INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

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

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600



Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/371

UNIVERSITY OF OKLAHOMA

GRADUATE SCHOOL

SEISMIC INVERSION BY ARTIFICIAL NEURAL NETWORKS

A Dissertation

SUBMITTED TO THE GRADUATE FACULTY

In partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

By

QIANG SUN
Norman, Oklahoma
2001

UMI Number: 3028799



UMI Microform 3028799

Copyright 2002 by Bell & Howell Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

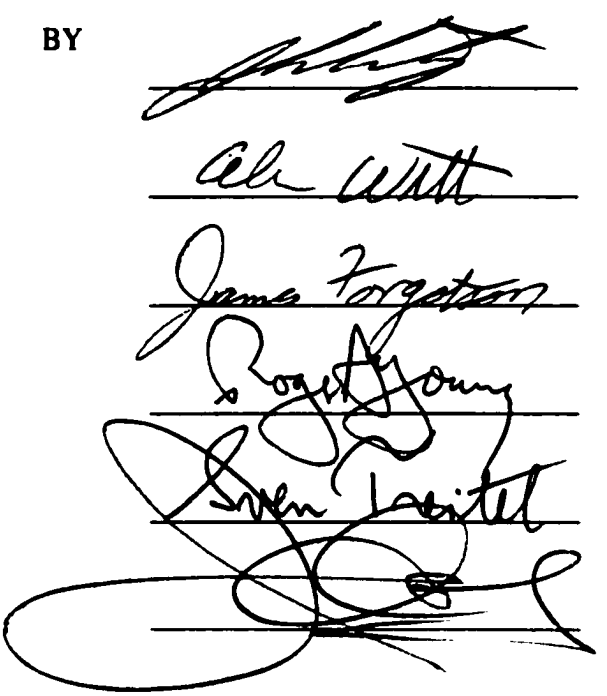
© Copyright by Qiang Sun 2001

All right reserved

SEISMIC INVERSION BY ARTIFICIAL NEURAL NETWORKS

A Dissertation APPROVED FOR THE
SCHOOL OF GEOLOGY AND GEOPHYSICS

BY



ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr. John P. Castagna, for all his guidance, advice, support and patience throughout my Ph.D. program. I wish to thank all of my committee members, Drs. John Cheung, James Forgotson, Sven Treitel, Alan Witten, and Roger Young for their advice, suggestion on my research, and their reviewing the manuscript. I would like to thank Dr. Roger M. Slatt, the director of school of Geology and Geophysics, for his assistance.

I would like to thank Drs. Zhengping Liu, Bill Lamb, and Carrie Decker of Institute of Exploration and Development Geoscience, OU for their advice on my study. I would like to thank Dr. Claren Kidd of Youngblood library, University of Oklahoma for her assistance and encouragement. I would like to thank Jan Dodson and Mohamed Eissa of School of Geology and Geophysics, OU for their assistance.

I would like to thank, Mr. Horacio D. Marín, Dario Cersósimo, Sebastián Olmos, Andrés Cerutti, and Sergio Ríos from Tecpetrol, Argentina for their assistances and providing some data in my research.

My thanks to all the staff members, Donna Mullins, Stephen Sandifer, Brian Silver, Terry Brad, Carol Drayton, Nancy Leonard, Therese Stone, Kathy Sokolic, Ucan Sezai for their help and cooperation. I would like to thank my friends, Shenjie Sun, Patrice Mahob, He Chen, Frederic Gallice and all others who helped me during my stay here.

Last by not the least, I thank my wife, Honglin Yuan, my father Zhenchun Sun, and my mother Xiaolian Xu for their support and encouragement.

TABLE OF CONTENTS

ACKNOWLEDGEMENT iv

TABLE OF CONTENTS vi

LIST OF TABLES ix

LIST OF FIGURES x

ABSTRACT xiv

CHAPTER 1. INTRODUCTION 1

CHAPTER 2. THEORY OF BP NEURAL NETWORKS 5

2.1 The structure of BP networks 5

2.2 The feedforward model of the neural network 7

2.3 The training of the neural networks 10

2.4 Over training 14

CHAPTER 3. METHODOLOGY OF SEISMIC INVERSION BY ANN 16

3.1 Frequency components 16

3.2 Training scheme 16

3.3 Resolution of the inversion 20

3.4 Noise sensitivity 24

3.5 Conclusion 28

CHAPTER 4. 2D SEISMIC INVERSION CASE STUDY, STRATTON FIELD 29

4.1 Introduction 29

4.2 Training 32

4.3 Results 36

CHAPTER 5. 3D SEISMIC INVERSION 40

5.1 Introduction 40

5.2 Training 41

5.3 The result of the inversion 44

CHAPTER 6. METHODOLOGY OF AVO ANOMALY DETECTION 49

6.1 Introduction 49

6.2 AVO inversion for the synthetic model 51

6.3 Synthetic model with constant NMO error 57

6.4 Synthetic model with variable NMO error 61

6.5 NMO error suppression by the phase unwrapping 64

CHAPTER 7. AVO INVERSION FOR 2D SEISMIC DATA 68

7.1 Introduction 68

7.2 Case study 1 69

7.3 Case study 2 76

CHAPTER 8. CONCLUSIONS AND DISCUSSION 86

8.1 Conclusions 86

8.2 Discussion and Future work 90

REFERENCE 99

Appendix A. 103

Consortium Report, 1998, Seismic inversion by artificial neural networks

Appendix B. 129

Resistivity and lithology fraction inversion for the YPF data set

LIST OF TABLES

Table 6-1. Model parameters (Castagna et al., 1998) 51

LIST OF FIGURES

Figure 2-1. The structure of the feedforward neural network 5

Figure 2-2. Activation functions of a neuron: (a) bipolar continuous and (b) unipolar continuous 10

Figure 2-3. BP neural network training rule 12

Figure 2-4. Training error and validation error with training times 15

Figure 3-1. The Learning pair 17

Figure 3-2. The configuration of training data set 19

Figure 3-3. The synthetic model made by GXII, the velocity in the wedge is 1000 m/s, and the background velocity is 4000 m/s. 20

Figure 3-4. Synthetic seismic trace, the dominant frequency is 20 Hz. 21

Figure 3-5. The result of conventional inversion 22

Figure 3-6. The result of inversion using neural network 24

Figure 3-7. The result of blocky inversion with noisy seismic traces 25

Figure 3-8. The result of inversion using neural networks, 10% noise was added before the inversion. Because we just use one random seed, the noise is not really random. 26

Figure 3-9. The result of inversion using neural networks, six traces with same noise level (10%) but different random noise system (different random seeds) were used in the training. 27

Figure 4-1. The seismic section and well log data used in the inversion of 31

resistivity. Among the three wells, the middle one was used in the predicting, and the other two wells were used in the training.

Figure 4-2. Flow Chart of Neural Network 33

Figure 4-3. Flow Chart of Emerge 34

Figure 4-4. Emerge Validation Analysis 35

Figure 4-5. Well 11 (training well), result of neural network inversion 37

Figure 4-6. Well 11 (training well), result of Emerge inversion 37

Figure 4-7. Well 12 (training well), result of neural network inversion 38

Figure 4-8. Well 12 (training well), result of Emerge inversion 38

Figure 4-9. Well 18 (testing well), result of neural network inversion 39

Figure 4-10. Well 18 (testing well), result of neural network inversion 39

Figure 5-1. Location of Bermejo Field 40

Figure 5-2. BS-04 (training well), observed porosity vs predicted porosity 42

Figure 5-3. BN-17 (Predicting well), observed porosity vs predicted porosity 43

Figure 5-4. Inversion of porosity for the arbitrary line. 43

Figure 5-5. Interpolated porosity cube 46

Figure 5-6. Porosity cube calculated by neural network 47

Figure 5-7. Comparison of results of inversion for impedance and porosity cube. High acoustic impedance is shown in white, yellow and red, low acoustic impedance in blue and green. Right shows porosity cube for same interval; warm colors indicate high porosity and cool colors indicate low porosity. Gamma ray log is shown to the left of the borehole; porosity log 48

is shown to the right in each figure.

Figure 6-1. The flow chart of AVO inversion using neural network 49

Figure 6-2. Synthetic Model 51

Figure 6-3. Training window AVO 52

Figure 6-4. Brine window AVO 53

Figure 6-5. Gas window AVO 55

Figure 6-6. AVO anomaly 55

Figure 6-7. The AVO anomaly for the whole synthetic section 56

Figure 6-8. Training Window AVO, 8ms constant NMO error 59

Figure 6-9. Brine Window AVO, 8ms constant NMO error 59

Figure 6-10. Gas Window AVO, 8ms constant NMO error 60

Figure 6-11. AVO Anomaly, 8ms constant NMO error 60

Figure 6-12. Training Window AVO, 8ms variable NMO error 62

Figure 6-13. Brine Window AVO, 8ms variable NMO error 62

Figure 6-14. Gas Window AVO, 8ms variable NMO error 63

Figure 6-15. AVO anomaly, 8 ms variable NMO error, with correlation 63

Figure 6-16. Flow chart of NMO error suppression by unwrapping the phase spectrum 66

Figure 6-17. Brine window AVO, before the phase unwrapping 67

Figure 6-18. Brine window AVO, after the phase unwrapping 67

Figure 7-1. Near partially stacked traces for the 2D seismic data set 71

Figure 7-2. Far partially stacked traces for the 2D seismic data set 72

Figure 7-3. The AVO anomaly of neural network inversion for the 2D seismic data set 73

Figure 7-4. Training window AVO 74

Figure 7-5. Brine window AVO 74

Figure 7-6. Gas window AVO 75

Figure 7-7. AVO anomaly 75

Figure 7-8. Partially stacked near traces 79

Figure 7-9. Partially stacked far traces 80

Figure 7-10. AVO anomaly without phase unwrapping 81

Figure 7-11. AVO anomaly with phase unwrapping 82

Figure 7-12. AVO inversion in the training window 83

Figure 7-13. AVO inversion in the brine window 83

Figure 7-14. AVO inversion in the gas window 84

Figure 7-15. AVO anomaly without phase unwrapping 84

Figure 7-16. AVO anomaly with phase unwrapping 85

Figure 8-1. Flow chart of adding convolution model 92

Figure 8-2. Flow Chart of AVO inversion in frequency domain 96

Figure 8-3. AVO anomaly, far traces were used as input. 97

Figure 8-4. AVO anomaly, CDP gathers in the gas window were used in the training. 98

ABSTRACT

Artificial neural networks (ANN) have recently attracted attention for their ability to "learn" and "estimate" the mapping relationships between the data (Liu et al., 1998). In this dissertation, I investigate the applications of the artificial neural network in geophysical discipline.

The first application is seismic inversion. Artificial neural networks were used to invert post-stack seismic data into the petrophysical attributes. This application involves two steps. In the first step, the seismic data, low frequency data, spatial and temporal constraints are used as input and petrophysical attribute data (usually come well logs) are used as desired output data to train the neural network. In the second step, the trained neural network is used to predict the petrophysical attribute by inputting the seismic data. Both synthetic data and real data were used to test the seismic inversion by ANN. The results of this inversion have higher resolution and accuracy than those of the conventional inversion.

The second application is AVO inversion. Artificial neural network were trained to learn the relationship between a near-offset partially stacked trace and a far-offset partially stacked trace for non-hydrocarbon bearing rocks. Far traces can then be predicted by this learned relationship and the difference between observed and ANN predicted traces can potentially be used as a hydrocarbon indicator. An advantage of this method over conventional cross-plotting techniques is that it can be made insensitive to incorrect normal moveout corrections.

CHAPTER 1. INTRODUCTION

Artificial Neural Networks (ANN) simulate the structure of human brain utilizing specialized hardware and/or sophisticated software. The field of the study of their application goes by many names, such as connectionism, parallel distributed processing, neuro-computing, natural intelligent systems, machine learning algorithms, and artificial neural networks. The purpose of this dissertation is to study the use of artificial neural networks in the inversion of seismic data to obtain reservoir properties for petroleum exploration and reservoir characterization.

An artificial neural network is composed of multiple layers of simple processing elements called neurons. Each neuron is linked to certain of its neighbors with varying coefficients of connectivity that represent the strengths of these connections. These coefficients are called weights. Learning is accomplished by adjusting these weights to cause the overall network to output appropriate results.

Neural networks can be classified by various properties. They can be classified by training method, such as supervised and unsupervised; by the learning rule, such as Hebbian, perceptron, delta, winner-take-all, etc.; by the number of layers, such as single layer and multi-layer; and by data flow direction, such as feedforward and feedback networks. According to these classifications, the neural network selected for this research is a supervised feedforward multi-layer network with error back-

propagation training, using a delta-learning rule. This network is the most common choice for seismic inversion.

In the past decade, neural networks have been widely used in many disciplines for (1) classification and recognition, such as pattern classification, finger print recognition, and voice recognition, and (2) prediction such as market prediction. ANN's are also increasingly finding application in geology and geophysics.

Calderon-Macias (1997) used Hopfield neural networks in seismic deconvolution and multiple attenuation. Boadu (1997) approximated the relationship between the Rock Properties and Seismic Attenuation by ANN. Liu (1998) used an ANN to invert the seismic data into the well log data. Boadu (1998) inverted seismic velocity to fracture density by ANN. Calderon-Macias (1998) used ANN to do automatic NMO correction. Zhang (1999) applied the ANN to do fast forward modeling simulation for resistivity well logs. Davies (1999) cross-correlated the seismic surveys acquired with different bandwidth by neural networks. Buffenmyer (1999) applied the ANN in seismic crew noise identification in marine surveys. Chakravarthy (1999) detected layer boundaries from array induction tool responses using neural networks. Meldahl (2001) used neural networks to identify fault and gas chimneys. All these applications can be classified to the same two categories, (1) recognition such as noise, fault, and gas chimney identification, and (2) prediction such as simulation for resistivity well

logs. The application of ANN discussed in this dissertation belong to the prediction category.

Theoretically, a neural network can simulate any complex physical relationship. Based on this property, in this research, ANN's are applied in seismic inversion for petrophysical attributes and detection of AVO anomalies.

Traditional inversion can only directly invert seismic data to impedance or perhaps velocities and density with sufficient constraints. ANN inversion can potentially invert seismic data to any petrophysical attributes, if those attributes have some relationship with the seismic data. In previous studies (reference consortium report; included as appendix A) I investigated the use of ANN's to invert the post-stack seismic data into petrophysical parameters such as P-wave velocity, impedance, gamma ray response, resistivity, and porosity. I determined that the method worked better than linear regression for both 2D and 3D data. Other studies indicate that post-stack neural network inversion can be used to recognize lithology (Hampson et al., 2001, Appendix B shows the application of neural networks to elastic impedance inversion results to create resistivity and lithology sections). In this dissertation, I show two case studies where 2D seismic data were inverted to resistivity and 3D seismic data were inverted to produce a porosity cube for the reservoir simulation.

Another objective of this thesis is to investigate the use of ANN's to detect AVO anomalies. An ANN was trained to learn the relationship between near-offset and the far-offset traces in a zone that doesn't contain hydrocarbons. Far traces are predicted using that relationship; the differences between the predicted trace and the observed trace can thus be considered a kind of AVO anomaly which should be large when the neural network fails to predict properly, as would occur in the case of hydrocarbons when the neural network is trained in a non-hydrocarbon bearing interval. The application of this method to two 2-D datasets confirmed the validity of this method.

CHAPTER 2. THEORY OF BP NEURAL NETWORKS

The neural network I used in my research is a feedforward neural network with error back-propagation training algorithm (hereafter referred to as a BP network). With one hidden layer, the BP ANN is a universal function approximator (Hornik, 1989).

2.1 The structure of BP networks

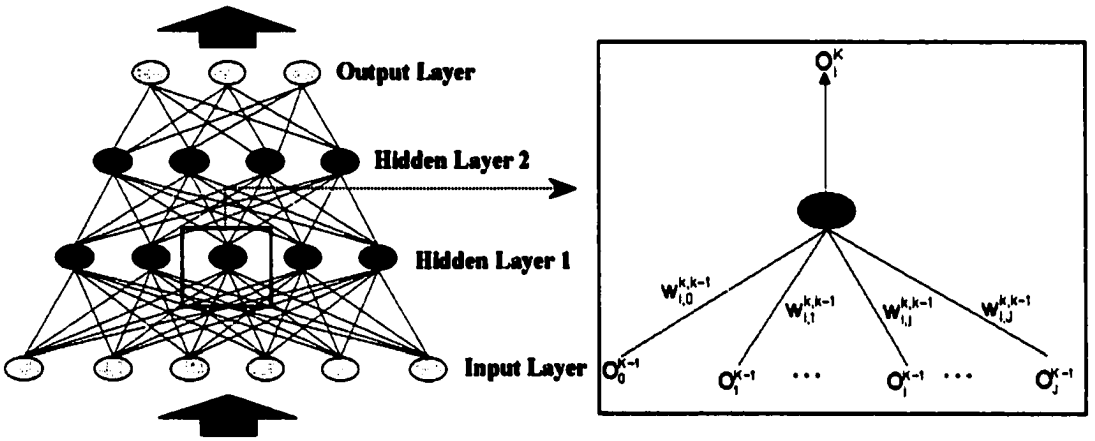


Figure 2-1. The structure of the feedforward neural network

Figure 2-1 shows the structure of the feedforward neural network. The network is composed of multiple layers of simple processing elements called neurons. Three kinds of layers, the input layer, the output layer and the hidden layer comprise the network. The input layer is the entrance for the data with every neuron representing a unit of the input data. Thus, the input data is simply a series of numbers. Theoretically, there is no limit on the length of the series. However, because this length is the most important factor to determine the size of the neural network and therefore the training

and execution time, it should be kept to a reasonable length. The larger the size of the network, the slower the speed of processing and training. The output layer is the exit for the calculated values with every neuron in this layer representing a calculated result. Because a network with pyramidal structure performs better than a network with a rectangular structure, the length of this series should be smaller than that of the input layer. The hidden layers are between the input layer and output layer. The closer the hidden layer to the output layer, the smaller the size of the hidden layer. Neurons in the same layer are not connected. The neurons in adjacent layers are connected with each other by weights.

Recently, methods to optimize the structure of the neural networks dynamically have been developed (Baan and Jutten, 2000). These methods involve changing the complexity of the structure, either from simple to complex, or the reverse. When the starting structure is very simple, it is made more and more complex step by step. For each step, there will be some neurons or a hidden layer added into the networks. If one step does not modify the performance of the neural network by a given level, the structure will stop changing. When the starting structure is unnecessarily complex, it is made more and more simple step by step. For each step, there will be some neurons or a layer taken away from the network. If one step does not hurt the performance of the network very much, it will go onto the next step until there is a step that reduces the performance of the network by a given amount. The basic idea of both categories is to

find the simplest and most efficient structure of the network for solving a specific problem.

The dynamical methods to optimize the structure are very CPU intensive. Therefore, in this research, the size and structure of the neural network are fixed. The size of the input layer is one wavelength; the size of the output layer is 1. Thus, all the seismic data points within a time window equal to the length of the wavelet are used to predict a single rock property at a given depth. Trial and error is used to adjust several structural schemes to find the best one to use.

2.2 The feedforward model of the neural network

When the data is input into the networks, it will go to the first hidden layer by the connections between the input layer and first hidden layer. Every connection has a coefficient of connectivity called a weight. From the right part of Figure 2-1, we can see that all the weighted output from the previous layer will be inputted into the neuron in the current layer and be summed. We call this sum the net value of this neuron. It is expressed as:

$$net_i^k = \sum_j^J w_{i,j}^{k,k-1} O_j^{k-1} - \theta_i^k \qquad (2\text{-}1)$$

where

k – the current layer

k-1 – the previous layer

i – the neuron in the current layer

j – the neuron in the previous layer

$w_{i,j}^{k,k-1}$ – the weight of the connection between the i_{th} neuron in the k_{th} layer and the j_{th} neuron in the $k\text{-}1_{th}$ layer

O_j^{k-1} – the output of the j_{th} neuron in the $k\text{-}1_{th}$ layer

net_i^k – the net value of the i_{th} neuron in the k_{th} layer

θ_i^k – the threshold value of the i_{th} neuron in the k_{th} layer

The threshold value is an option. This parameter can mute the outputs of some neurons that are less than a given level. The net value of the neuron will be inputted into a special function called the activation function to calculate the output of the neuron.

There are four basic types of activation functions. They are bipolar continuous (equation 2-2), bipolar binary (equation 2-3), unipolar continuous (equation 2-4), and unipolar binary (equation 2-5), .

$$f(net) = \frac{2}{1+\exp(-net)} - 1 \quad (2\text{-}2)$$

$$f(net) = \begin{cases} +1, & net > 0 \\ -1, & net < 0 \end{cases} \quad (2\text{-}3)$$

$$f(net) = \frac{1}{1+\exp(-net)} \qquad (2\text{-}4)$$

$$f(net) = \begin{cases} +1, & net > 0 \\ 0, & net < 0 \end{cases} \qquad (2\text{-}5)$$

The binary functions are usually used in recognition and classification. In prediction, and data mapping, continuous functions are used. The continuous activation functions are shown in Figure 2-2. The value of bipolar the continuous function varies from –1 to +1, and the value of the unipolar continuous function varies from 0 to 1. In this research, the input and output can be positive or negative, so the bipolar continuous activation function is used. For the properties of the activation function, both the input data and the desired output data should be standardized to the value between –1 and +1 before input into the neural network.

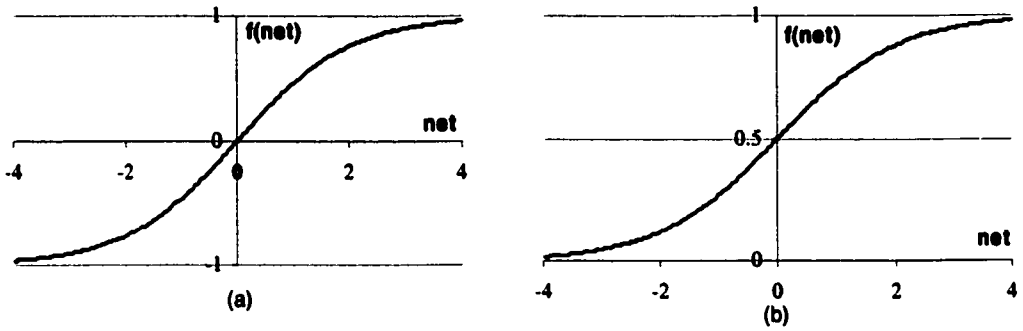


Figure 2-2. Activation functions of a neuron: (a) bipolar continuous and (b) unipolar continuous

The input data will feed forward through the hidden layer, and reach the output layer. The relationship between the input data and the output data exists in the weights of the neural network. The neural network establishes this relationship using a process called learning, or from the standpoint of the user of the neural network, training. This process is achieved by adjusting the weights in the network using some algorithm to minimize the prediction error.

2.3 The training of the neural networks

There are two basic training schemes, supervised and unsupervised training. In supervised training, there is desired output d provided by the teacher. The desired output serves to determine prediction error and is used to modify the network parameters externally. In unsupervised training, there is no such desired output. Learning must somehow be accomplished based on observations of responses to inputs for which there is marginal or no knowledge. "The technique of unsupervised

leaning is often used to perform clustering as the unsupervised classification of objects without providing information about the actual classes." (Zurada, 1992).

This dissertation addresses only supervised neural network training. Both input data and desired output data are provided to train the network.

The weights of the neural network are initialized by some random number between -0.5 and 0.5. When the input data come into the network, the network will calculate relative outputs. The differences between the calculated output (o_i) and the desired output (d_i) are called errors. The mean square error (equation 2-6)

$$E = \frac{1}{2}\sum_{i}^{n}(d_i - o_i)^2 \qquad (2\text{-}6)$$

is propagated backward along the links in the network and the weights are modified along the negative gradient direction of the error function to minimize E:

$$w_{i,j}^{l,k+1} = w_{i,j}^{l,k} - \eta \frac{\partial E}{\partial w_{i,j}^{l,k}} + \alpha\left(w_{i,j}^{l,k} - w_{i,j}^{l,k-1}\right) \qquad (2\text{-}7)$$

where the superscript k refers to the iteration step, l is the layer number, i and j are node numbers, and α and η are momentum and learning rate respectively. (Liu et al., 1998) In practice, the training is processed from output layer to input layer. The Figure 2-3 and equations (2-8)~(2-11) explain this process.

