if value is a minimum or maximum
*****
          if (val(loop,fnum).lt.min) min=val(loop,fnum)
          if (val(loop,fnum).gt.max) max=val(loop,fnum)
          endif
10      continue
          fnum=fnum+1
          goto 5
*****
*   average sum for clipping
*****
20      ave=sum/float(scnt)
          if (ave.lt.min) min=ave
          min=min-max
          do 45 count=1,fnum-1
            do 55 loop=1,32
*****
*   if in clipped region replace with average value
*****
          if (frq(loop,count).lt.m*kx(loop,count)+b) then
            val(loop,count)=ave
c          write(6,*)ave
          endif
*****
*   replace value with linear interpolated value from 0 to 1
*****
          val(loop,count)=(val(loop,count)-max)/(min)
c          write(3,15)kx(loop,count),frq(loop,count)
c          &      ,val(loop,count)

55      continue
45      continue

          do 65 count=1,fnum-1
            write(2,25)(val(loop,count),loop=1,32)
25      format(1x,32(f6.3,1x))
65      continue
          write(6,*)'enter angle?'
          read(5,*)ang
          write(6,*)'enter depth?'
          read(5,*)dep

```

```
          write(2,94) ang, dep
94      format(1x,f6.4,2x,f6.4)

          close(unit=1)
          close(unit=2)
c      close(unit=3)

          end
```