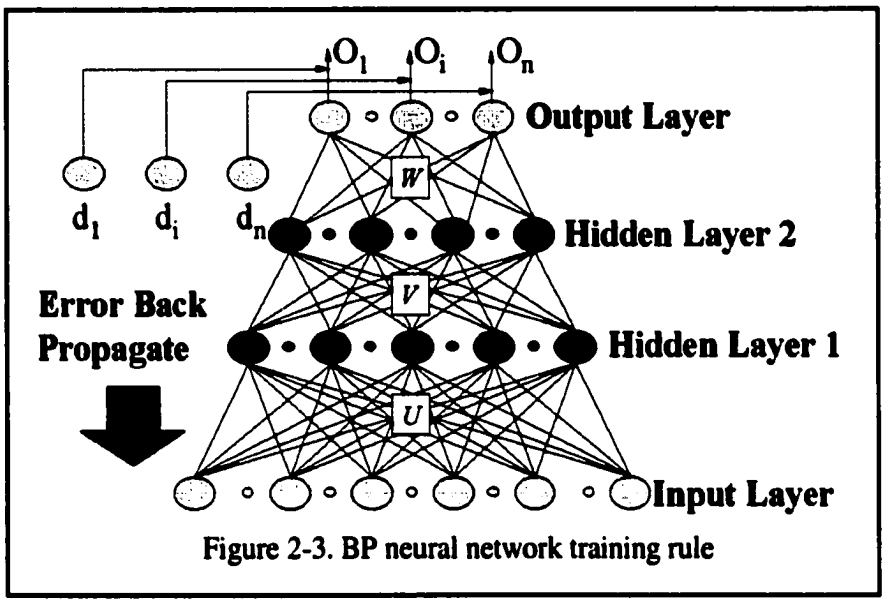


Figure 2-3. BP neural network training rule

$$\delta_{oi} = \frac{1}{2}(d_i - o_i)(1 - o_i^2) \qquad (2\text{-}8)$$

$$\delta_{yj} = \frac{1}{2}(1 - y_j^2)\sum_{i=1}^{l}\delta_{oi} w'_{ij} \qquad (2\text{-}9)$$

$$w_{ij} = w'_{ij} + \eta\delta_{oi} y_j + \alpha\Delta w'_{ij} \qquad (2\text{-}10)$$

$$v_{jk} = v'_{jk} + \eta\delta_{yj} z_k + \alpha\Delta v'_{jk} \qquad (2\text{-}11)$$

where

d_i -- The desired output for ith neuron in the output layer

o_i -- The calculated output for ith neuron in the output layer

y_j -- The calculated output for jth neuron in the hidden-layer-2

z_k -- The calculated output for kth neuron in the hidden-layer-1

δ_{oi} -- The error signal vector for ith neuron in the output layer

δ_{yj} -- The error signal vector for jth neuron in the hidden-layer-2

w_{ij} -- The weight between the ith neuron and the jth neuron

v_{jk} -- The weight between the jth neuron and the kth neuron

w'_{ij}, v'_{jk} -- The w_{ij}, v_{jk} for the previous iteration

$\Delta w'_{ij}$, $\Delta v'_{jk}$ -- The adjusted amount of w_{ij}, v_{jk} for the previous iteration

η, α -- the momentum and the learning rate

The equation of error signal vector is derived from equations (2-2) and (2-6). The error signal vectors are used to adjust the weights between the output layer and the hidden-layer-2. Then the error will propagate backward by calculating δ_{yj} to adjust the weights between the hidden-layer-2 and hidden-layer-1. The weights between hidden-layer-1 and the input layer can be adjusted in the same way. When all the weight in the network been adjusted, the network will calculate the output again with the new weights. This output will be compared with the desired output, if the error is smaller

than a given level, the training will be halted, otherwise a new iteration of adjusting weights will begin.

In this way, the neural network can learn the relationship between the input data and the desired output data.

2.4 Over training

The following analogy helps explain the concept of over training: If you try to teach a boy what is a cat, you may bring one black cat to the boy, and let him observe it. He may observe the cat very carefully and many times and remember every detail of the properties of the cat. Next time, when he meets a white cat, he may think the white cat is not a cat because it is not black. Every member in a class has its common properties and special properties. If the learner mixes up the special properties with common properties, it will not recognize the class properly. We call this phenomenon over training.

Over training can be caused by two factors. The first is deficient training examples. The second is overlong training times. For the above example, if we bring the boy many cats with different colors, he will learn that the cats may have many different colors. So the number of the sample is very important for training. We should give as many as possible representative samples to the neural network to let the network learn the common properties of the class.

Overlong training times also can cause the over training. The proper training times can be found by dividing the samples into two groups, a training and a validation group. The training group will be used in adjusting weights of the network, the validation group won't take part in training but will be used to test the network response after every training time. The error (the differences between the desired output and calculated output) of the training group will decrease with the training time, but the error of the validation group will decrease and then increase with the training time. The point at which the error of the validation group begins to increase is the time that the training should be stopped. (Figure 2-4)

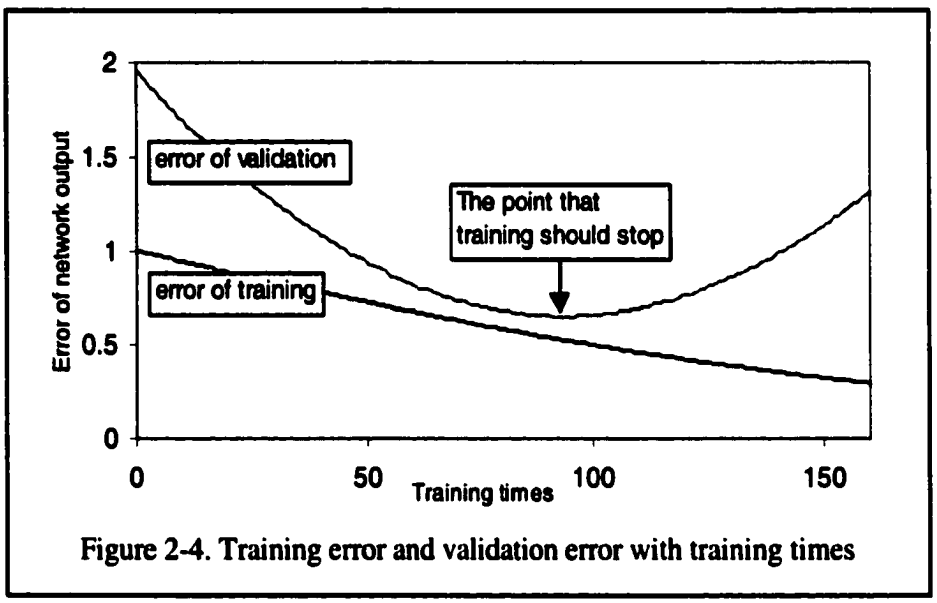


Figure 2-4. Training error and validation error with training times

CHAPTER 3. METHODOLOGY OF SEISMIC INVERSION BY ANN

3.1 Frequency components

In seismic inversion, we try to invert the post-stack seismic data into petrophysical attributes. Those attributes mostly come from the well logs. The frequency spectrum of the well log data is broader than that of the seismic data, which means that the well log has some lower and higher frequency components that the seismic data do not have. Using a special training scheme for the neural network, we can add the higher frequency components into the inversion results. However, it is impossible for the neural network to obtain the lower frequency components that seismic data does not have directly from the seismic traces alone. So, this low frequency component must be provided to the neural network.

The low frequency data is obtained by filtering interpolated well log data with a low frequency band pass filter (less than 15 Hz). This data will be used with the post-stack seismic data as the input data set for the neural networks. The low frequency data should have the same sampling rate and lateral resolution as the post-stack seismic data. If there are some geological surfaces to constrain the interpolation, the low frequency data will be more accurately interpolated.

3.2 Training scheme

Training is the most important step for neural network application. The purpose of this step is encoding the relationship between the input data (seismic data) and output data (well log) into the weights of the neural network. To fulfill this purpose, input data and desired output data should be provided to the neural network. We call this pair of data a learning pair.

The input data used are one wavelength of seismic data and low frequency data of the petrophysical parameter to be inverted. When it is necessary, temporal and spatial constraints may be added to the input data set. The output data consists of one point of petrophysical parameter at the center of each window. The learning pair is shown in Figure 3-1.

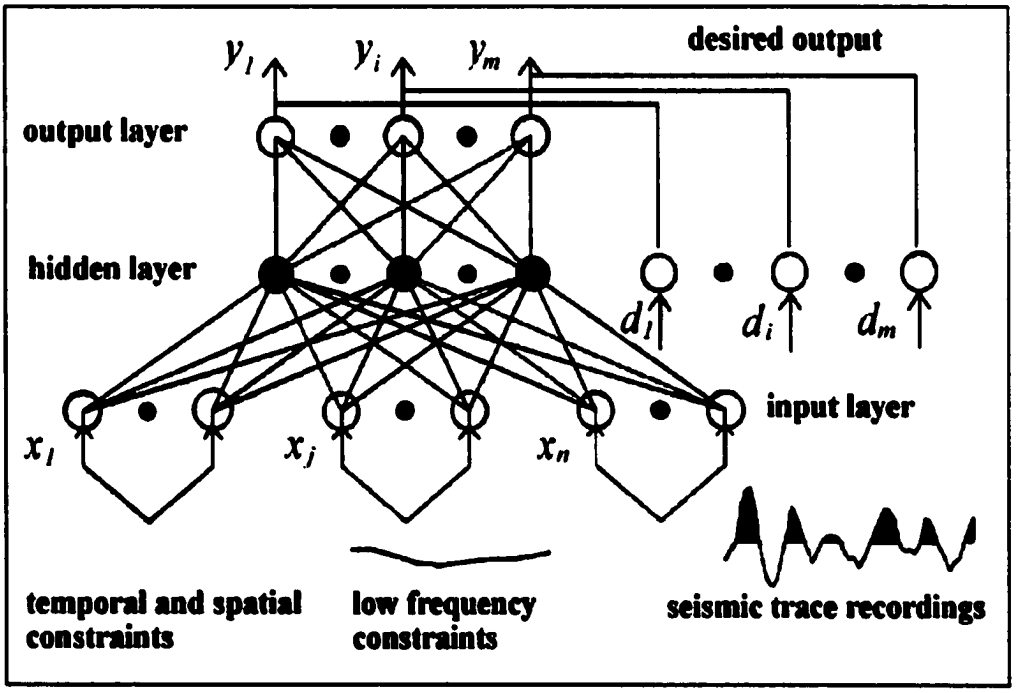


Figure 3-1. The Learning pair

The use of one point petrophysical attribute correlating to one wavelength of seismic and corresponding low frequency data is based on the assumption that any underground reflector will influence one wavelength of seismic response. This is shown in Figure 3-2. By sliding the sampling window, we can create another learning pair. The length of the sampling window is determined by the reciprocal of the main frequency of the seismic data. The petrophysical attributes mostly come from well logs. The well log should be depth to time converted, and smoothed, and re-sampled to the same sampling rate as the seismic data.

There are two advantages for creating the learning pair in this way. (1) Avoiding over training as many learning pairs are constructed. For example, if we have a 200 ms seismic trace and corresponding well log, with 2 ms sampling rate and 30 ms wavelength, we can make 85 learning pairs. (2) Modifying the resolution. With this training method, the resolution of the inversion approximates the seismic sampling rate. The result of synthetic modeling verifies this.

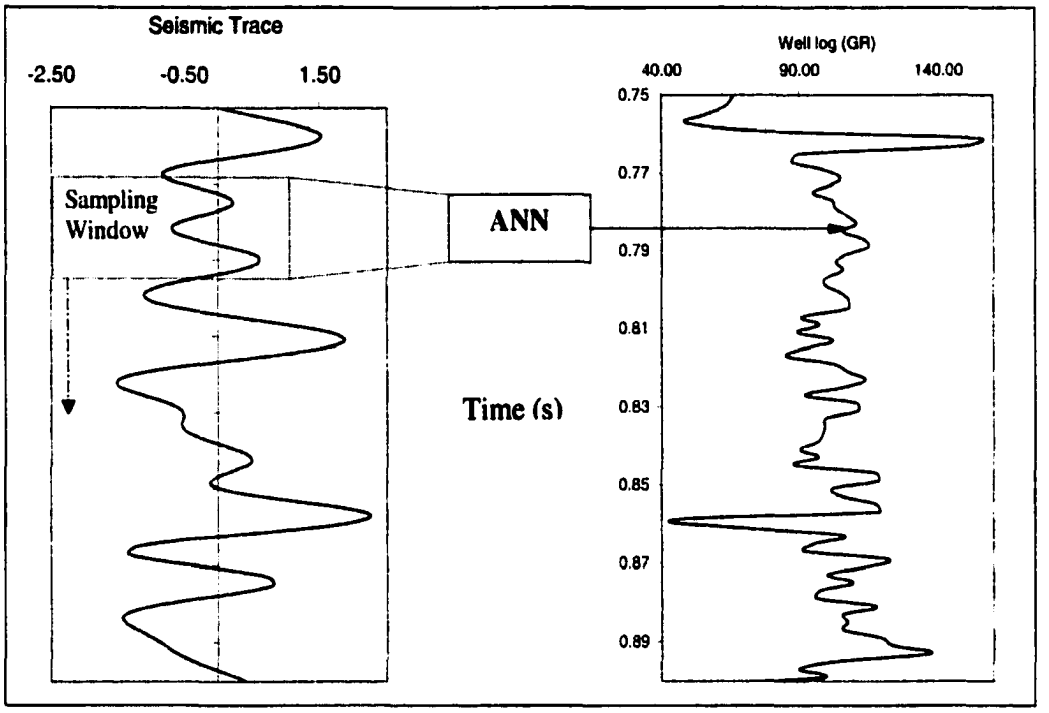


Figure 3-2. The configuration of training data set

3.3 Resolution of the inversion

The resolution of inversion using neural networks can be tested using simple synthetic models (using GXII seismic modeling software). This wedge model is shown in Figure 3-3.

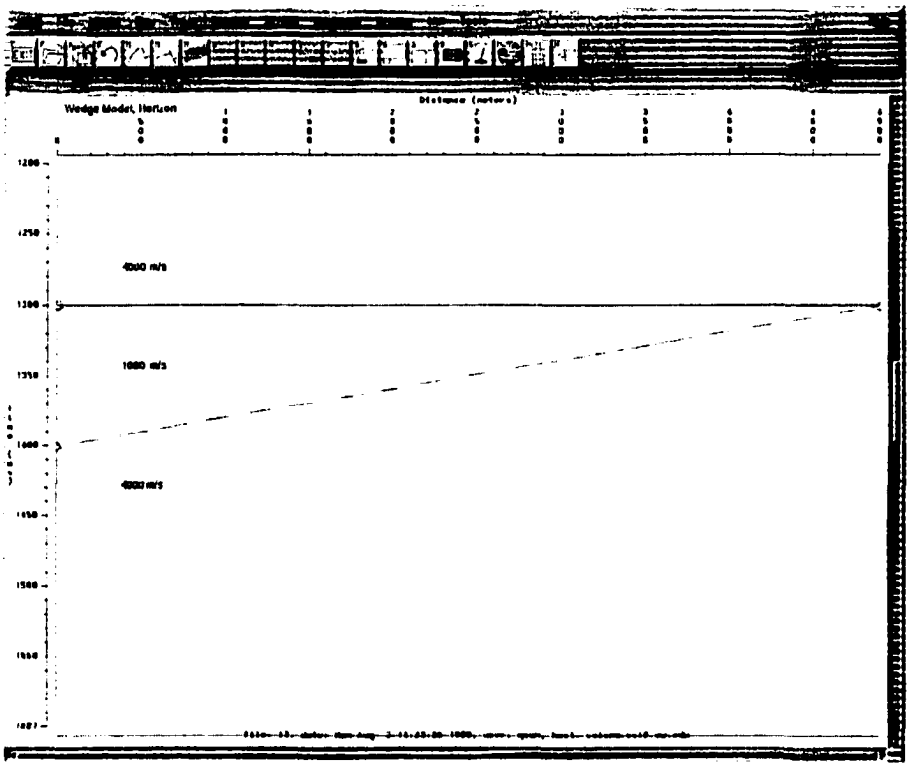


Figure 3-3. The synthetic model made by GXII, the velocity in the wedge is 1000 m/s, and the background velocity is 4000 m/s.

The velocity of the background is 4000 m/s and the velocity of the wedge is 1000 m/s. The maximum thickness of the wedge is 100 m in depth, or 100 ms in time. The wavelet is a 20 Hz Ricker wavelet. So the wavelength in the wedge is 50m. Along the horizontal

direction, there are 50 zero-offset traces. The synthetic seismic traces are shown in Figure 3-4.

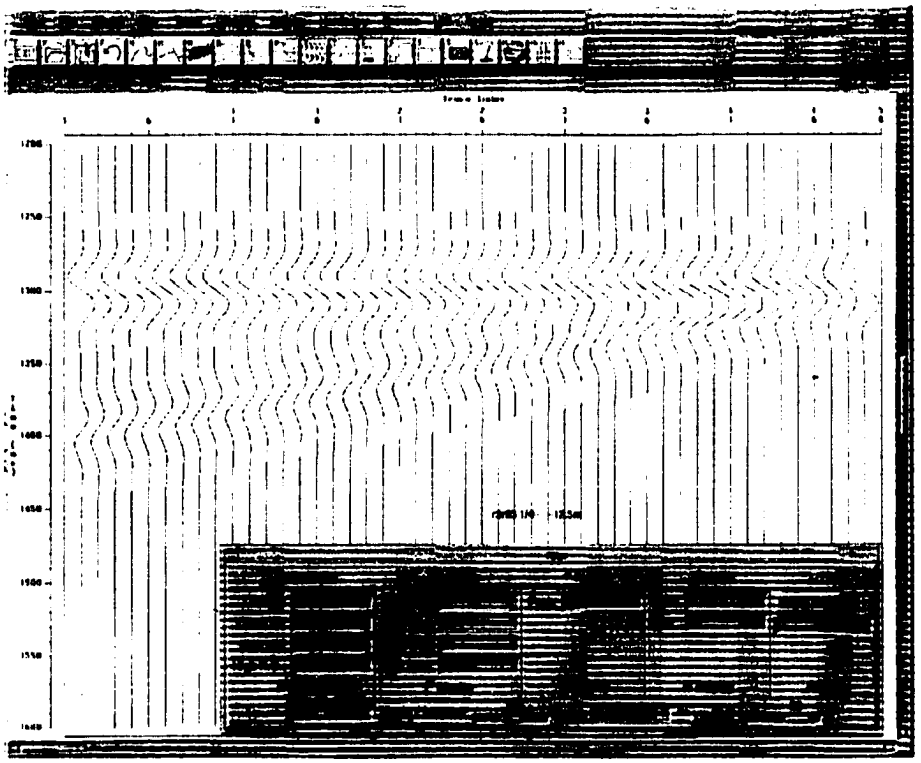


Figure 3-4. Synthetic seismic trace, the dominant frequency is 20 Hz.

Usually, the practical seismic resolution is ¼ wavelength. When a bed embedded in a medium of different impedance is about ¼ wavelength in thickness, the reflection from the top and the base of the bed interfere constructively and the amplitude increases. This is called the tuning thickness. At and below this thickness, we cannot resolve the reflection at the top of a layer from the reflection at the base. Within the wedge model, the thickness decreases from left to right, and the tuning thickness is at trace #35.

The synthetic seismic data was inverted for velocity using two methods. (1) conventional iterative GLI inversion and (2) neural network inversion. The conventional method we used for comparison (called blocky inversion) creates an impedance model whose forward synthetic matches the real data as closely as possible. (Hampson and Russell;1999). The software used to accomplish the conventional inversion is STRATA, a Hampson-Russell software product. The result of conventional inversion is shown in Figure 3-5.

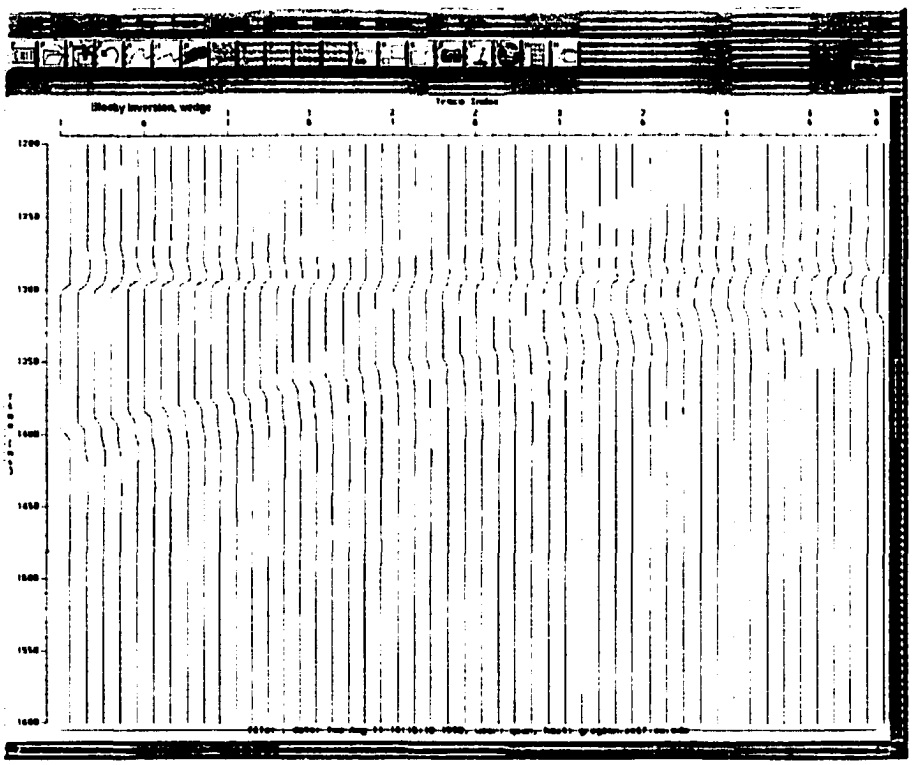


Figure 3-5. The result of blocky inversion

We can tell the outline of the wedge from the inversion result. From left to right the thickness of the wedge is decreasing before it reaches the trace #33. When the trace

number is larger than 33, the thickness of the wedge does not change. It means that the resolution of blocky inversion is about ¼ wavelength, the same as that of seismic data. From this figure we also can see that the top and the base of the wedge are not very clear. Especially at the base of the wedge, there are some "overshoots" and "undershoots" in the inverted impedance.

In inversion using neural networks, one seismic trace and one trace of low frequency data are used as input, and the velocity model for this trace is used as desired output to train the neural network. After the training, the neural network is used to calculate the velocity for the other traces comprising the entire wedge model.

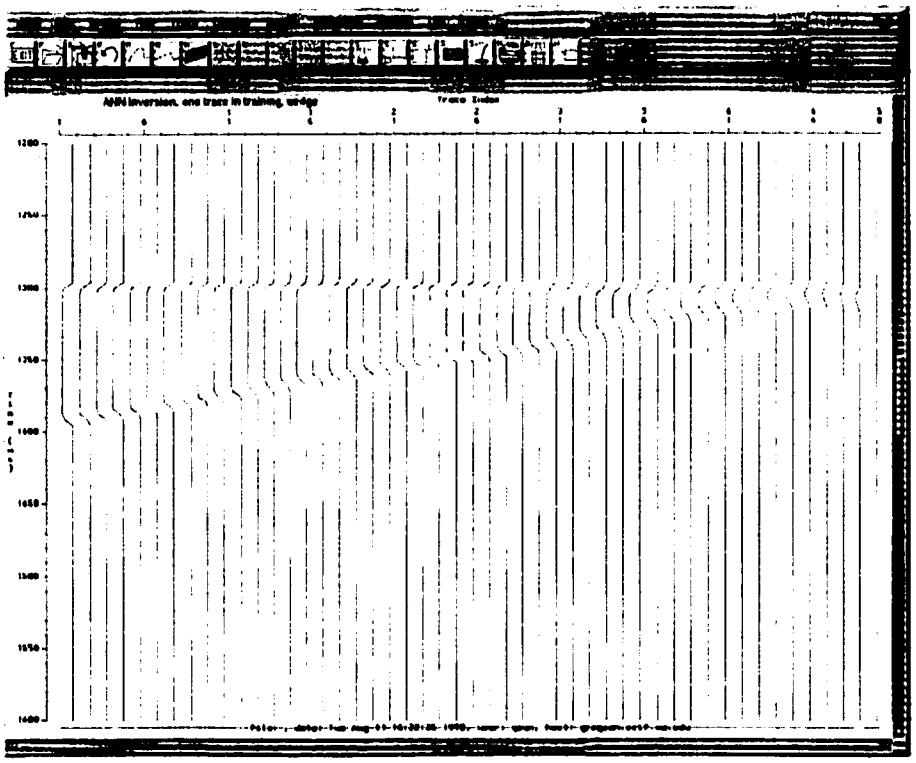


Figure 3-6. The result of inversion using neural network

Figure 3-6 shows the result of inversion using neural network. From this figure we can see that the top and base of the wedge are much clearer than those of the blocky inversion. From left to right, the thickness of the wedge is decreasing. Even when the trace number is larger than 35, the location of tuning thickness, the thickness of the wedge is still decreasing until it reaches the last trace. It means the resolution of the inversion using neural network can be much higher than that of conventional seismic inversion. The resolution is almost same as the sampling rate of the seismic data.

3.4 Noise sensitivity

In order to test the noise sensitivity of the inversion by neural network, 10% random noise was added to the seismic traces. With this noisy data, the performances of the blocky inversion and neural network inversion are very different.

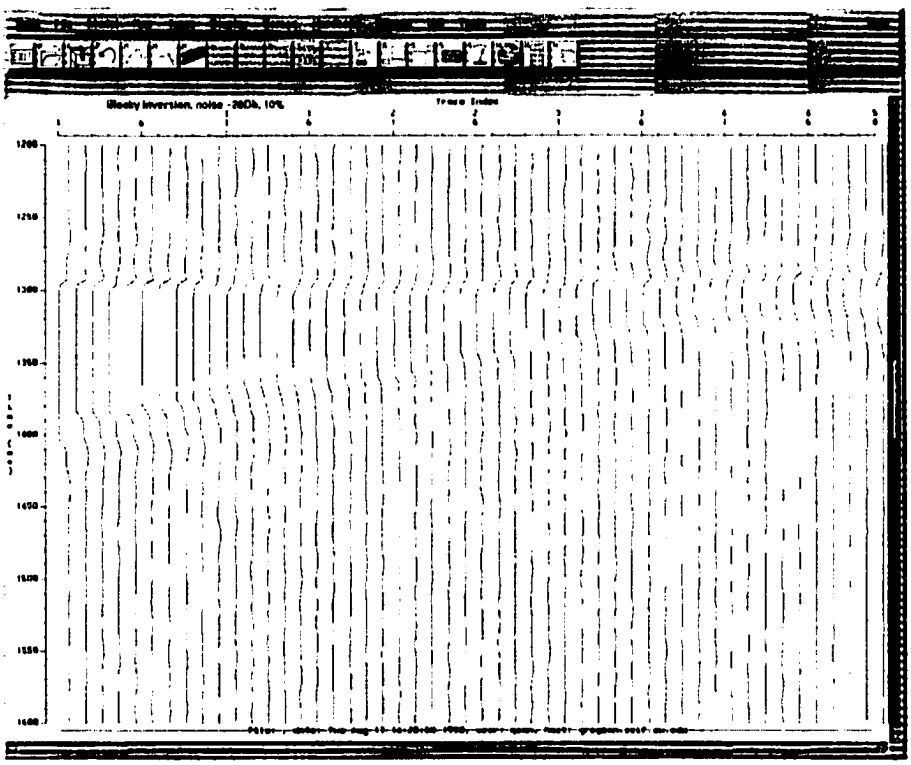


Figure 3-7. The result of blocky inversion with noisy seismic traces

The Figure 3-7 shows the result of blocky inversion with the noisy seismic traces. From this figure we can see that, with 10% noisy seismic traces as input, the result of inversion is also a little bit noisy. But we still can tell the outline of the wedge from the background.

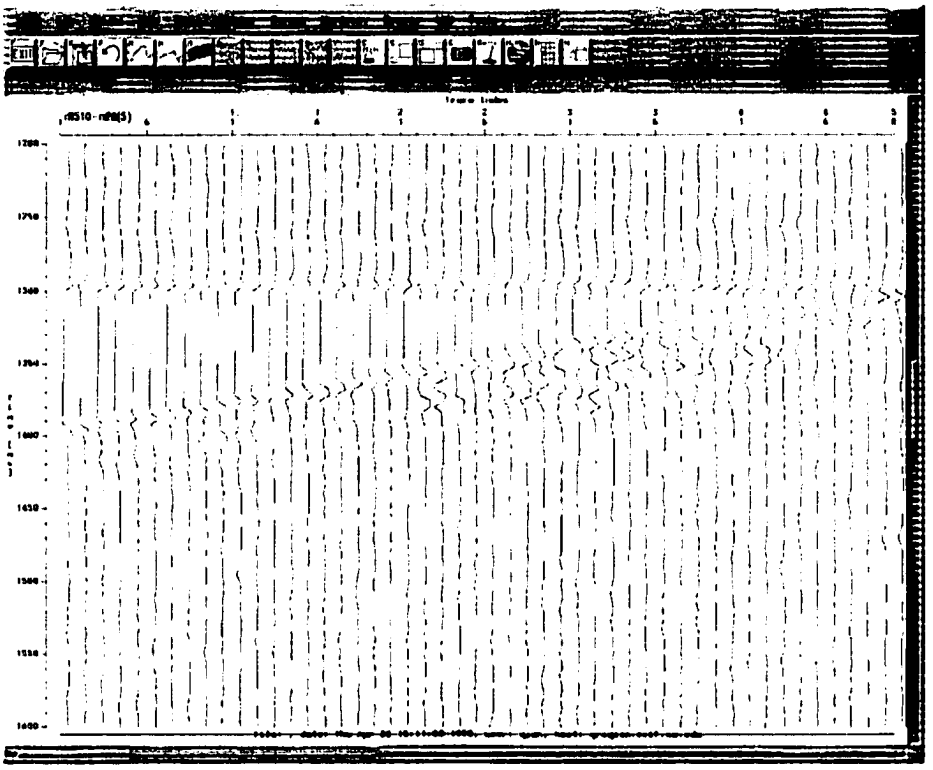


Figure 3-8. The result of inversion using neural networks, 10% noise was added before the inversion. Because we just use one random seed, the noise is not really random.

The Figure 3-8 shows the result of neural network inversion with the noisy seismic traces. From this figure we can see that, with 10% noisy seismic traces as input, the result of inversion is very poor. This inversion is very unstable. This is believed to be due to an artifact caused by the fact that the added noise is not really random, it is made by a formula. For the same random seeds, the neural network also can learn that formula; this is not what we want. To solve this problem, we use six different random seeds to create some noises, and use six traces in the training of neural network. After training, the

neural network calculated the velocity of whole section. Figure 3-9 shows this inversion result.

From the Figure 3-9, we can see the inversion result is significantly modified. The resolution and accuracy are the same as those without noise added. So the neural network can be trained to ignore the real random noise.

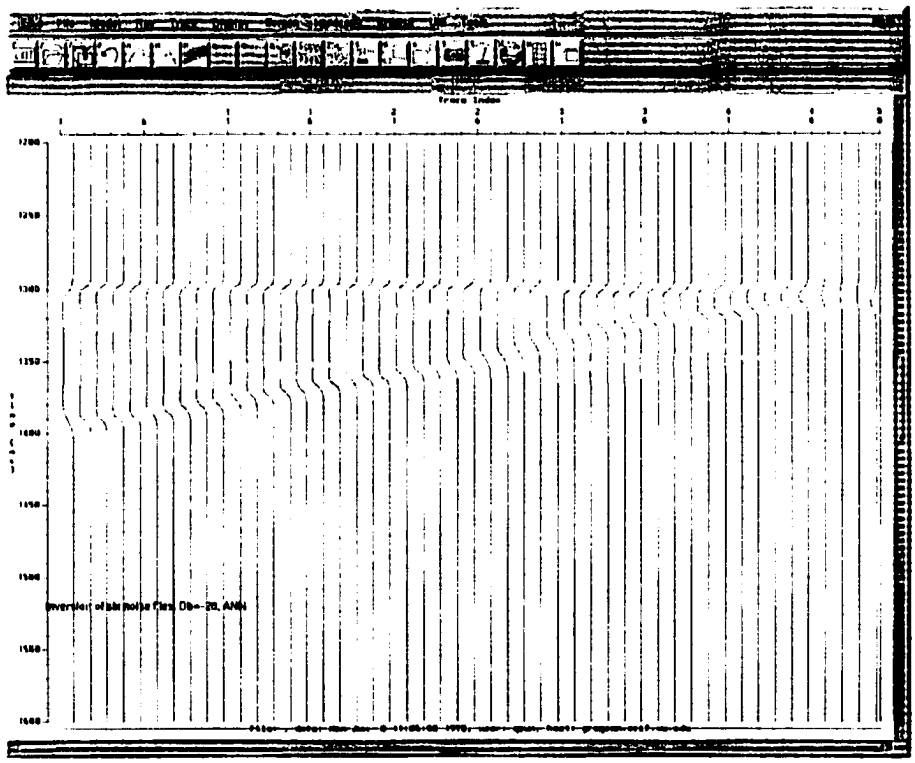


Figure 3-9. The result of inversion using neural networks, six traces with the same noise level (10%) but different random noise system (different random seeds) were used in the training.

Unlike conventional seismic inversion, there is no known forward model so model error is not introduced into the inversion results. When we train the neural network to find that model, we should consider what the neural network can learn and what it cannot learn.

3.5 Conclusion

1. Since the well logs have lower frequency components, we should provide the neural network with low frequency interpolation data.

2. Using the learning pair as one wavelength of seismic data correlated with one point well log data to train the neural network, we can raise the resolution of the inversion.

3. The accuracy and resolution of inversion using neural network can be higher than that using conventional inversion.

4. The neural network can be trained to ignore the real random noise.

CHAPTER 4. 2D SEISMIC INVERSION CASE STUDY, STRATTON FIELD

4.1 Introduction

A 2D seismic line extracted from the Stratton 3D seismic data set was used to test neural network inversion and to compare to multi-linear regression (using Hampson-Russell Emerge Version 1.11).

The extracted 2D seismic line crosses three wells; well 11, well 12, and well 18. (Figure 4-1) Each of which contains digital resistivity logs. This data set was used to test the hypothesis that neural networks can be used to invert for petrophysical parameters that are not directly linked to the seismic data. The inversion for resistivity is particularly interesting as high resistivity may be indicative of pay.

The well log data was converted to the time domain using sonic logs and one check shot. In order to match the seismic data sampling rate, the resistivity logs were smoothed and re-sampled to 2 ms.

The linear-regression option in Emerge has the same general procedure flow as the neural network procedure. Two wells were used in training and one well was used in testing. The conditions of training and testing for the neural network and multi-linear

regression methods were the same. Both methods were applied to the extracted 2D line.

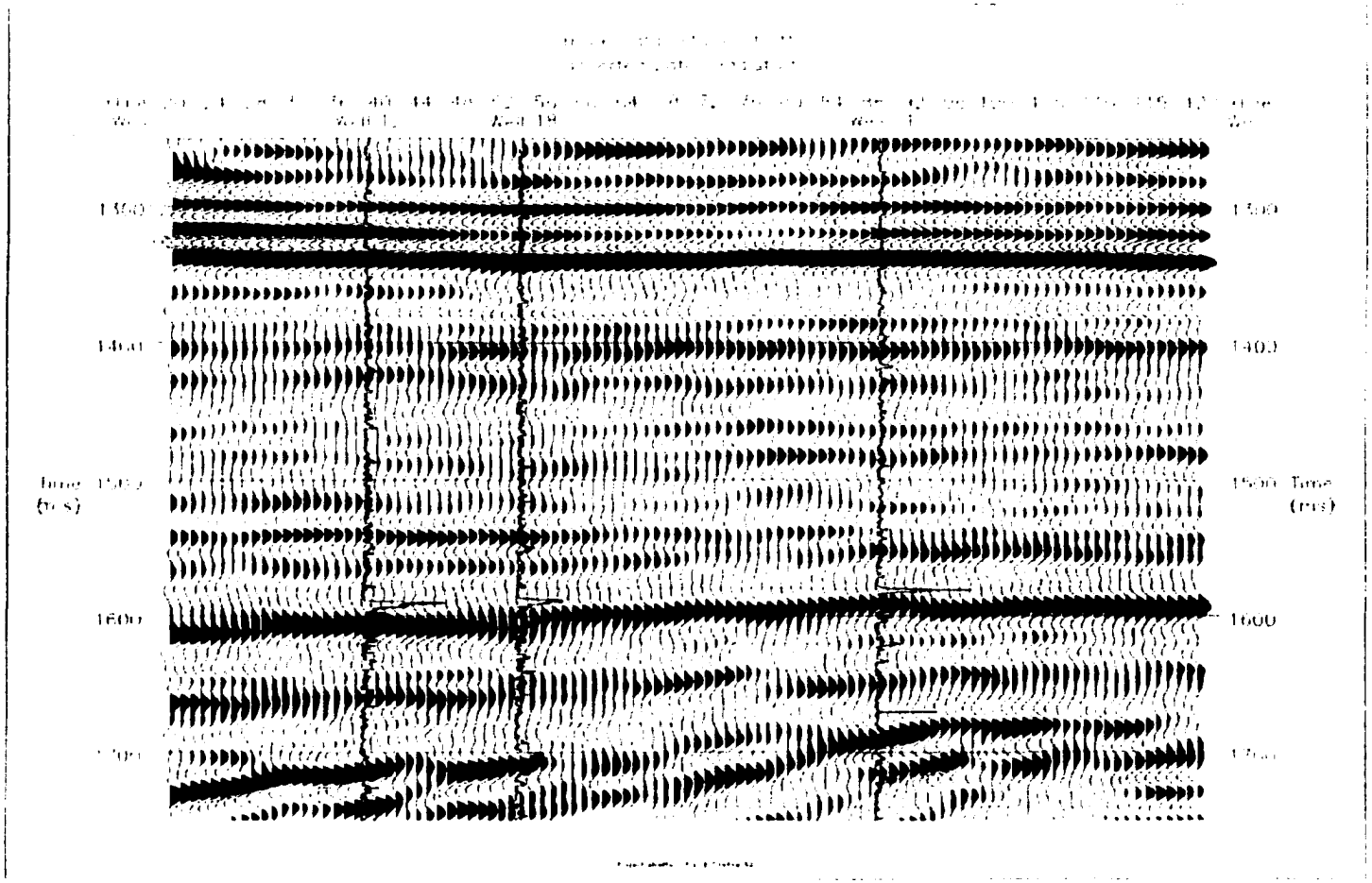


Figure 4-1. The seismic section and well log data used in the inversion of resistivity. Among the three wells, the middle one was used in the predicting, and the other two wells were used in the training.

4.2 Training

The neural network training algorithm is described in chapter 2. In order to compare with the Emerge multi-linear regression option, two flow charts were created. Figure 4-2 is a flow chart for the neural network approach used. Seismic data and the low frequency component of the well logs are used as input and the well log as desired output to train the neural network. The weights between layers of the network are modified iteratively until the differences between the output and desired output are less than a given level. After training, the seismic traces are input into the ANN algorithm and ANN predicts the well log response from the seismic data.

Emerge, version 1.11, uses the very same scheme as ANN, except for the training algorithm (Figure 4-3), it uses multi-linear regression. Emerge will extract seismic attributes from the raw seismic data and use the well logs which did not take part in the training to do the validation analysis. The function of validation analysis is to find the best seismic attribute combination. As shown in Figure 4-4, as the number of seismic attributes increases, the total error decreases, but the validation error decreases first and then increases. The point at which the validation error begins to increase is the best combination of seismic attributes. After training, Emerge 1.11 uses the same predicting scheme as the ANN algorithm to predict the well log response.

Wells 11 and 12 and the near well traces were used in the training, and well 18 was used in the testing.

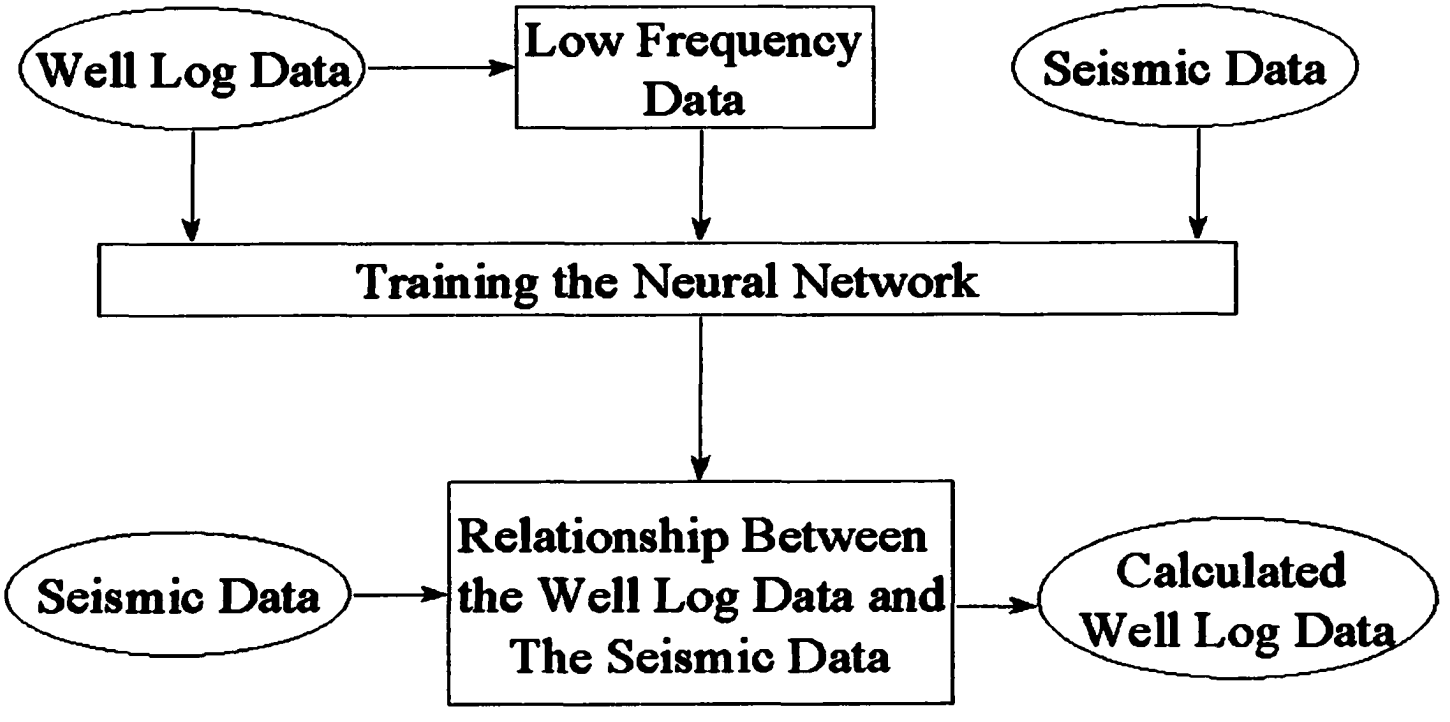


Figure 4-2. Flow Chart of Neural Network

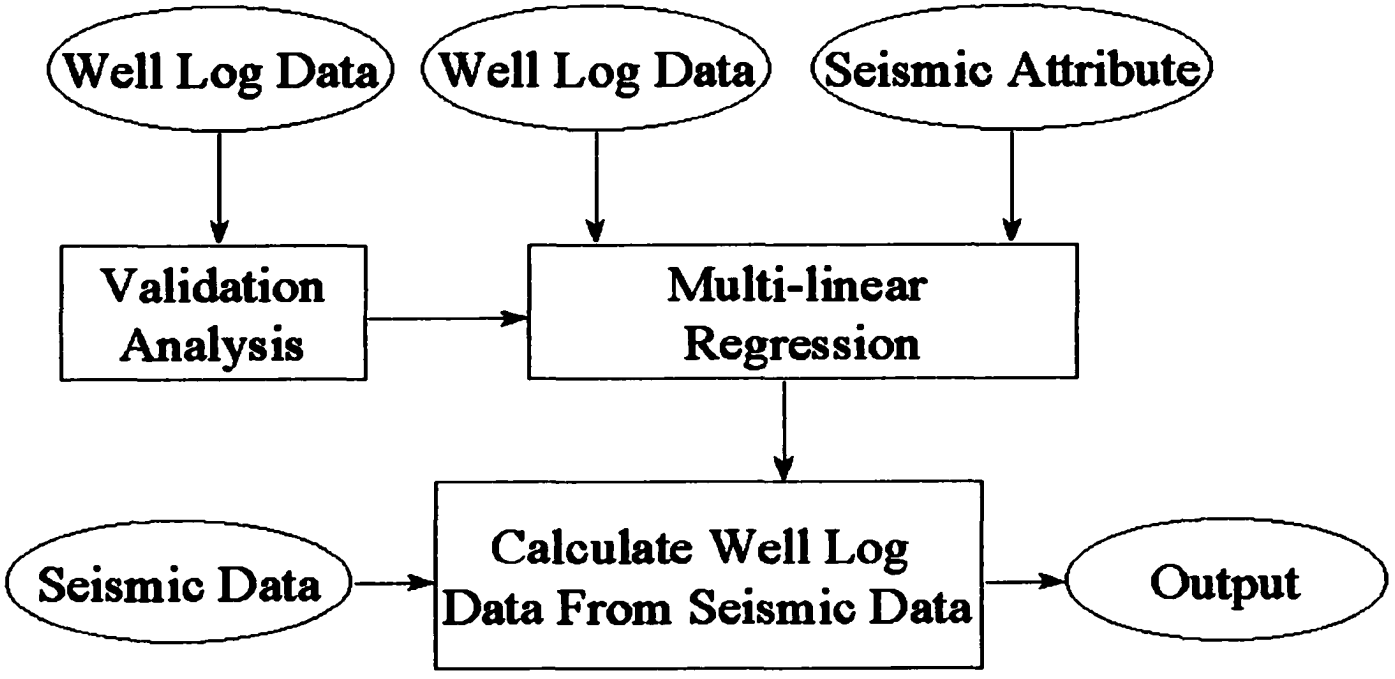


Figure 4-3. Flow Chart of Emerge

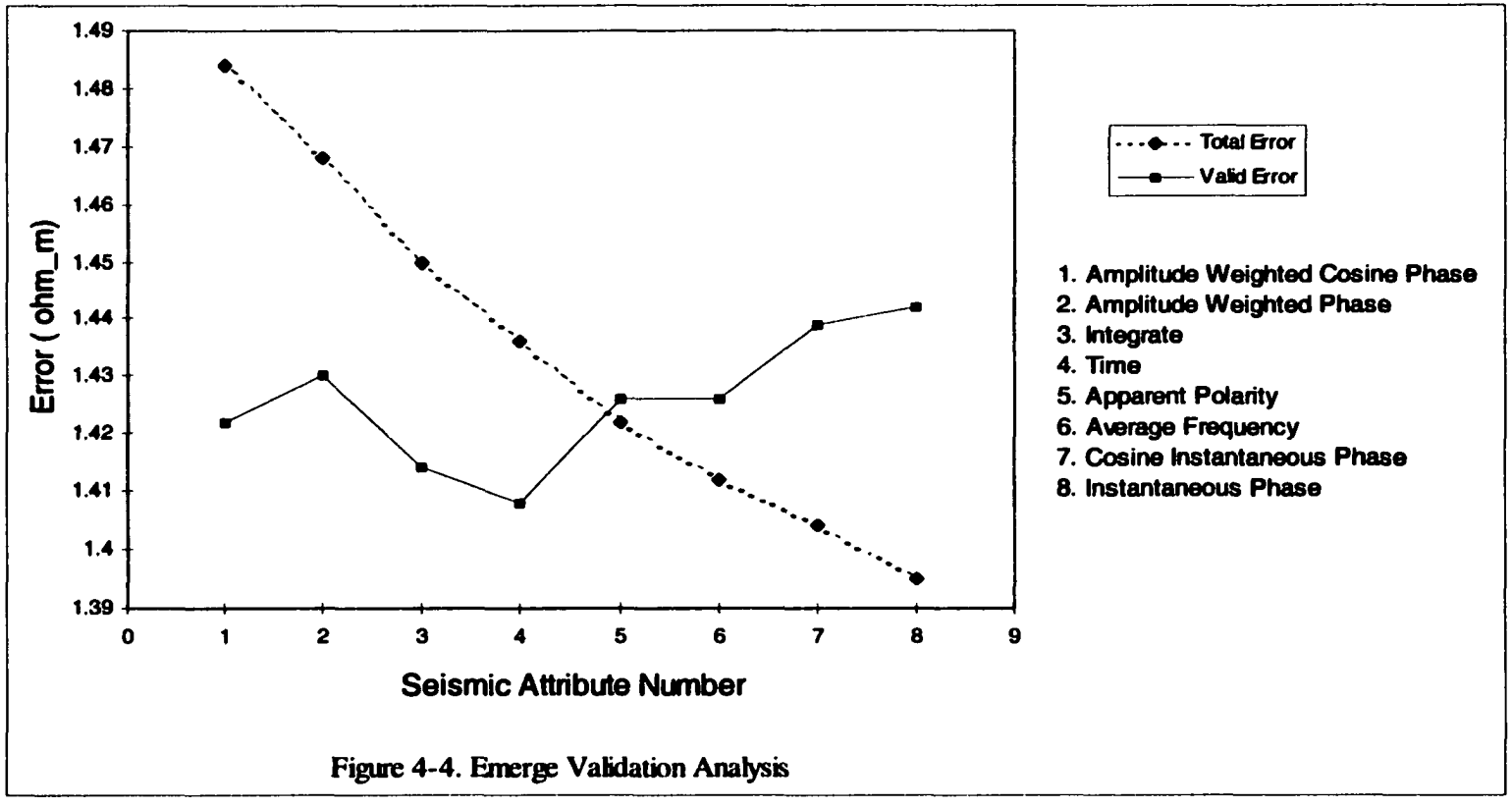


Figure 4-4. Emerge Validation Analysis

4.3 Results

For both inversion algorithms, training is the process to find the operators. Applying the operators on the seismic data, we can invert the seismic data into the well log data. Figures 4-5 to 4-10 show the results of inversion using both algorithms.

For the training wells (well 11 and well 12), Figure 4-5 and 4-7 show the result of inversion using the neural network, Figure 4-6 and 4-8 show the result of inversion using Emerge. For the testing well (well 18), Figure 4-9 shows the result of inversion using the neural network, Figure 4-10 shows the result of inversion using the multi-linear regression option of Emerge. From those figures we can see that neural network performed much better than multi-linear regression. It is apparent that the calculated resistivity logs by the neural network have higher resolution and accuracy.

With the neural network, the results of the inversion in the training window are better than those in the testing window. In the training window, we find that calculated logs almost match the true logs perfectly. In the testing window (at location of well 18, Figure 4-9), we can see that most of the time, the observed resistivities match the ANN calculated resistivities, the two obvious errors are the first big peak and last big peak. The mismatch for the last peak may be due to an error of depth-time conversion resulting from lateral velocity variations rather than a problem with the neural network inversion. If we stretch/shift the last peak of calculated resistivity about 15 ms, it will match that of observed resistivity.

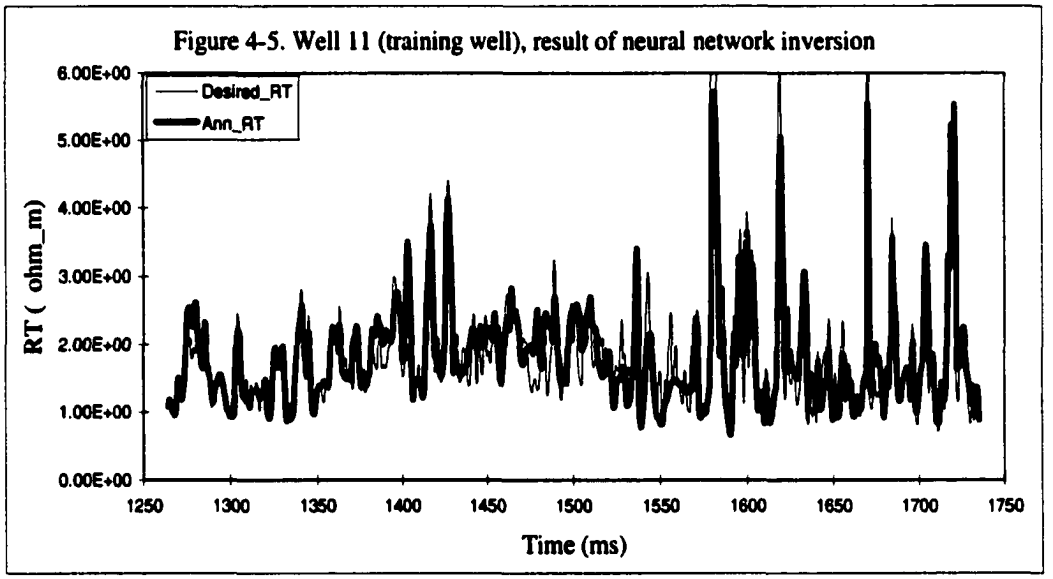


Figure 4-5. Well 11 (training well), result of neural network inversion

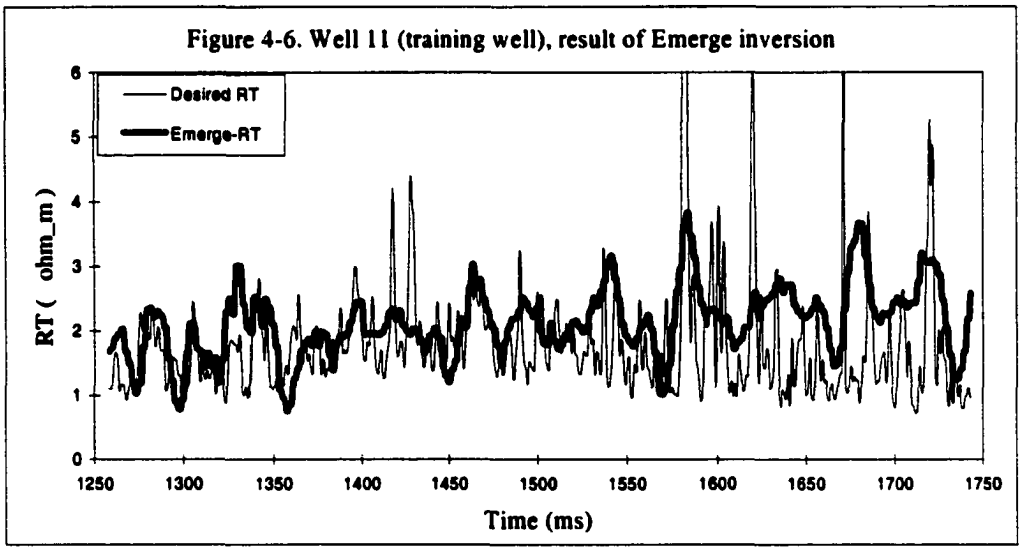


Figure 4-6. Well 11 (training well), result of Emerge inversion

Figure 4-7. Well 12 (training well), result of neural network inversion

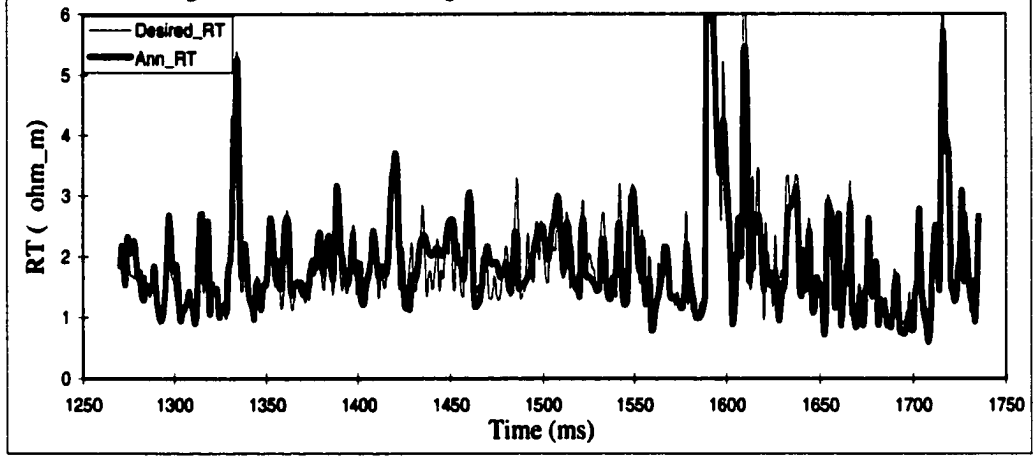


Figure 4-8. Well12 (training well), result of Emerge inversion

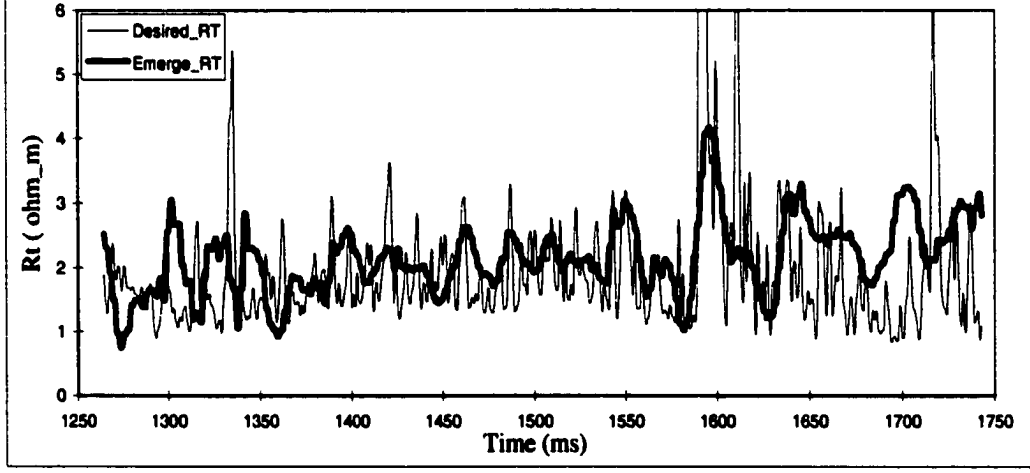


Figure 4-9. Well 18 (testing well), result of neural network inversion

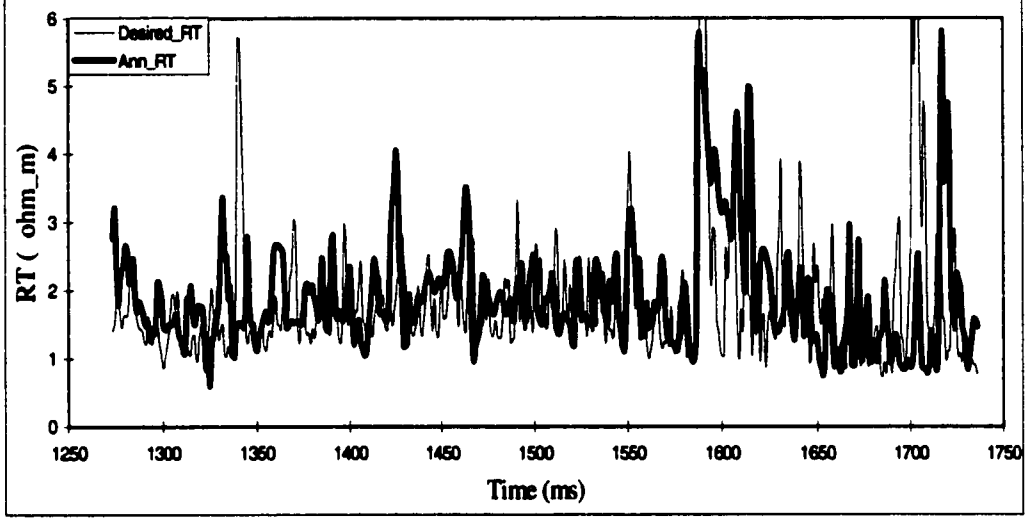
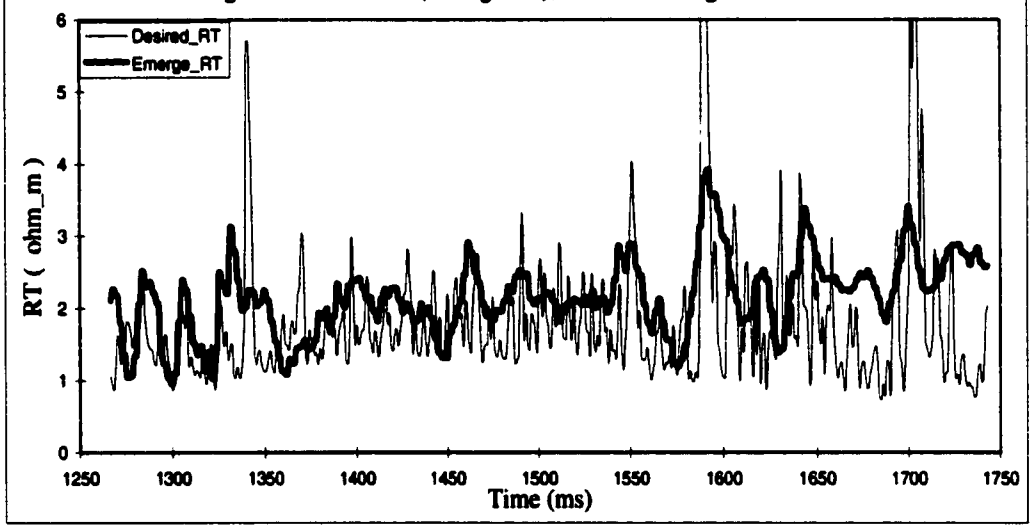


Figure 4-10. Well 18 (testing well), result of Emerge inversion



CHAPTER 5. 3D SEISMIC INVERSION

5.1 Introduction

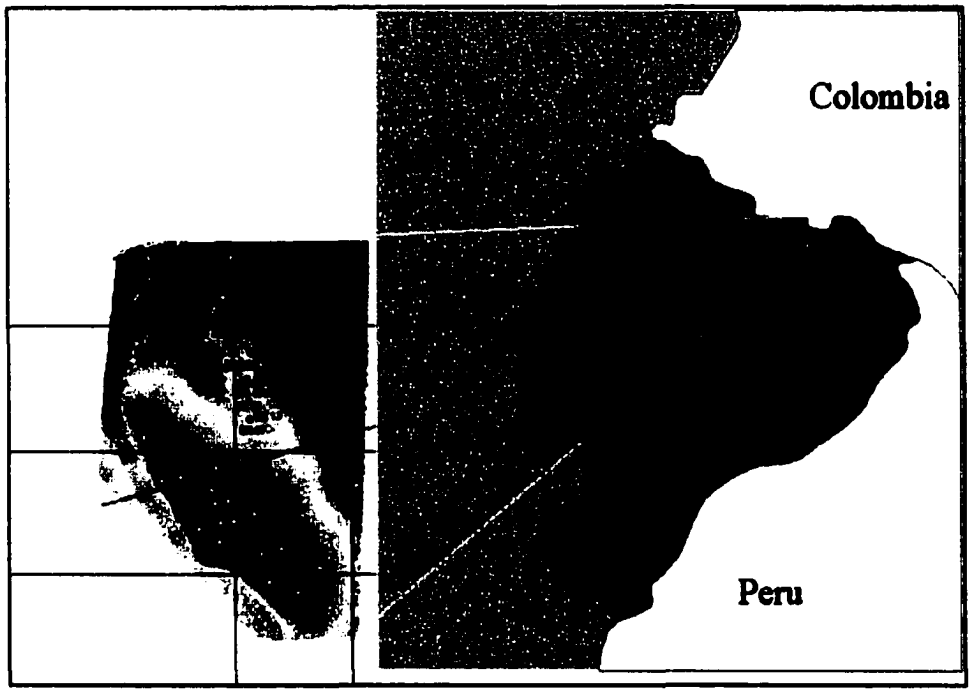


Figure 5-1. Location of Bermejo Field

As part of a cooperative reservoir characterization project between Tecpetrol and the University of Oklahoma, a reservoir model of the Hollin sand in Bermejo Field, Ecuador, was generated. The reservoir model required quantification of porosity throughout the volume, covering an area of approximately 9 x 13 km. Available data covering the area of interest included logs from 45 wells, and AGC migrated seismic

data. The seismic data was shot in 2000 and interpreted as part of a reservoir characterization project.

The Hollin formation is of fluvial to estuarine origin, and consists of interbedded sands and shales that are laterally discontinuous. The stratigraphy is further complicated by large- and small-offset faults, some of which were reactivated during basin inversion. Because of the complex stratigraphy and structure, simple interpolation of porosity among wells could not yield satisfactory results. Instead, it was necessary to rely upon seismic data to create a clear picture of and quantify lateral and to some extent vertical variability throughout the volume of interest. The neural network was verified to be a very good method to generate the porosity cube using the seismic data set.

5.2 Training

There are some more factors to consider in the training of 3D data set: (1) With the 2D data set, the near well seismic traces are selected along the 2D line. For the 3D data set, the near well seismic traces are selected in a circular area. The radius of the circular area is chosen to be large enough to include 5~10 traces in the area. (2) Spatial location may be considered in the 3D area. The relative X, Y coordinate value is provided as input for the neural network. (3) The low frequency data interpolated from well logs are also input as a cube.

There are 42 wells with porosity logs in the field. For the first step, 10 wells were in the testing, and the other 32 wells in the training. 5-10 near well seismic traces for each well were used in the training. The well logs were depth to time converted to the time domain. In order to match the seismic sampling, the well logs were smoothed and re-sample to 2ms. Figure 5-2 shows the inversion results in the training window, and figure 5-3 shows the inversion result in the testing window.

From these two figures, we can see the neural network inversion works well in both windows if slight time shifts are allowed with the exception of data above 1 second in the testing window, which does not compare favorably to the logs. The result in the training window is better than that in the testing window. In the testing window, the peaks and troughs of predicted and observed porosity are consistent, except that there are some shifts and stretches. These errors may due to the improper depth-time conversion.

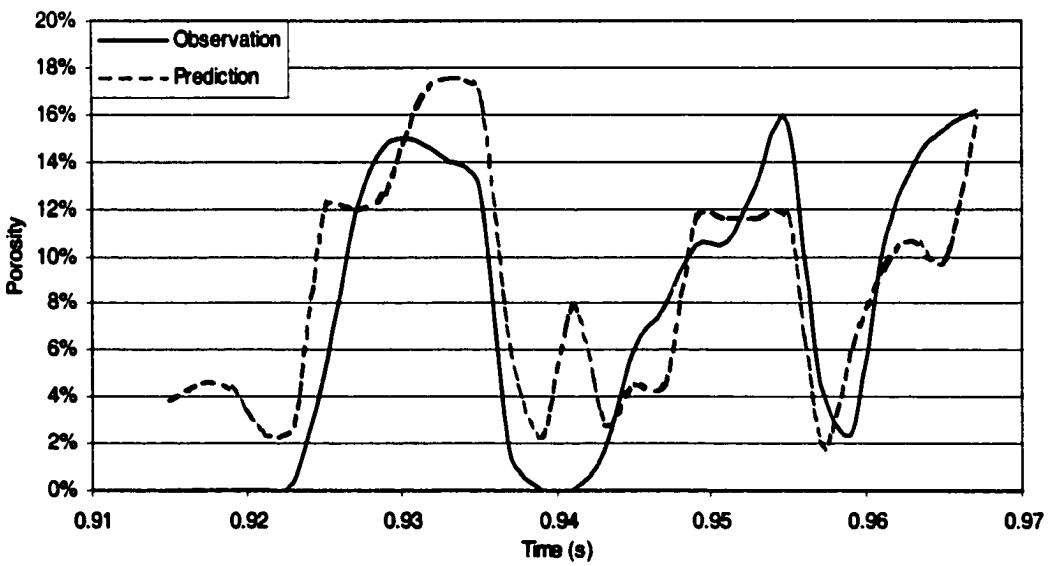


Fig. 5-2. BS-04 (training well), observed porosity vs predicted porosity

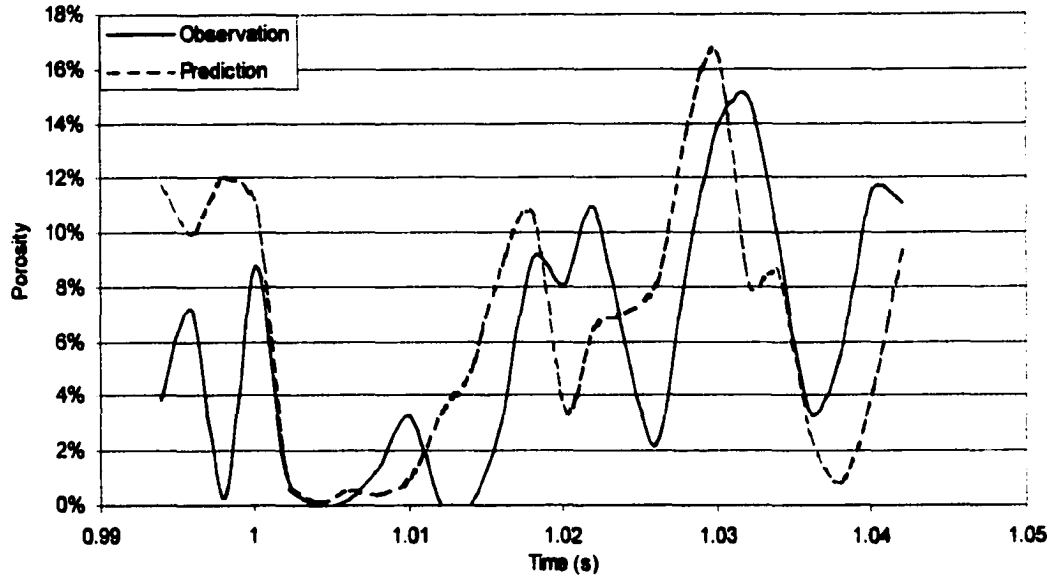


Figure5-3. BN-17 (predicting well), observed porosity vs predicted porosity

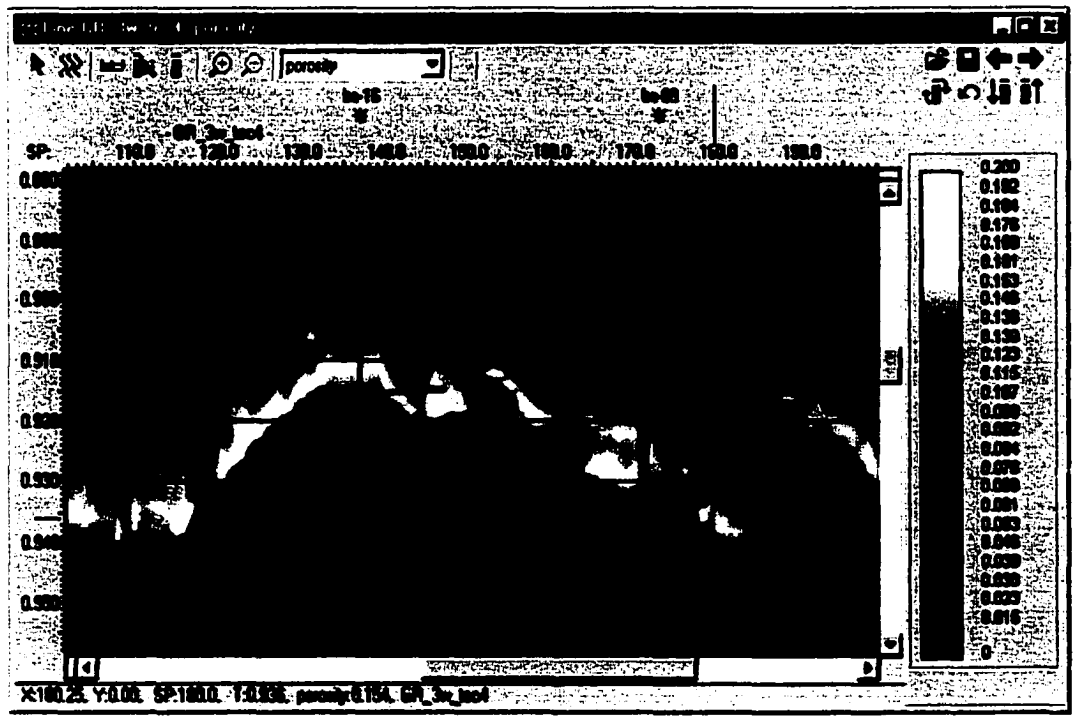


Figure 5-4. Inversion of porosity for the arbitrary line.

Figure 5-4 shows an arbitrary line in the 3D cube. The colored background is predicted porosity by ANN, and the well logs overlapped are the observed porosity. Well 16 is for the training, and well 09 is for the testing. From the figure, we can see that the observed porosity matches predicted porosity very well, for both the training and predicting wells. The predicted porosity for the whole section is geologically reasonable. The continuous sand strata, pinch outs and the faults can be interpreted easily.

For the second step all 45 wells were in the training to best match the actual porosity. Because the geological model is in depth, the seismic data was converted into the depth domain. The well logs were smoothed and re-sampled to match the seismic sampling rate. After the training, the porosity cube was generated by inputting the seismic data into the neural network.

5.3 The result of the inversion

Without the porosity cube from the seismic data, the porosity model used in reservoir characterization was generated by interpolation. Figure 5-5 shows the coverage of the interpolated porosity cube. In comparison the porosity cube from the seismic data using the neural network covered the whole 3D area. (Figure 5-6) The distance between the nearest two wells in the area is 500 ft, so the horizontal resolution of the interpolated porosity cube is very low. The porosity cube generated by neural network

has much higher horizontal resolution, which is 30 ft (the distance between two seismic traces). Using traditional seismic inversion, only an impedance cube can be directly obtained. Due to the inhomogeneous geological situation in this area, the impedance has poor relationship with the porosity as shale content, porosity, and pore fluids all affect the impedance. Figure 5-7 shows the impedance and the porosity for the same line. The log on the left is the gamma ray and the log on the right is the porosity. Two circles highlighted the same sand layer in the impedance section and porosity section. According to the impedance, the high porosity sand should be continuous. But the porosity calculated by the neural network shows the high porosity sand is not continuous. The production data from the well verified that the ANN calculated porosity was correct.

This ANN-generated porosity cube was used to directly re-evaluate reservoir quality and continuity near proposed well locations. This porosity cube has the advantage over the impedance cube in that it is able to pick up on differences between good reservoir sands and tight sands that the impedance cube could not resolve. In addition, the vertical resolution appears to be better than in the impedance cube. Another significant benefit of using ANN to determine the porosity cube is that it is able to ignore bad porosity data in an individual well (e.g., base of BS07) if there is sufficient well control nearby. Well BS07 had a bad porosity log for some reason. When used in the training this bad well did not distort the calculated porosity result very much. The good data of the wells nearby balanced the error the bad well brought in.

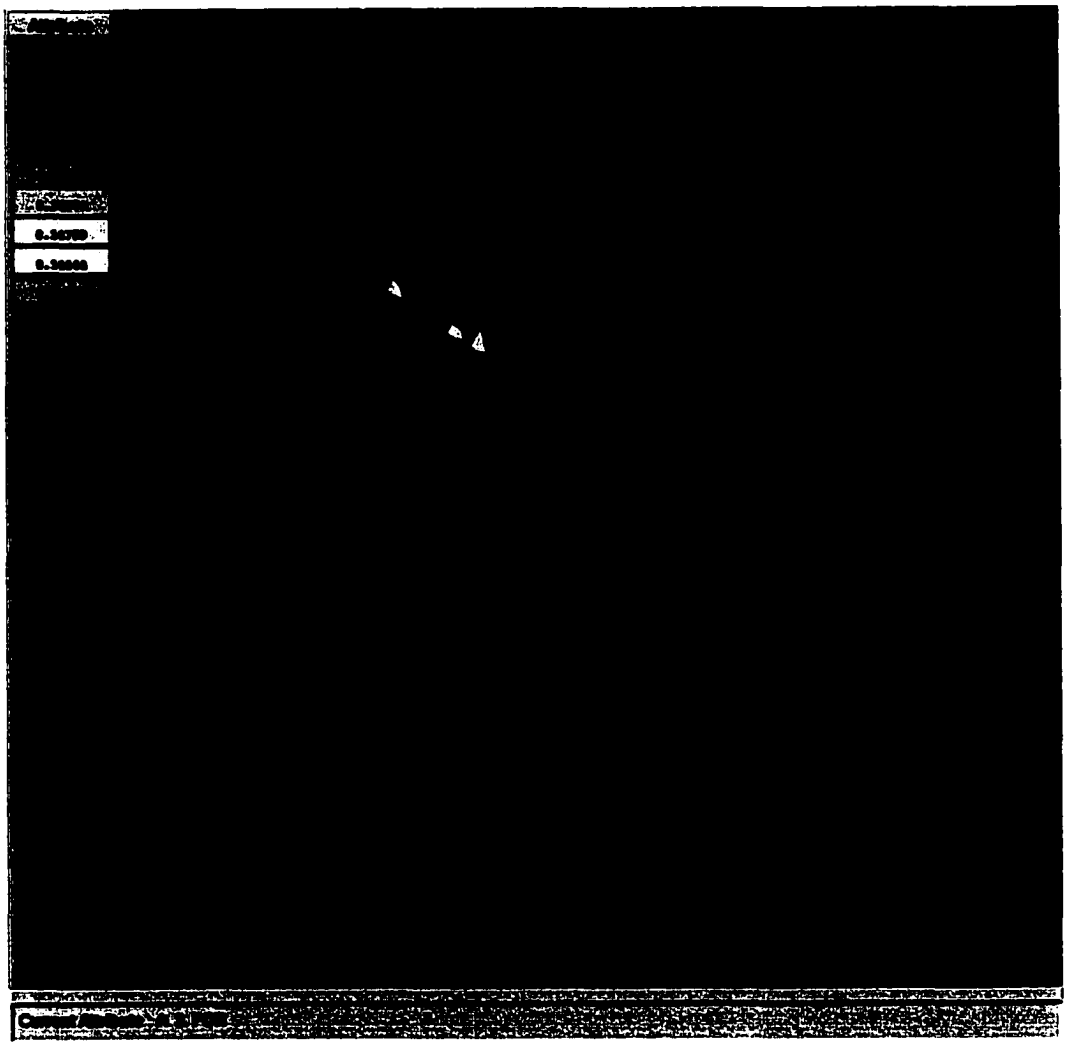


Figure 5-5. Interpolated porosity cube



Figure 5-6. Porosity cube calculated by neural network



Figure 5-7. Comparison of results of inversion for impedance and ANN calculated porosity cube. High acoustic impedance is shown in white, yellow and red, low acoustic impedance in blue and green. Right shows porosity cube for same interval; warm colors indicate high porosity and cool colors indicate low porosity. Gamma ray log is shown to the left of the borehole; porosity log is shown to the right in each figure.

CHAPTER 6. METHODOLOGY OF AVO ANOMALY DETECTION

6.1 Introduction

In this chapter, the use of artificial neural networks (ANN) for amplitude variation with offset analysis is investigated. A key aspect of AVO analysis is separating anomalies from the normal background, and recognizing the representative anomalies for different types of hydrocarbon reservoirs (eg., Castagna; 1993). The normal background is defined as the amplitude variation with offset (AVO) response within a given time and space window where hydrocarbon reservoirs are known (or presumed) not to exist. This background is expected to be specific to a given vintage of seismic data acquisition and processing and to be time and space variant (Castagna et al., 1998).

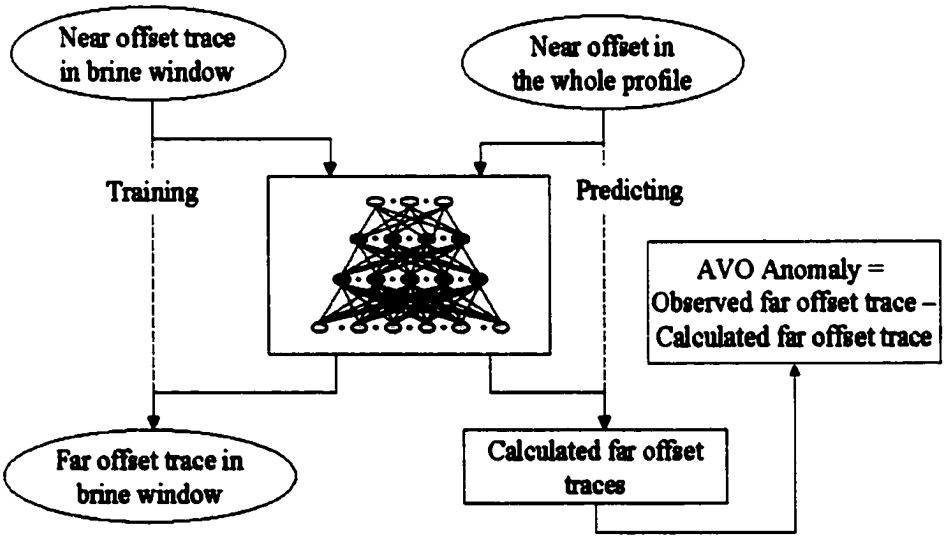


Figure 6-1. The flow chart of AVO inversion using neural network

Artificial neural networks are good at "learning" and "estimating" the mapping relationships between the data (Liu et al., 1998). Thus they can be trained to learn background AVO responses. Once an artificial neural network learns what is normal, it can be used to identify abnormal behavior.

Figure 6-1 shows the flow chart of AVO anomaly detection using the artificial neural networks (ANN). In order to teach the ANN what is normal, near offset partially stacked traces in the background window are used as the input and the far offset partially stacked traces in the same window are used as desired output in the training process. When training is complete, the far traces are predicted for the entire dataset being evaluated. The difference between the predicted traces and the observed far traces is an indicator of anomalous AVO behavior, possibly due to hydrocarbon reservoirs.

NMO correction is a very important processing step for the application of AVO technology. Small NMO error may not be important for stacked data, but will significantly influence AVO modeling and interpretation. In this chapter, several methods to make the neural network calculated AVO anomaly insensitive to incorrect normal moveout corrections are discussed.

6.2 AVO inversion for the synthetic model

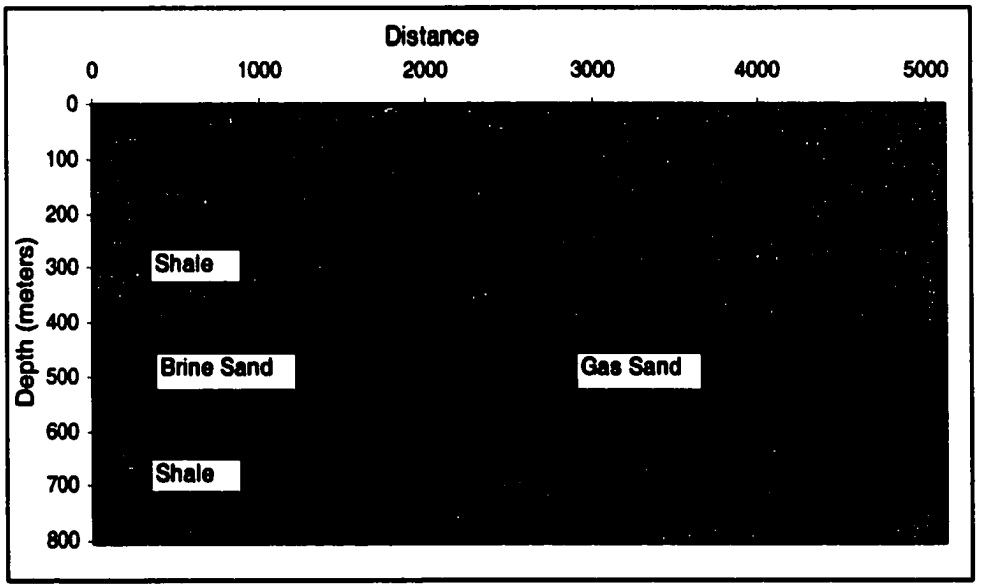


Figure 6-2. Synthetic Model

Lithology	Vp (km/s)	Vs (km/s)	Density (gm/cc)	Poisson's Ratio
Shale	3.24	1.62	2.34	0.3333
Brine Sand	2.59	1.06	2.21	0.3994
Gas Sand	1.65	1.09	2.07	0.1128

Table 6-1. Model parameters (Castagna et al., 1998)

A synthetic model (Fig. 6-2) was made using GXII, 2D tools. The zero offset ray trace was used as near trace, and the 500m offset ray trace was used as far trace. They were filtered by a 30 Hz Ricker wavelet. The traces were NMO corrected and migrated to correct position before the ANN training.

Two CDPs from a brine window were used to train the neural network. Inputs were the near offset of each CDP and desired outputs were the far offset of each CDP. After

training, the ANN predicted the far traces for the whole line by inputting the near offset traces. Three traces representing the training window, the brine window, and the gas window respectively were selected to show the testing results. Figure 6-3 shows that the predicted far traces fit the observed far traces very well in the training windows. Because the training window was selected from the brine window, the predicted far trace also fits the observed far trace in the brine window. (Figure 6-4) For this synthetic model, in the brine window the amplitude of near trace is less than that of far trace, but in the gas window, the amplitude of near trace is about same as that of far trace. This is the difference of the AVO behavior between the brine sand and gas sand.

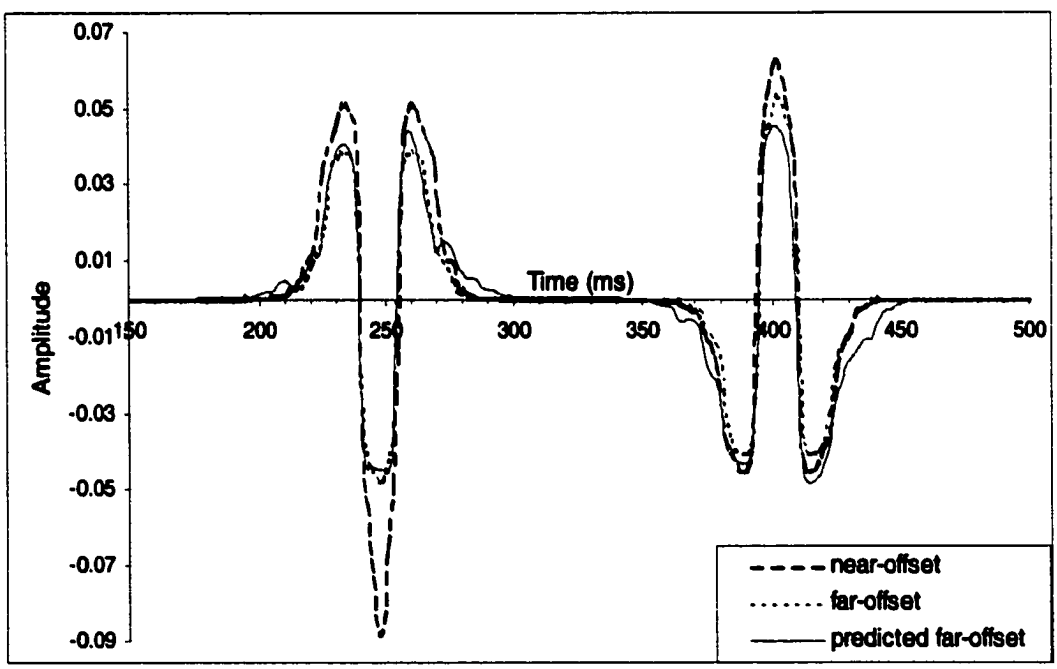


Figure 6-3. Training window AVO

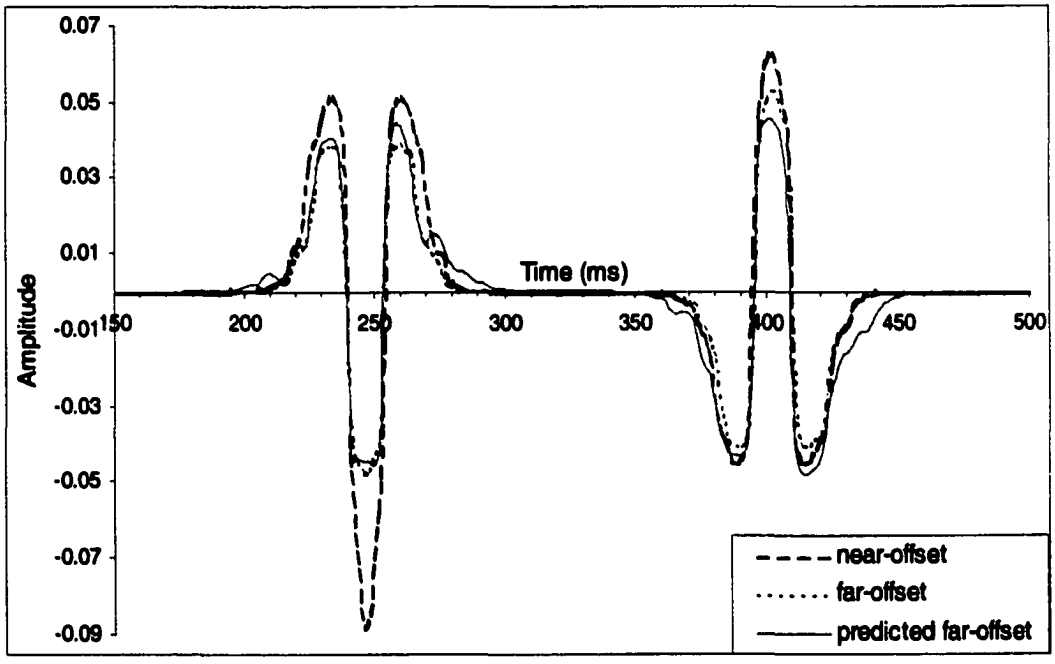


Figure 6-4. Brine window AVO

The trained neural network learned the relationship between the near trace and far trace in the brine window. This trained neural network was used to predict far traces in the gas window. The predicted far traces were very different from the observed far traces in the gas window. This is shown in Figure 6-5. The AVO anomaly was calculated by subtracting the predicted far trace from the observed far trace. Three traces of AVO anomalies from training, brine, and gas windows is shown in Figure 6-6. The anomalies for the training and brine windows are very small, approaching zero. The anomaly for the gas window is relatively large. Figure 6-7 shows the AVO

anomaly for the whole synthetic seismic section. From this figure, we also can see the AVO anomaly of gas sand is much larger than that of brine sand. So using the neural network inverted AVO anomaly we can discriminate the gas sand from the brine sand.

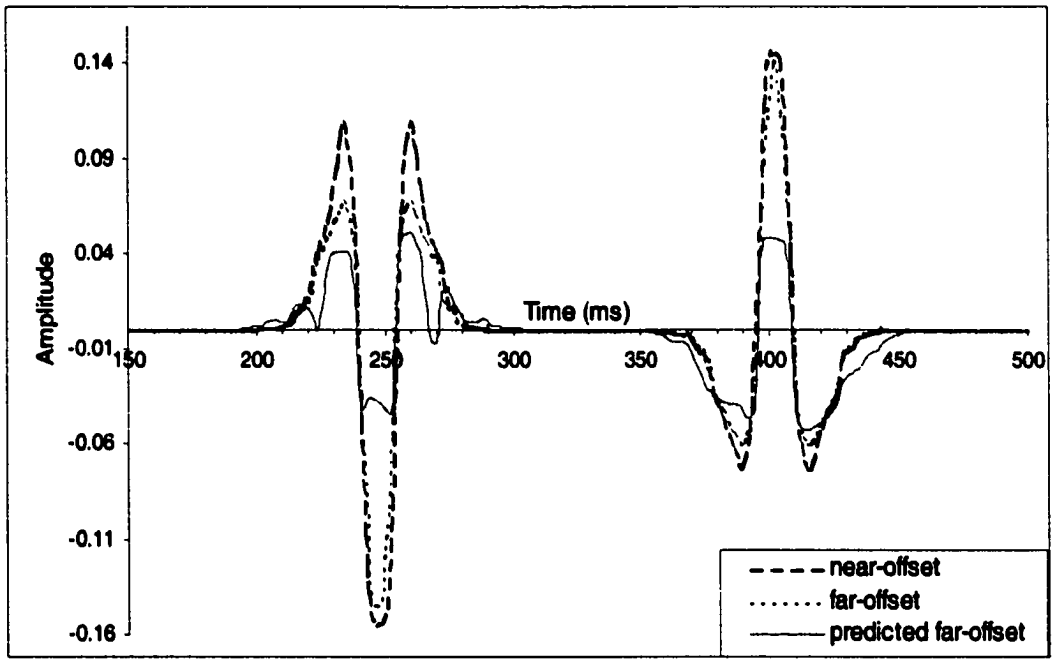


Figure 6-5. Gas window AVO

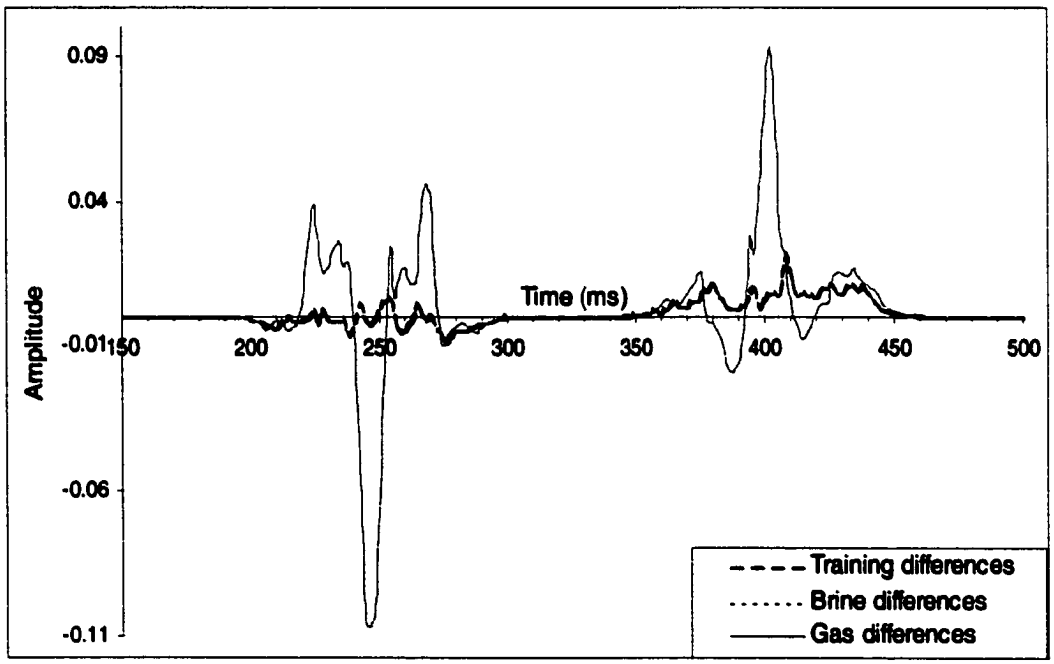


Figure 6-6. AVO anomaly

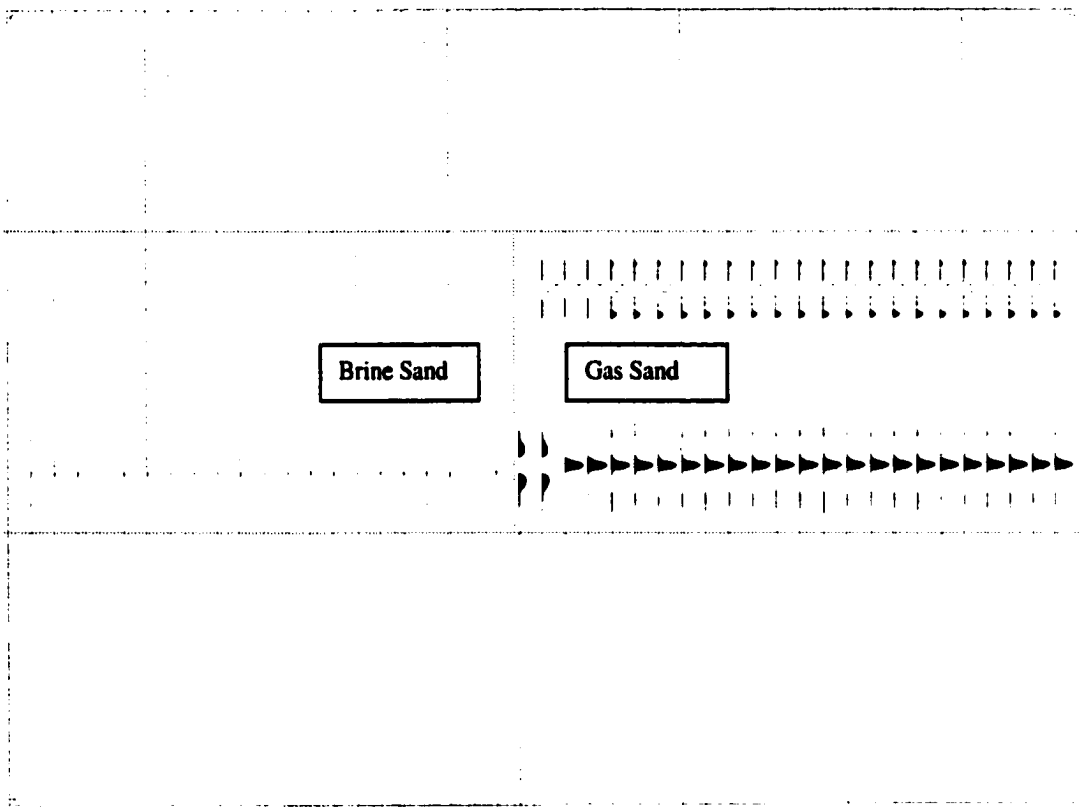


Figure 6-7. The AVO anomaly for the whole synthetic section

6.3 Synthetic model with constant NMO error

In the synthetic model above, it was assumed that the NMO correction is perfect. However, in the real case, the NMO correction will have more or less error. Whereas the method should properly learn to deal with NMO stretch and constant NMO error in the training process, variable NMO error must be handled differently.

Using the synthetic model above, with the addition of some NMO error, the effects of NMO errors can be evaluated. The far traces were shifted 8 ms downward relative to the near trace. This time shift between the near trace and far trace is same for the whole seismic section.

Two traces in the brine window were used in the neural network training. The near traces were used as the input, and the 8 ms time-shifted far traces were used as desired output. After training, the neural network calculated the far traces by inputting the near traces.

Figure 6-8 shows that the predicted far trace fit the observed far trace very well in the training window. With the shifted far trace, the predicted far trace also shifts to match it. Figure 6-9 shows that, in the brine window, the predicted far trace also fit the observed far trace very well. Figure 6-10 shows that, in the gas window, the amplitude of predicted far trace is smaller than that of observed far trace, but the locations of

peaks and troughs are the same. Figure 6-11 shows the AVO anomaly in training window, brine window, and gas window. We still can see that the anomalies in the gas window are much larger than those in the brine window. So, AVO inversion by ANN is insensitive to the constant NMO error.

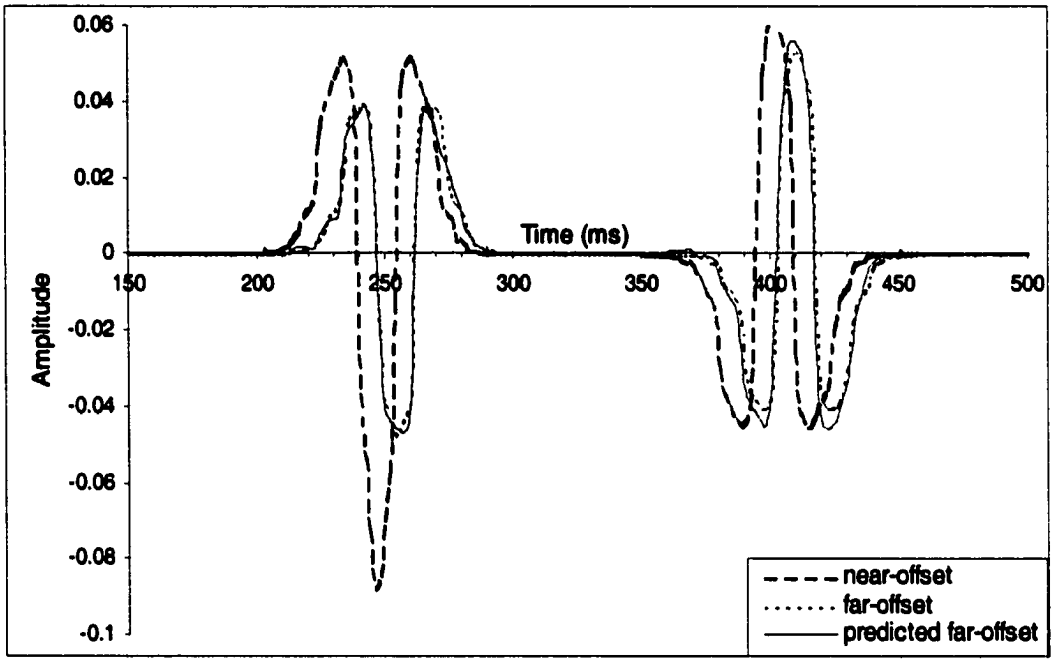


Figure 6-8. Training Window AVO, 8 ms constant NMO error

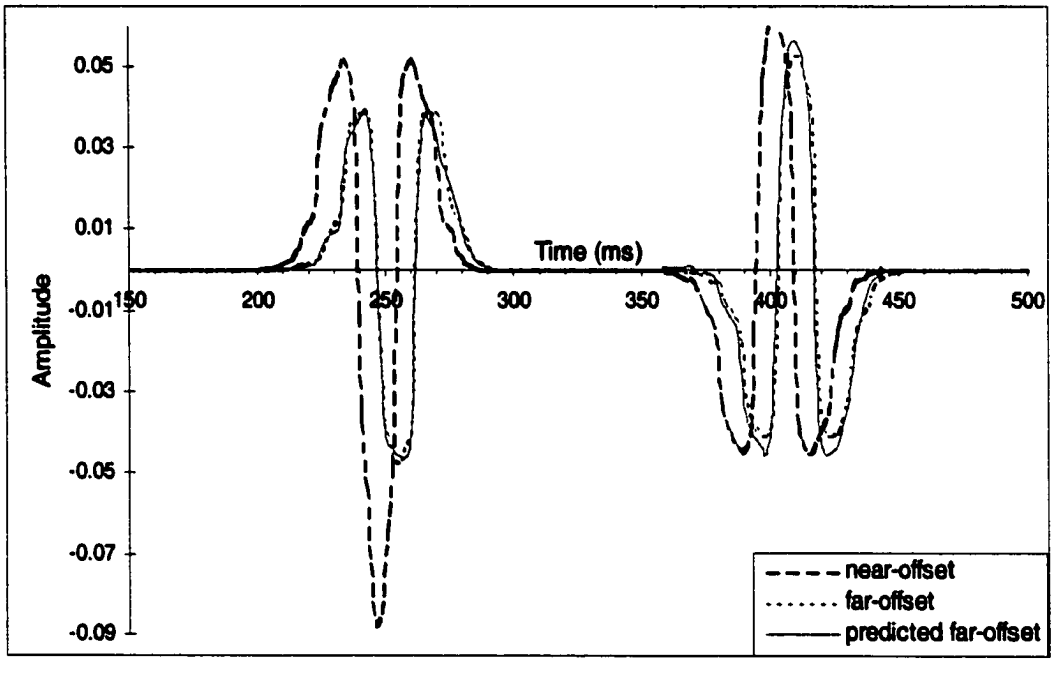


Figure 6-9. Brine Window AVO, 8ms constant NMO error

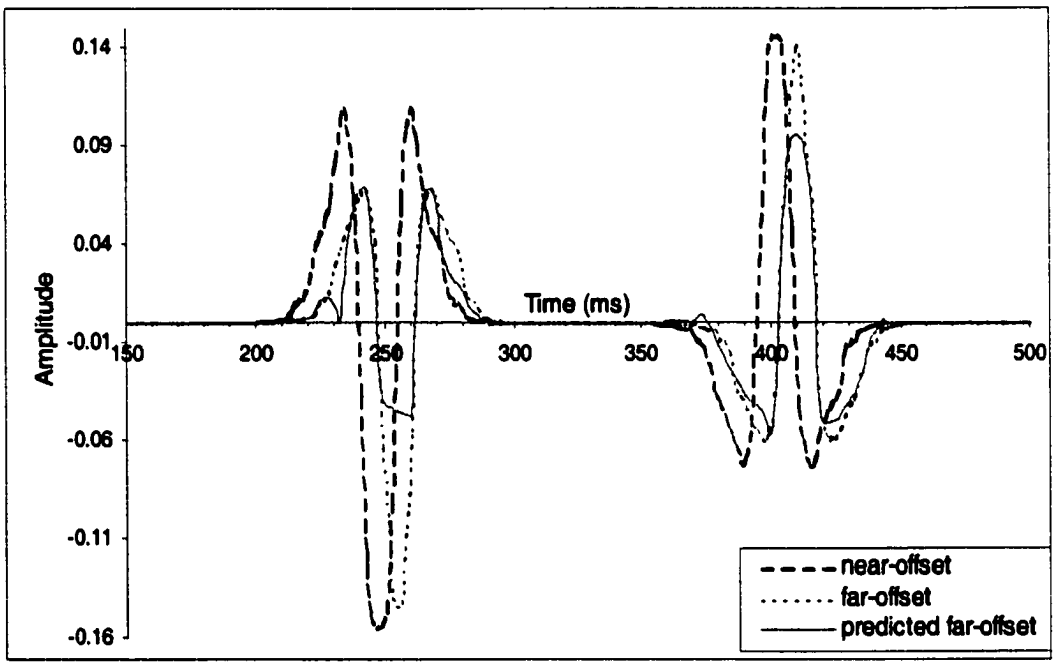


Figure 6-10. Gas Window AVO, 8ms constant NMO error

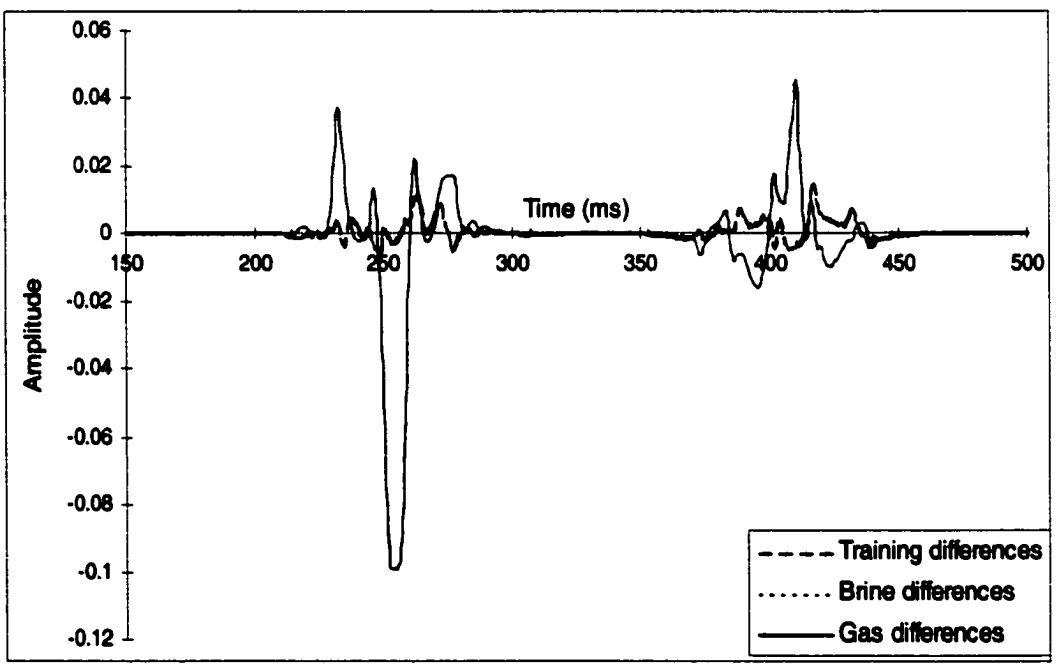


Figure 6-11. AVO Anomaly, 8ms constant NMO error

6.4 Synthetic model with variable NMO error

In the real data we see some constant NMO error. However, in more realistic cases, near and far offsets will not be perfectly aligned due to NMO errors, and this error will vary across the line. Thus, the training dataset may teach the neural network to expect a certain time shift between near and far offsets which may not be correct for a given CDP location. Using the synthetic model, this situation was simulated as follows: In the training window, there was no NMO error. In the brine window and gas window, far traces were shifted downward by 8 ms. So, after training and predicting, the predicted trace matched the observed trace in the training window (Figure 6-12), but in the brine window and gas window, there are 8 ms time shift between the predicted far trace and observed far trace. (Figure 6-13 and Figure 6-14) Upon subtraction of near and far traces, the result would be apparent AVO anomalies due simply to variations in NMO error. This problem also interferes with extraction of AVO parameters such as intercept and gradient, and may cause crossplotting techniques to fail. However, a simple modification, whereby observed and predicted far traces are cross-correlated prior to subtraction, readily solves this problem as illustrated below. Figure 6-15 shows the AVO anomaly in training window, brine window, and gas window after such correlation. This AVO anomaly can be used to discriminate the gas sand from the brine sand. Note that, because the cross-correlation is performed between predicted and observed traces, it is not confounded by differential tuning,

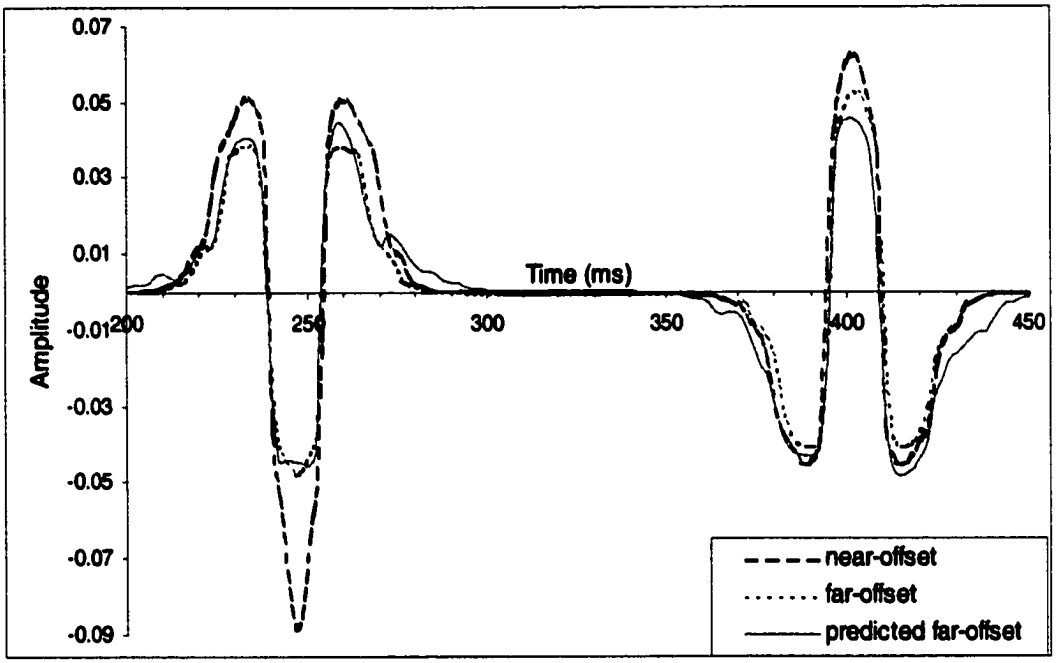


Figure 6-12. Training Window AVO, 8ms variable NMO error

stretch, frequency content, interference, etc. that would occur if an attempt were made to correct the NMO by cross-correlating with the near trace for example.

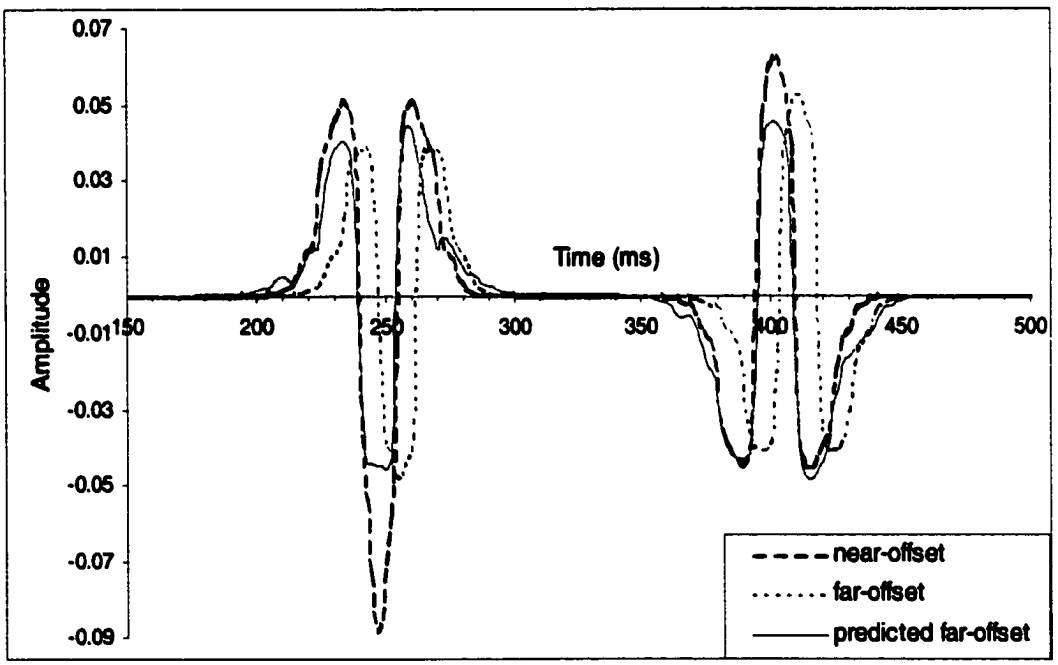


Figure 6-13. Brine Window AVO, 8ms variable NMO error

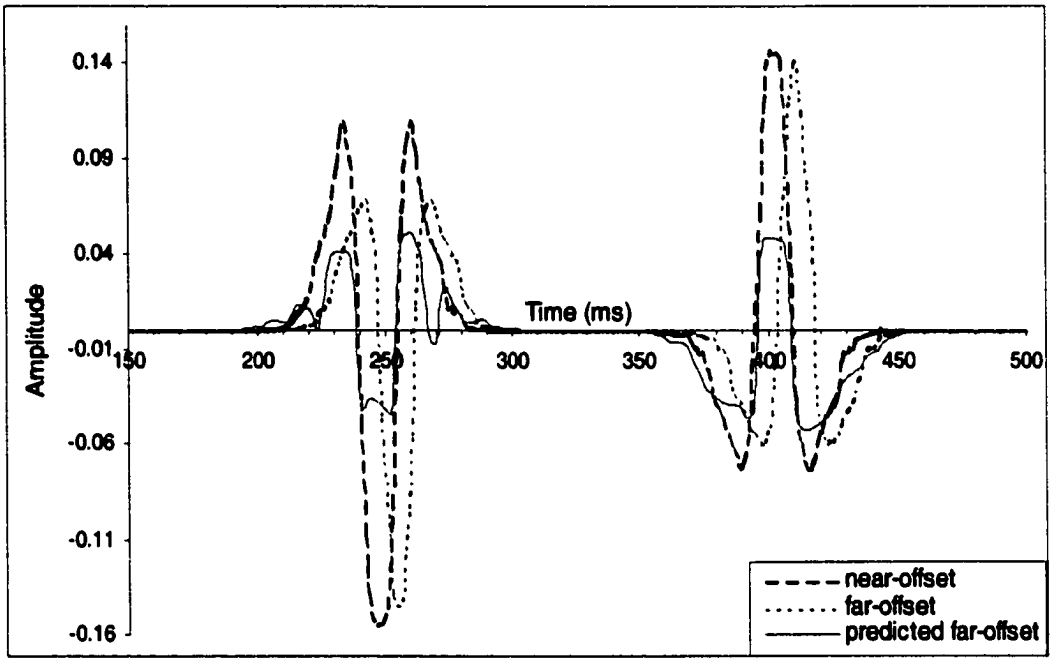


Figure 6-14. Gas Window AVO, 8ms variable NMO error

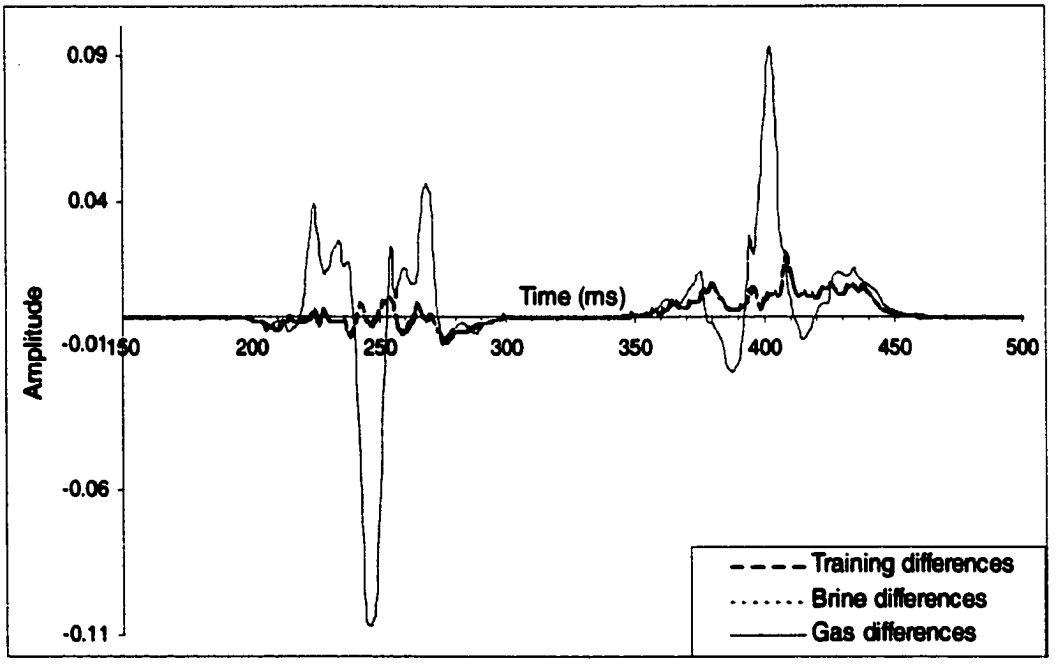


Figure 6-15. AVO anomaly, 8 ms variable NMO error, with correlation

6.5 NMO error suppression by phase unwrapping

Alternatively to the cross-correlation between the predicted far trace and observed far trace, there is another way to suppress the NMO error; called phase unwrapping. The basic idea of this method is that we assume the phase change is the factor to cause the NMO error, and the amplitude spectrum is independent of the NMO error. So, we use the phase spectra from the observed far traces and the amplitude spectra from the predicted far traces to compose new far traces, we call them composite far traces. The composite far traces should be NMO error free.

To test this technique, the synthetic model above with 8 ms variable NMO error, which means that there is no NMO error in the training window, and 8 ms downward shift of far traces in the brine and gas window. When the trained network was used to predict the far trace in the brine and gas window, there will be 8 ms time shift between the predicted far traces and observed far traces just as for the example above.

With the phase unwrapping, the ANN predicted far trace and observed far trace are used as shown in Figure 6-16. The predicted far trace by ANN with NMO error and the observed far trace are Fourier transformed, and their amplitude and phase spectra are calculated. Then the amplitude spectrum from the predicted trace and the phase spectrum from the observed trace are combined to calculate a new predicted trace by reverse Fourier transform. The NMO error is suppressed in this new calculated trace.

The reverse Fourier transform requires that the phase spectrum be unwrapped. This is accomplished by retaining the sign of the real part of the forward Fourier transform.

Fig. 6-17 shows the observed far trace and ANN predicted far trace in the brine window. Because of the variable NMO error, there is an 8 ms time shift between the observed far trace and predicted far trace. Fig. 6-18 shows the two traces after this phase unwrapping. In this figure, the combined far trace fits the observed far trace very well. NMO error is suppressed efficiently by this method.

However, there is a disadvantage of this method. The AVO behavior may include the phase change. After phase unwrapping, the phase spectra between the predicted far trace and observed far trace are enforced to be the same. So, it may ignore that part of the AVO anomaly between the predicted far trace and observed far trace caused by phase change. For this reason, when we interpret the ANN inverted AVO anomaly, the AVO anomaly calculated before the phase unwrapping should be used with the AVO anomaly calculated after the phase unwrapping as shown in the next chapter.

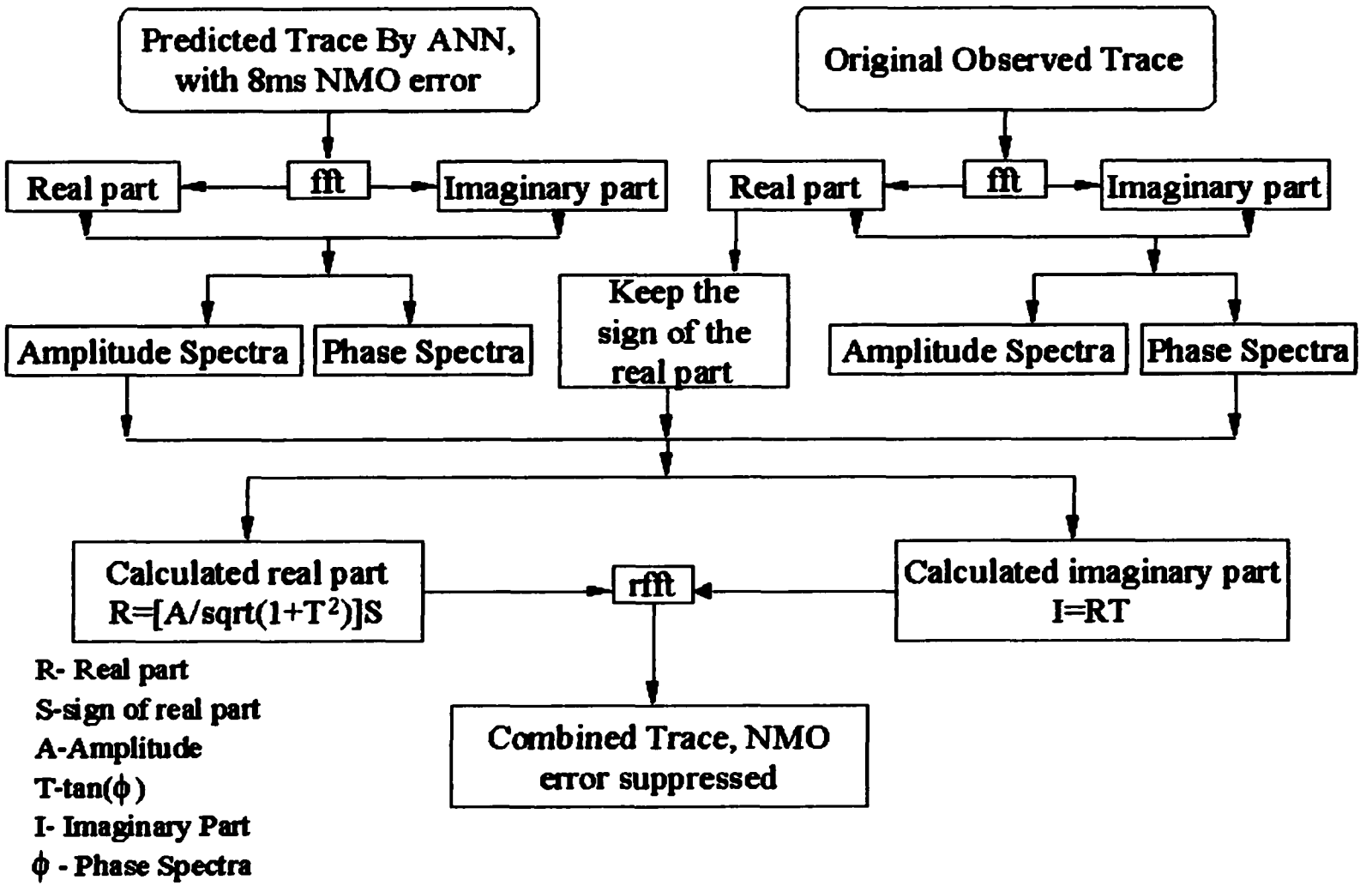


Figure 6-16. Flow chart of NMO error suppression by unwrapping the phase spectrum

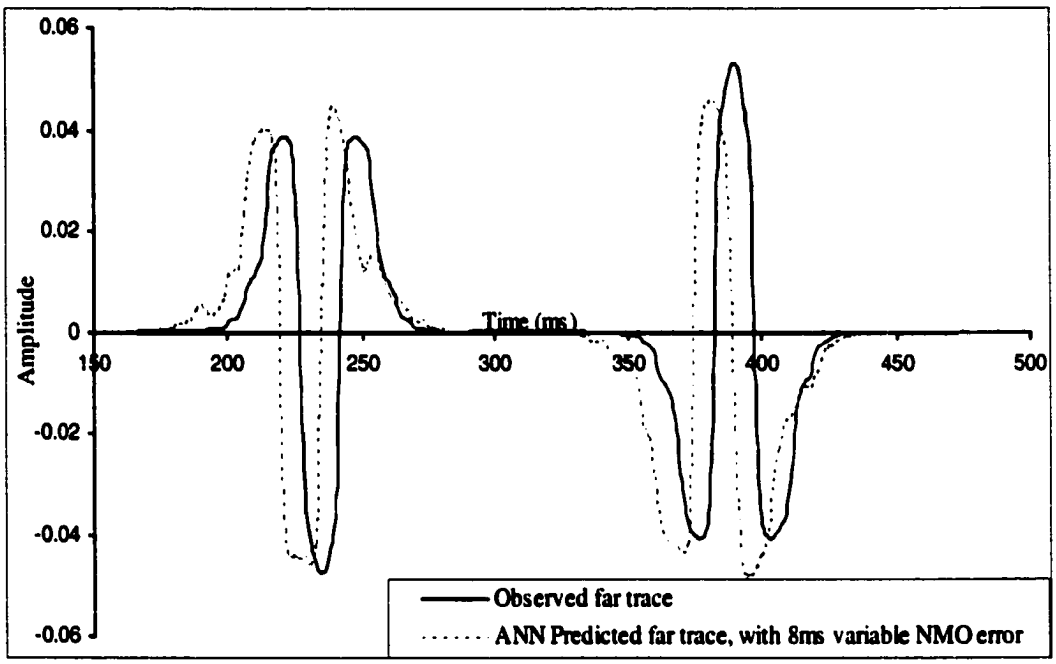


Figure 6-17. Brine window AVO, before the phase unwrapping

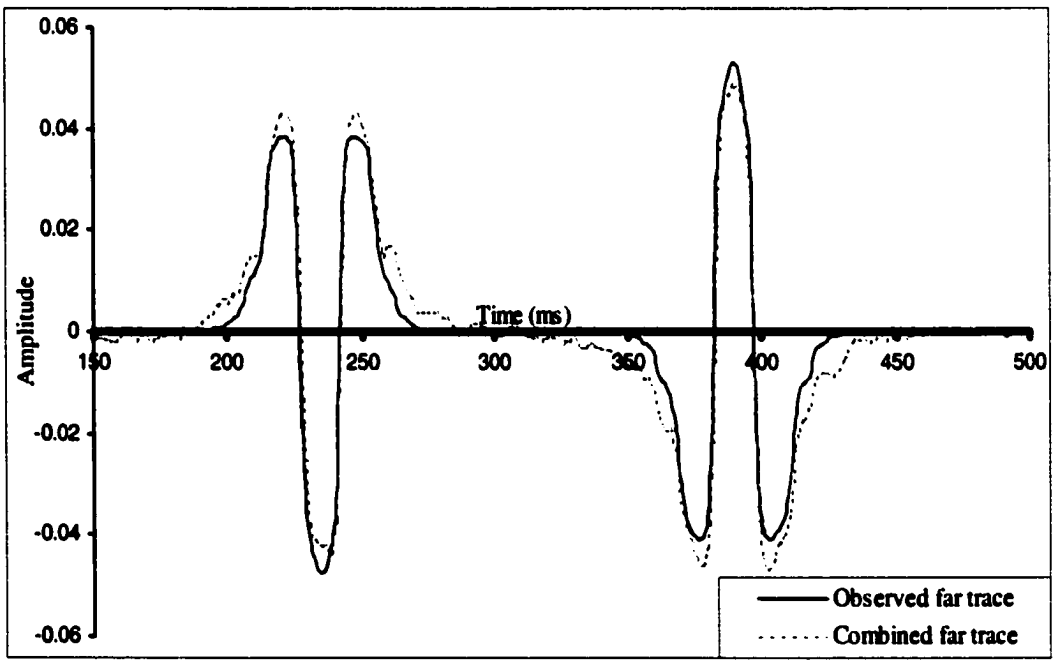


Figure 6-18. Brine window AVO, after the phase unwrapping

CHAPTER 7. AVO INVERSION FOR 2D SEISMIC DATA

7.1 Introduction

In this chapter two case studies of AVO inversion by artificial neural network for 2D seismic data using the same methods described in Chapter 6 are performed. With the real data, the near partial stack trace will be used as near trace, and the far partial stack trace will be used as far trace. In order to decrease the influences of depth and geological structure, the seismic data were flattened along the reservoir top.

The seismic data used had been previously interpreted. The reservoir locations were known before the application of AVO inversion. The reservoirs for both cases are gas reservoirs.

Partially stacked near traces and far traces in the brine window were used to train the neural network. After that, the trained neural network was used to predict the partially stacked far trace for the whole 2D section, and then the AVO anomalies (the differences between the predicted far traces and observed far traces) were calculated. For both cases, these anomalies were found to be much larger in the gas window than that in the brine window. Phase unwrapping was used in the second case to suppress the NMO error. The location and names of these two gas fields are confidential and will not be discussed here.

7.2 Case study 1

The 2D seismic line used has 900 CDP gathers. The CDP gathers were partially stacked into near traces and far traces. In order to decrease the influence of geological structure and depth, the stacked data, both near traces and far traces were flattened along the reservoir top. Figure 7-1 and 7-2 show the partially stacked near traces and far traces. The reservoir top is flat, and the flat spot is curved. According to the interpretation result, we knew that the gas reservoir window is from trace #1~#620, the brine window is from #621~#900. 140 traces (trace #621~#760) in the brine window were used to train the neural network. Near traces in this training window were used as the input and far traces in this training window were used as the desired output in the training. After training, the trained neural network was used to predict the far traces for the whole section (900 CDPs). The differences between the observed far traces and predicted far traces were then calculated. Figure 7-3 shows this AVO anomaly for the 2D line. From this figure we can see that the AVO anomaly is much larger in the gas window than that in the brine window. With the AVO anomaly, we can easily tell the reservoir top and base. To show this more clearly, three CDP's from the training window, brine window, and gas window were selected to show the predicted far trace, observed far trace, and AVO anomaly. Figure 7-4 shows the predicted far trace and observed far trace of one CDP in the training window. From

this figure we can see that predicted far trace and observed far trace match very well. Figure 7-5 shows the same things in the brine window. The observed and predicted trace also matched very well. Figure 7-6 shows the predicted far trace and observed far trace of one CDP in the gas window. These two traces did not fit at all. The differences between the observed far trace and predicted far trace are much larger in the gas window than that in the brine and training windows (Figure 7-7). It is interesting to note that large differences are also observed beneath the reservoir, presumably due to offset dependent transmission loss and other complicating factors.

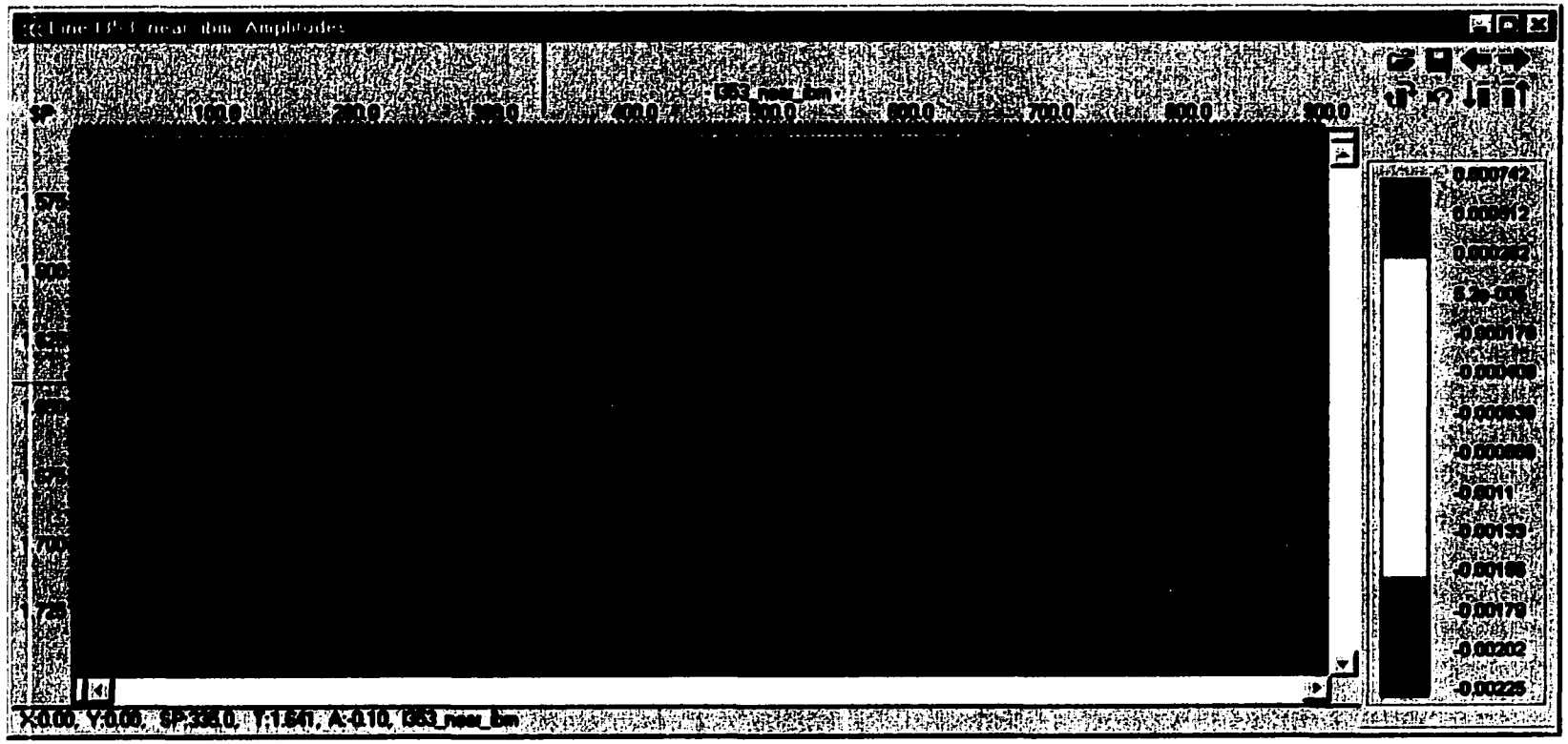


Figure 7-1. Near partially stacked traces for the 2D seismic data set

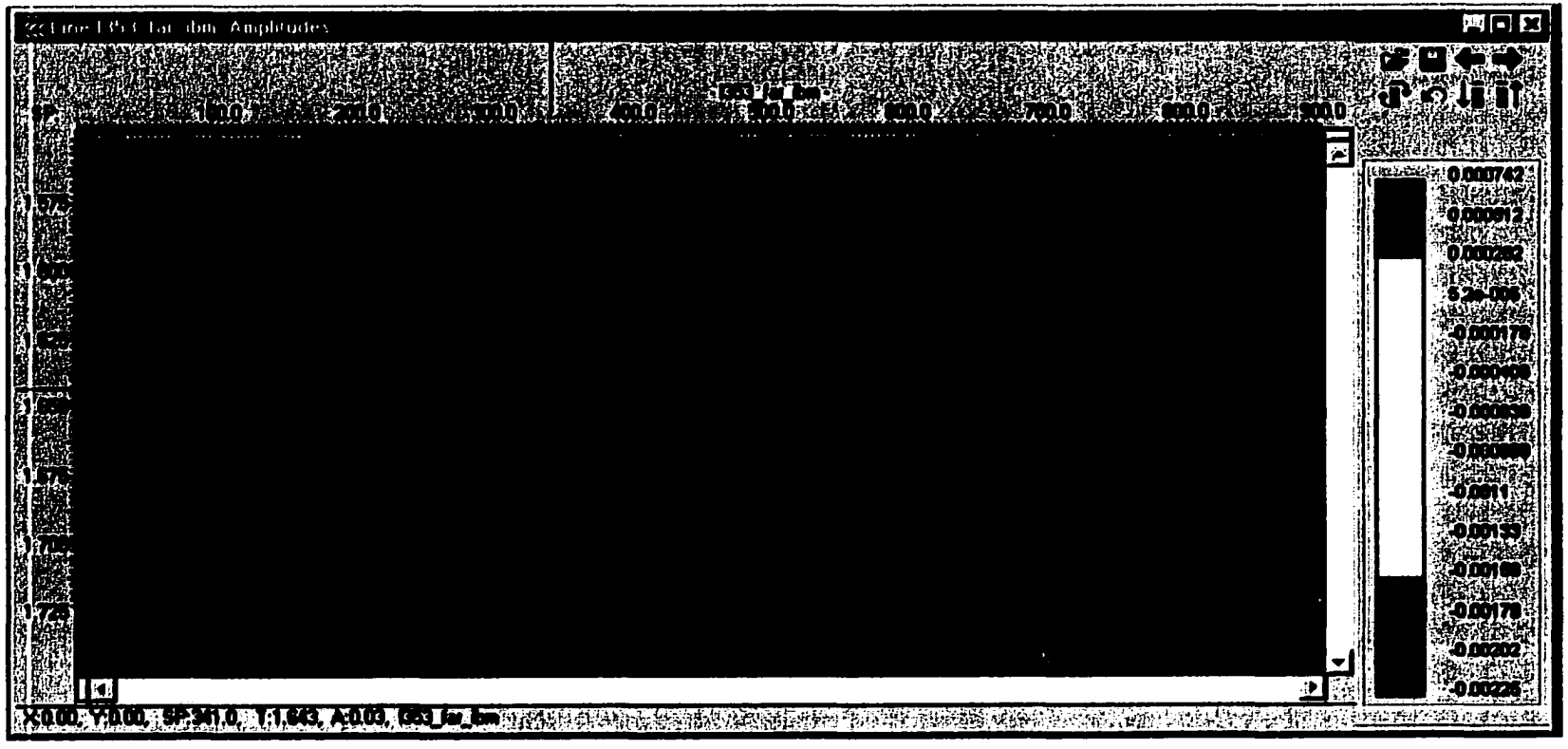


Figure 7-2. Far partially stacked traces for the 2D seismic data set

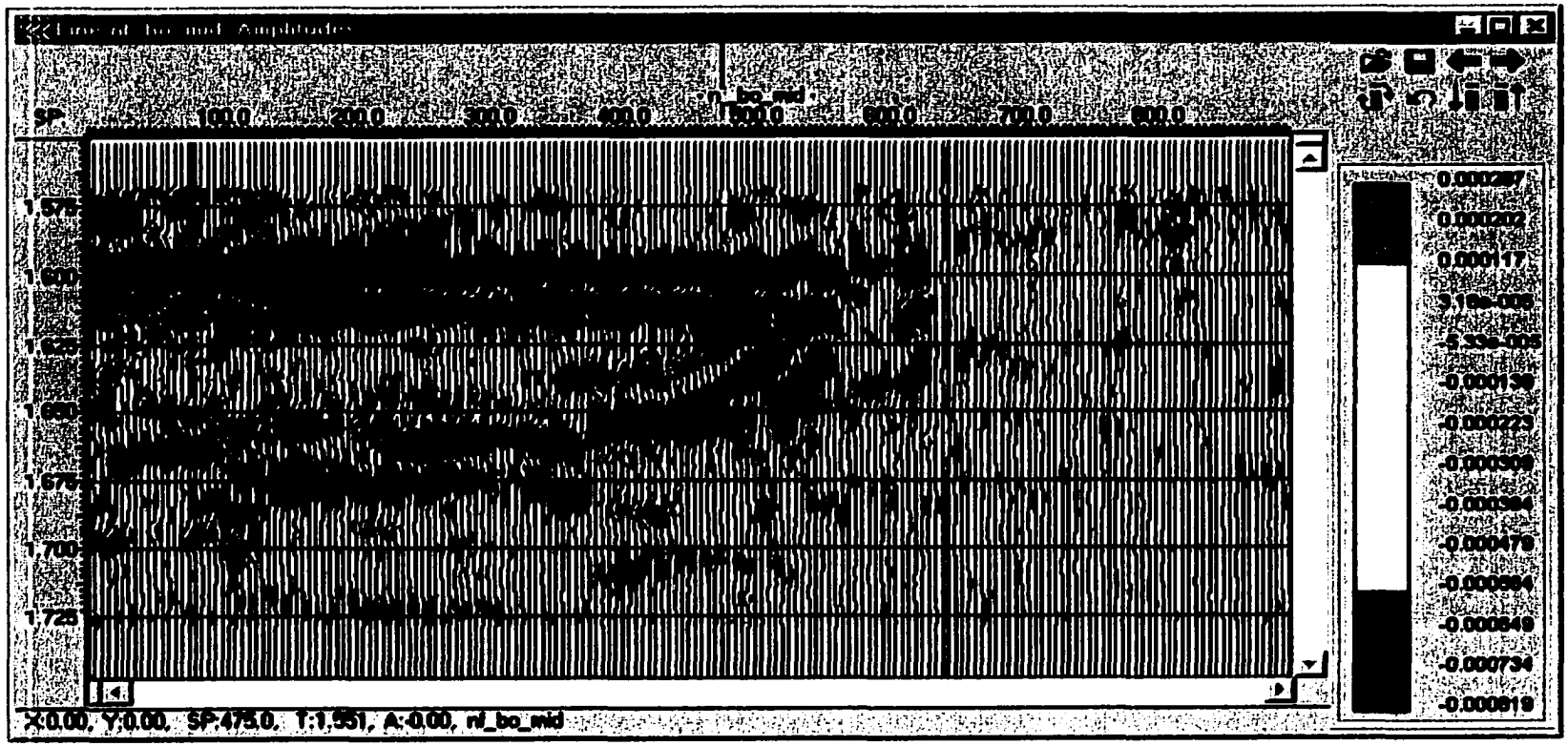


Figure 7-3. The AVO anomaly of neural network inversion for the 2D seismic data set

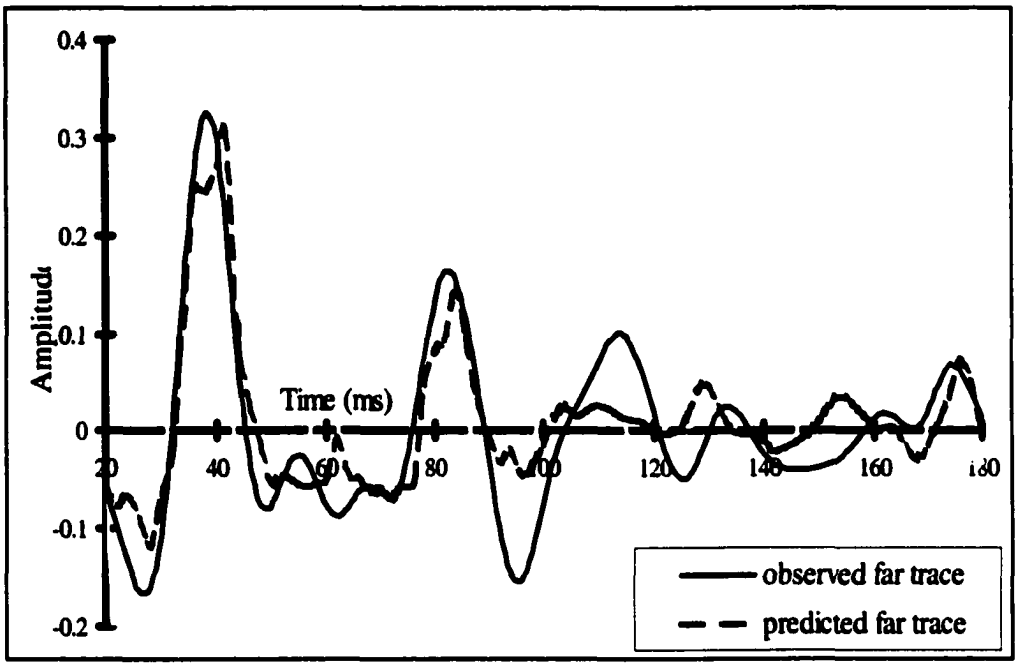


Figure 7-4. Training window AVO

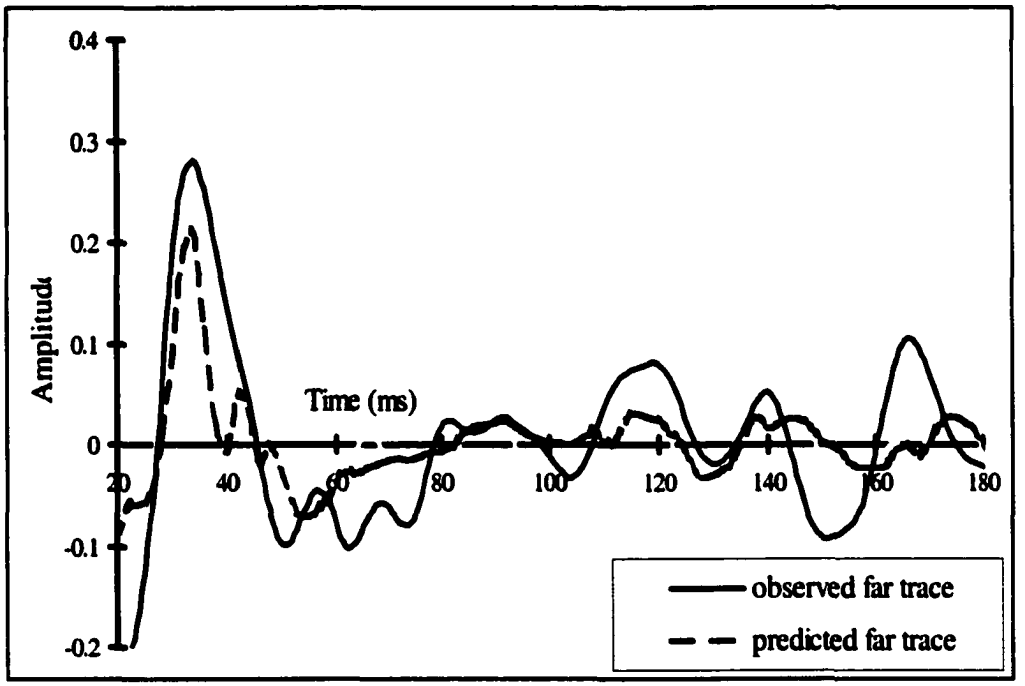


Figure 7-5. Brine window AVO

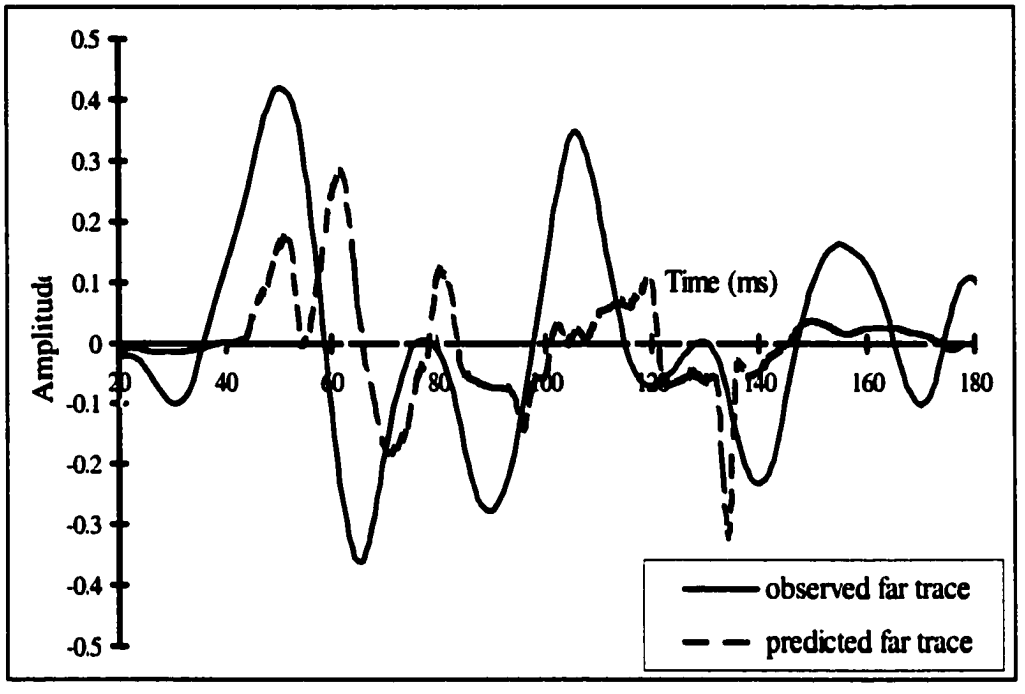


Figure 7-6. Gas window AVO

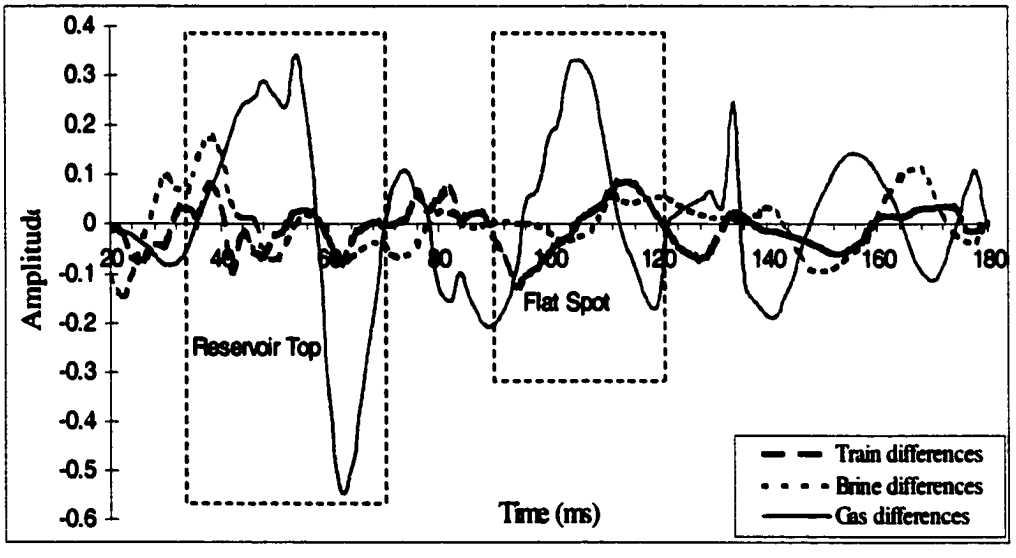


Figure 7-7. AVO anomaly

7.3 Case study 2

In this case there are 50 CDP gathers in the 2D line. The gathers had been partially stacked into near traces and far traces. In order to decrease the influence of geological structure and depth, the stacked data, both near traces and far traces were flattened along the reservoir top. Figure 7-8 shows the partially stacked near traces, and Figure 7-9 shows the partially stacked far traces. The time window selected is about 120 ms, including only the reservoir top. There is known to be a gas reservoir around CDP #16-50. 4 near traces (trace#1 ~ trace#4) in the brine window were used as input and 4 far traces at same location were used as the desired output to train the neural network. The trained neural network was used to calculate the far traces of the whole CDP gathers. The differences between the observed far traces and predicted far traces were next calculated. These anomalies are shown in the Figure 7-10. From this figure we can see that the anomalies in the gas window (CDP #16-50) are larger than those in the brine window. This AVO anomaly is verified as a hydrocarbon indicator.

Figure 7-11 shows the AVO anomalies after phase unwrapping to suppress the NMO error. The area of the gas sand indicated by the AVO anomaly is smaller than that Figure 7-10 showed. With phase unwrapping we can erase the influence of the NMO error and be much more confident to use the AVO anomaly as a hydrocarbon indicator. However, the AVO behavior should include both amplitude change and the

phase change. When we corrected the spectra change caused by NMO error, we also sacrificed the spectra change caused by hydrocarbon. Accordingly, the best idea is to balance the AVO anomaly before the phase unwrapping and after the phase unwrapping, use the AVO anomaly before the phase unwrapping to indicate the largest area of the hydrocarbon, and use the AVO anomaly after the phase unwrapping to indicate the most possible location of the hydrocarbon.

Three selected traces from the training window, the brine window, and the gas window respectively are used to show the AVO behavior more clearly. Figure 7-12 shows the observed far trace, neural network predicted far trace, and phase unwrapped far trace of one CDP in the training window. The predicted trace matches the observed trace very well in the training window. Figure 13 shows the same traces in the brine window. Both the predicted far trace and phase unwrapped far trace match the observed far trace very well, but the phase unwrapped far trace matches better. Figure 14 show the same three traces in the gas window. Comparing with that in the brine window, the predicted far trace does not match the observed far trace well. The phase unwrapped far trace matches the observed far trace better that the predicted trace. The amplitude of the phase unwrapped far trace at the trough location is much smaller than that of observed far trace, but the locations of the trough are the same. Figures 7-12 through 7-14 show that the method of phase unwrapping can modify the predicted far trace, and can correct the NMO error if there is some. Figure 7-15 shows the AVO anomaly without the phase unwrapping for the CDP gathers in the training window,

brine window and gas window respectively. The AVO anomaly for the gas window is much larger that in the brine window. Figure 7-16 shows the AVO anomaly with the phase unwrapping for the same three CDP gathers. Comparing the AVO anomaly in Figure 7-15, the anomalies in this figure are smaller. However, the AVO anomaly in the gas window is still very outstanding comparing with that in the brine window. So the AVO behaviors with and without phase unwrapping are the same in the most possible hydrocarbon area.

In conclusion, the AVO anomaly calculated by neural networks can be a hydrocarbon indicator. The phase unwrapping can suppress the NMO error. The AVO anomalies with and without phase unwrapping should be used comprehensively to interpret the likelihood of hydrocarbons.

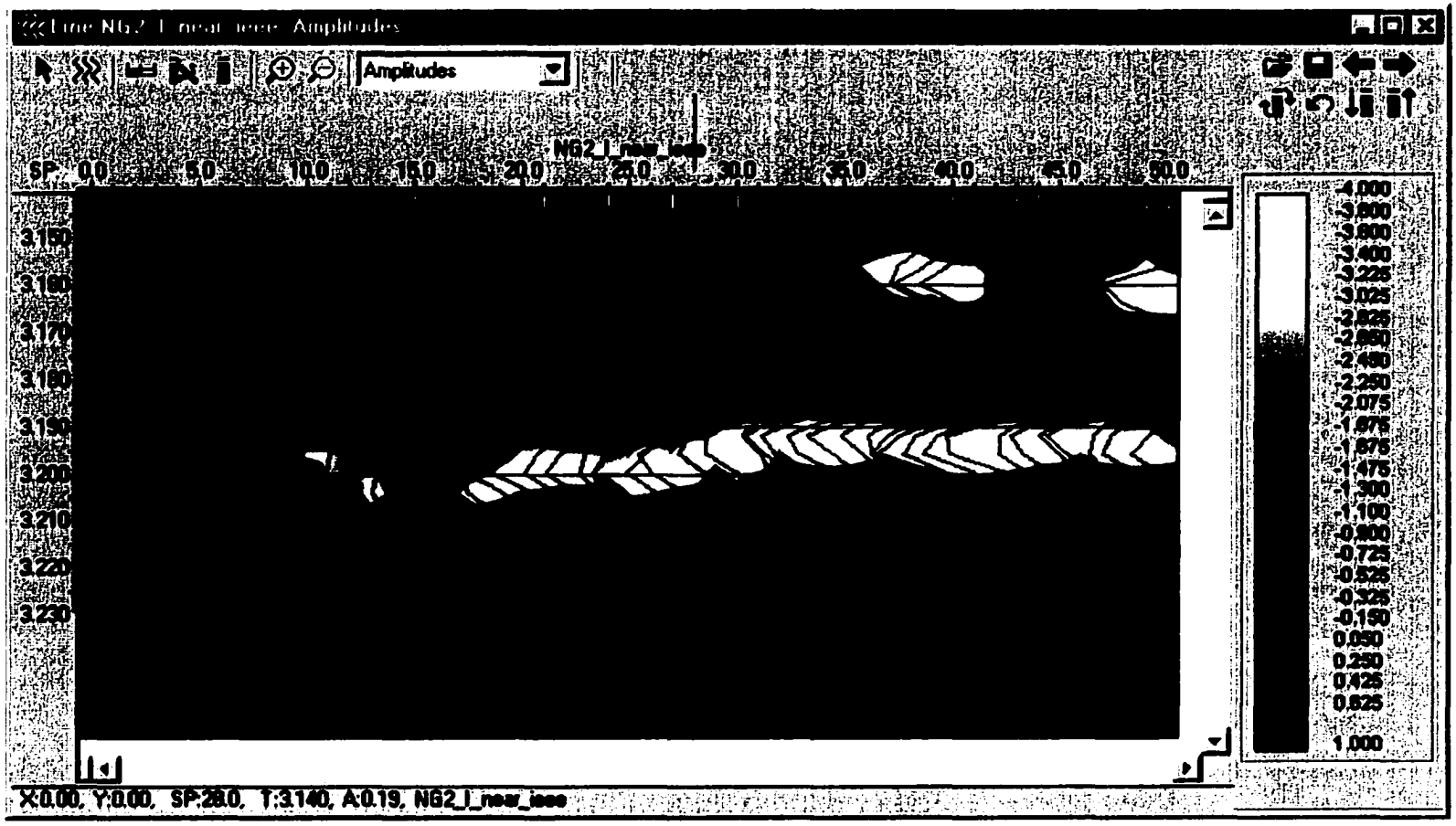


Figure 7-8. Partially stacked near traces

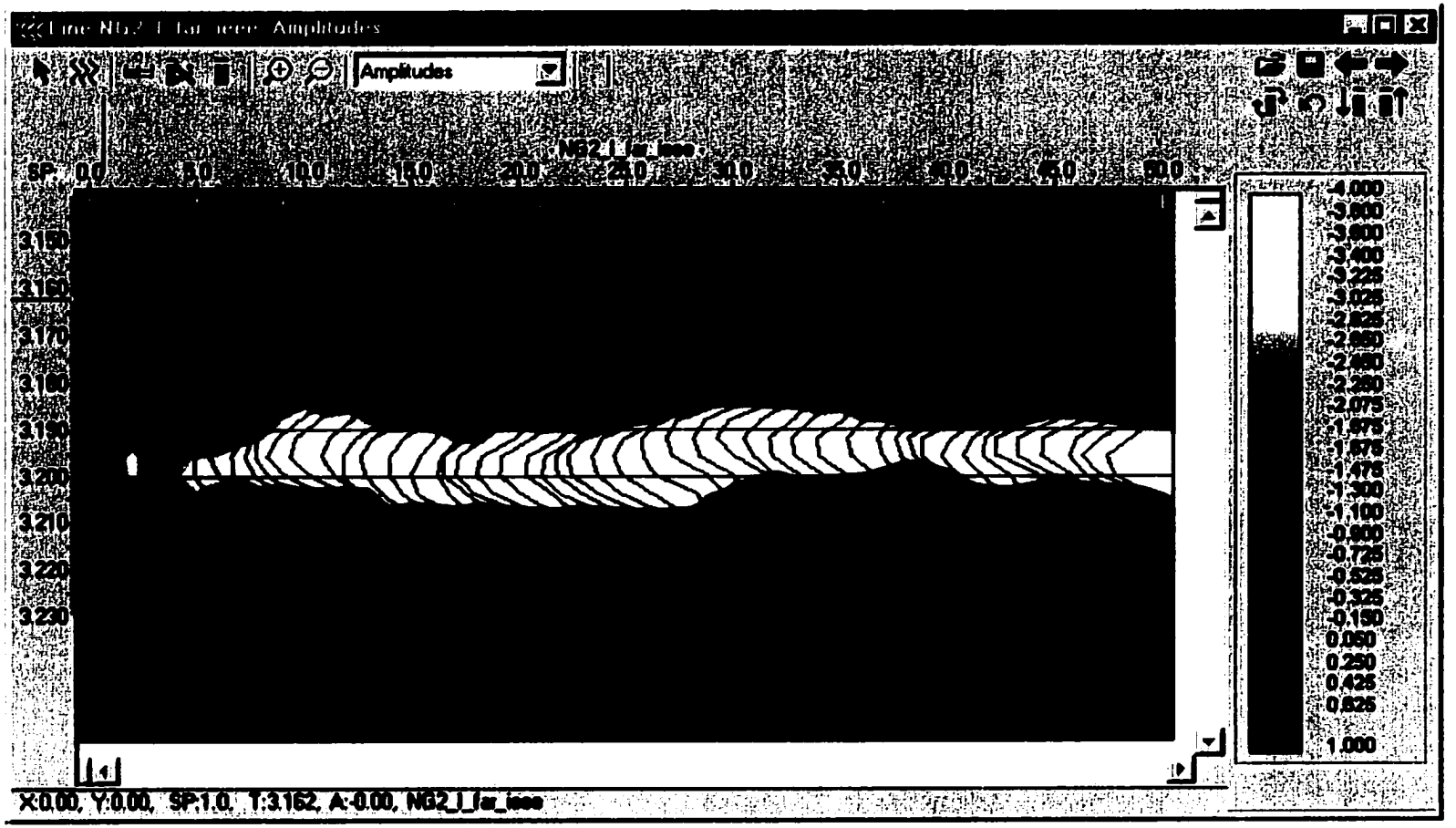


Figure 7-9. Partially stacked far traces

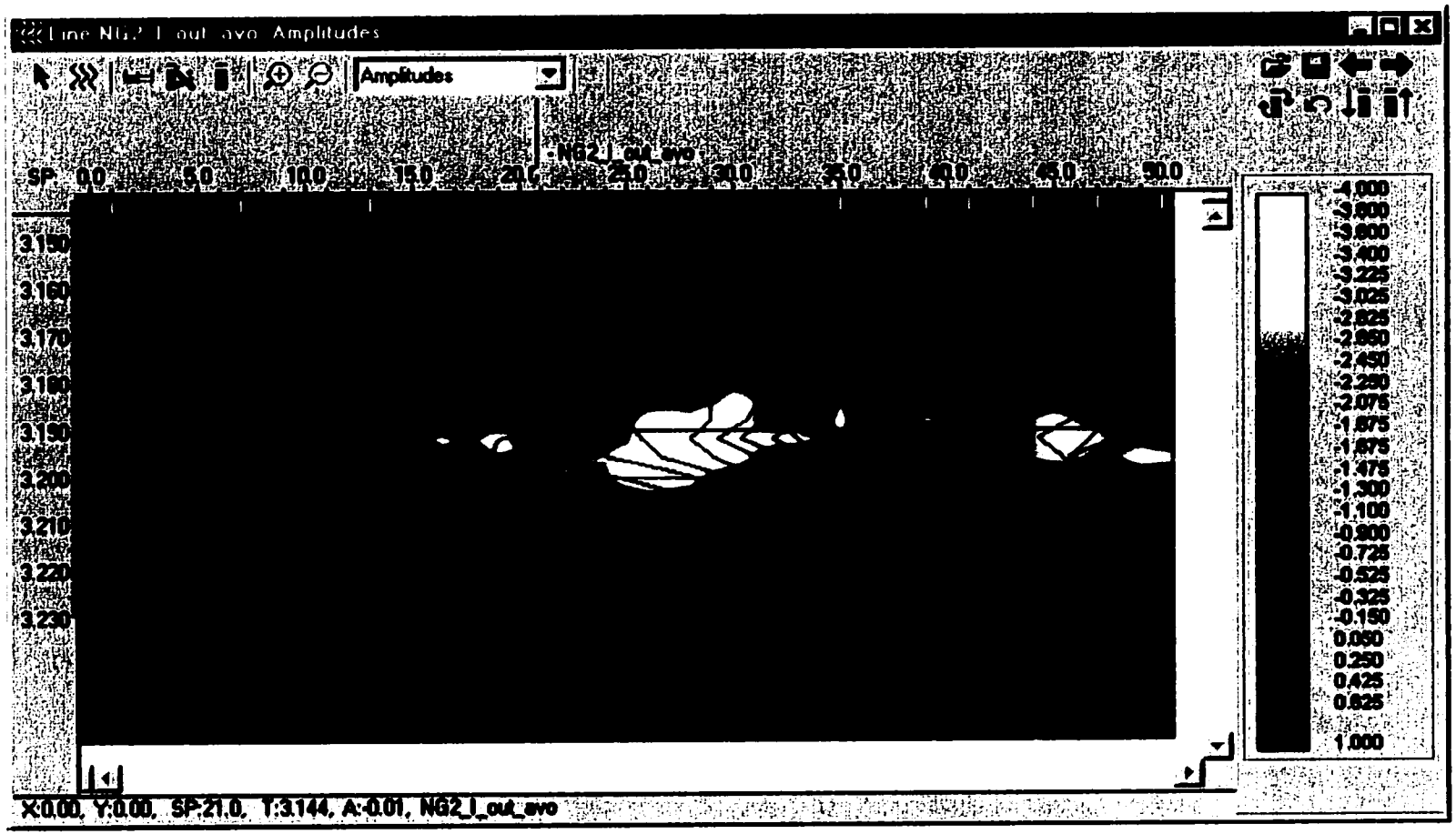


Figure 7-10. AVO anomaly without phase unwrapping

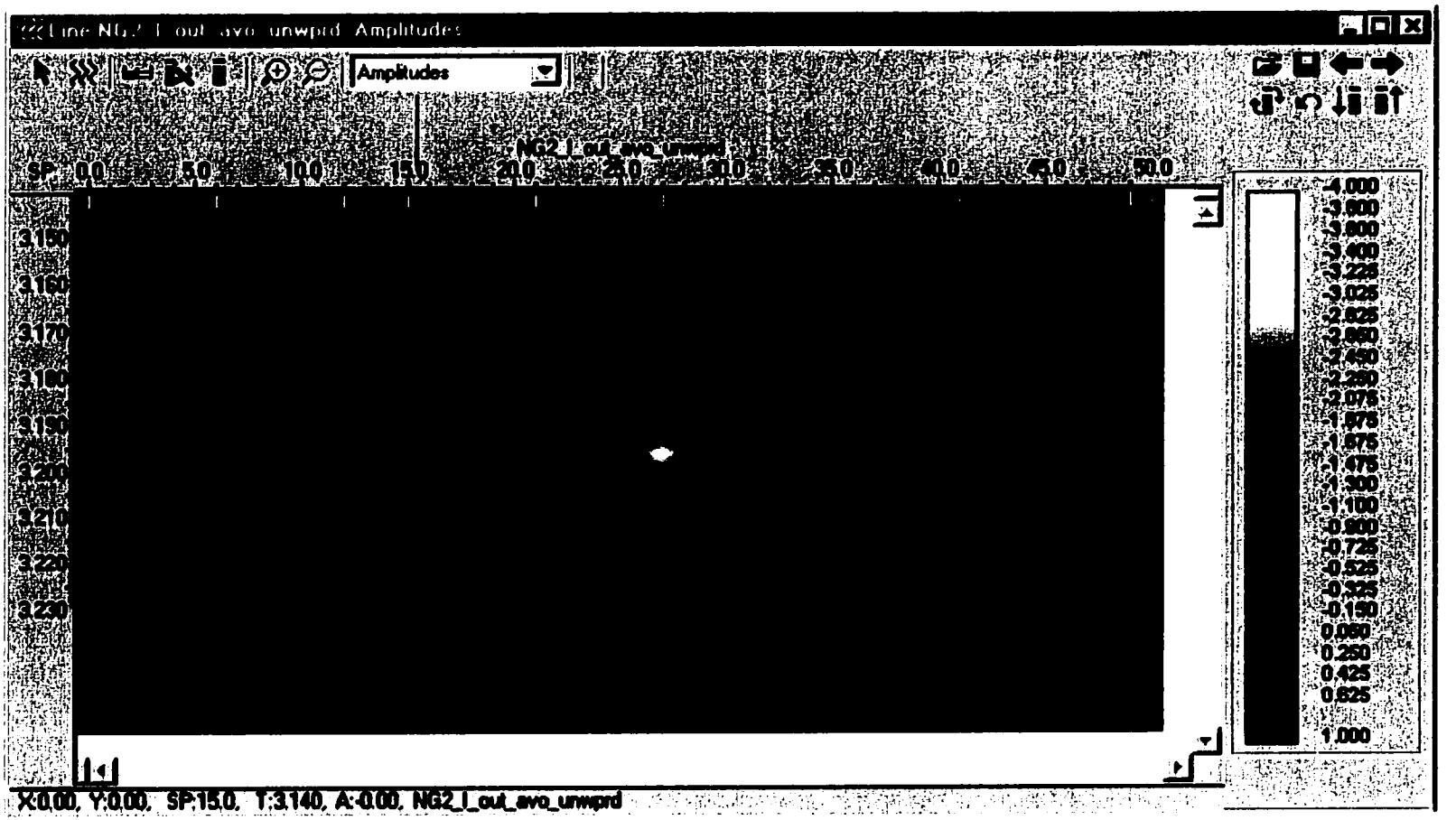


Figure 7-11. AVO anomaly with phase unwrapping

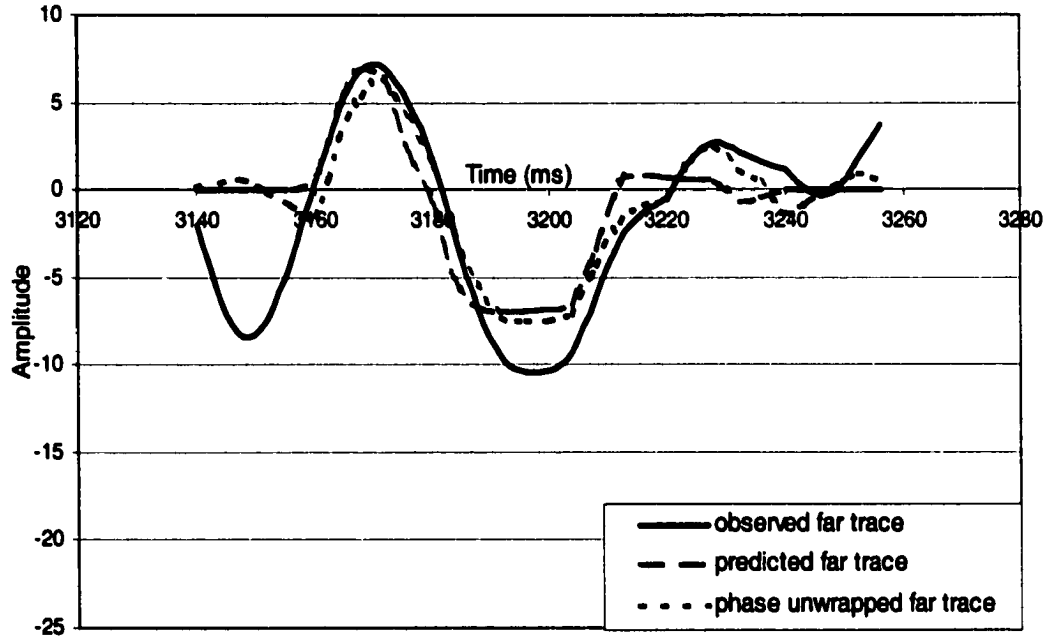


Figure 7-12. AVO inversion in the training window

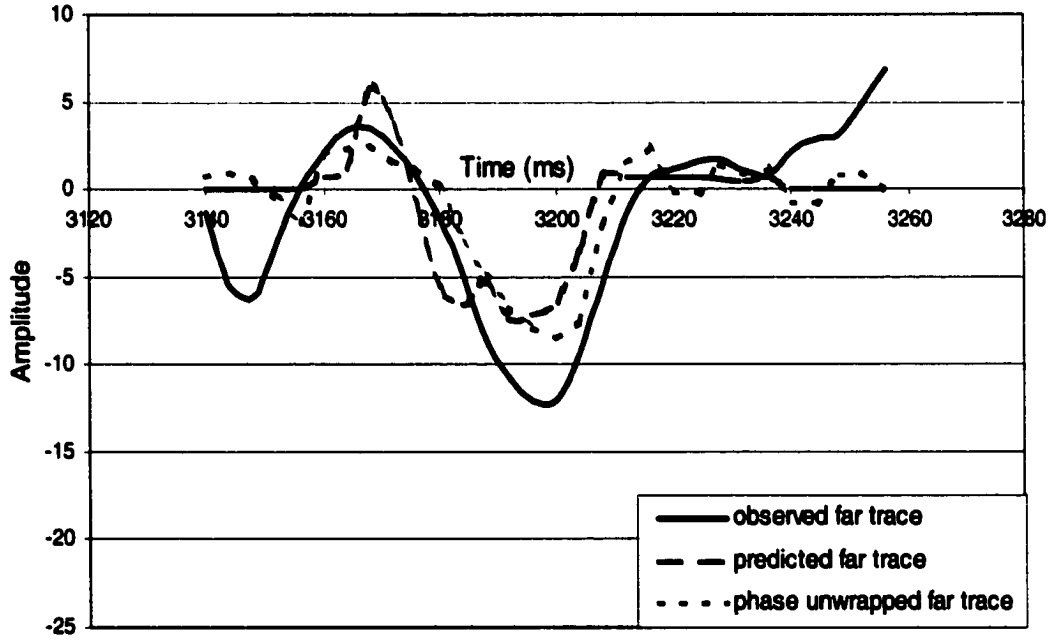


Figure 7-13. AVO inversion in the brine window

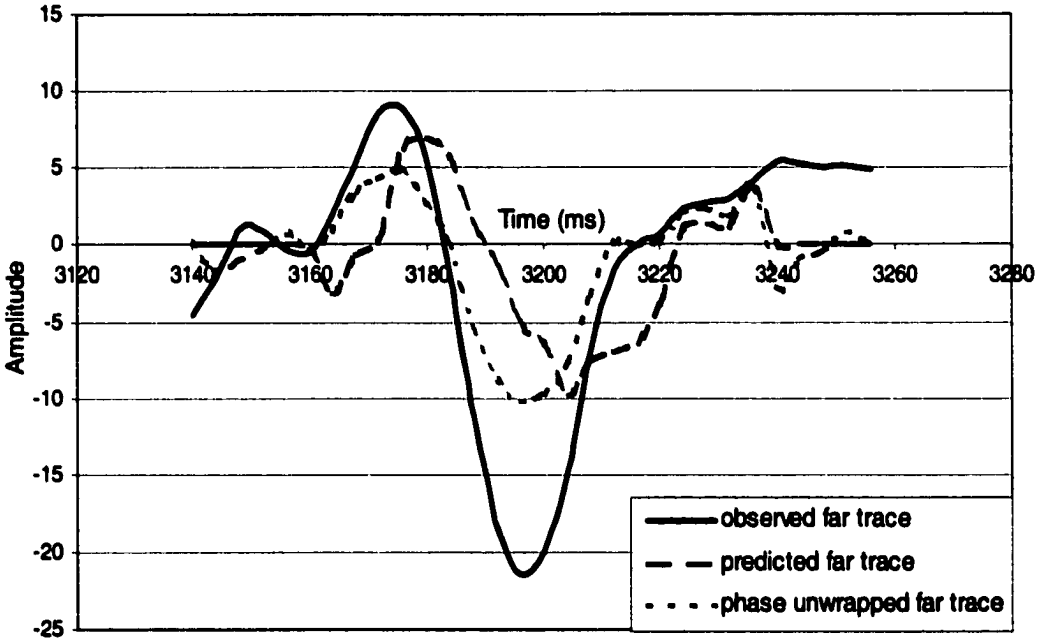


Figure 7-14. AVO inversion in the gas window

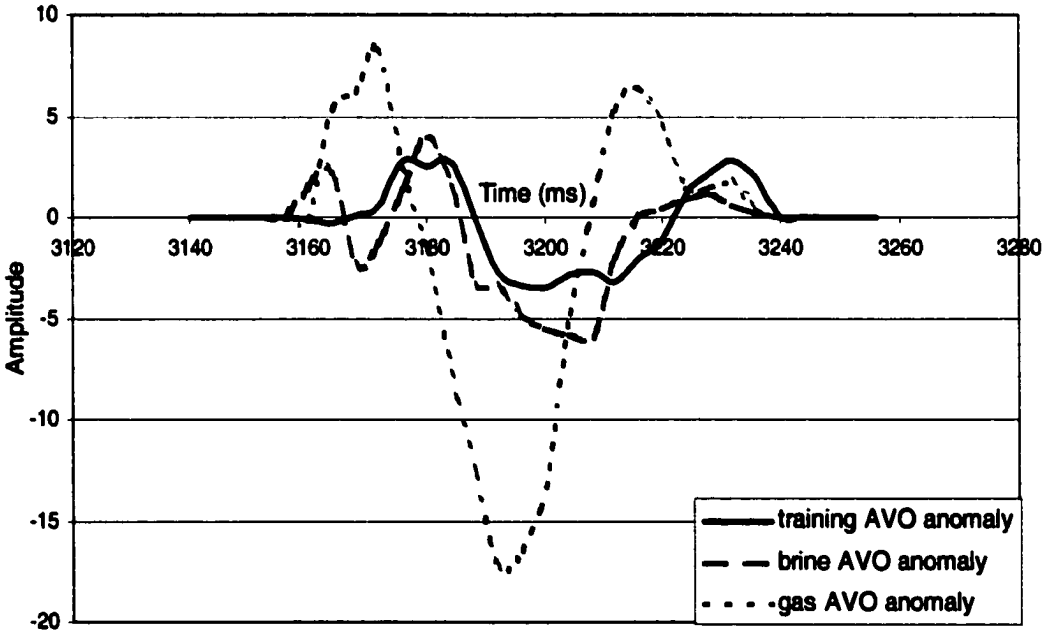


Figure 7-15. AVO anomaly without phase unwrapping

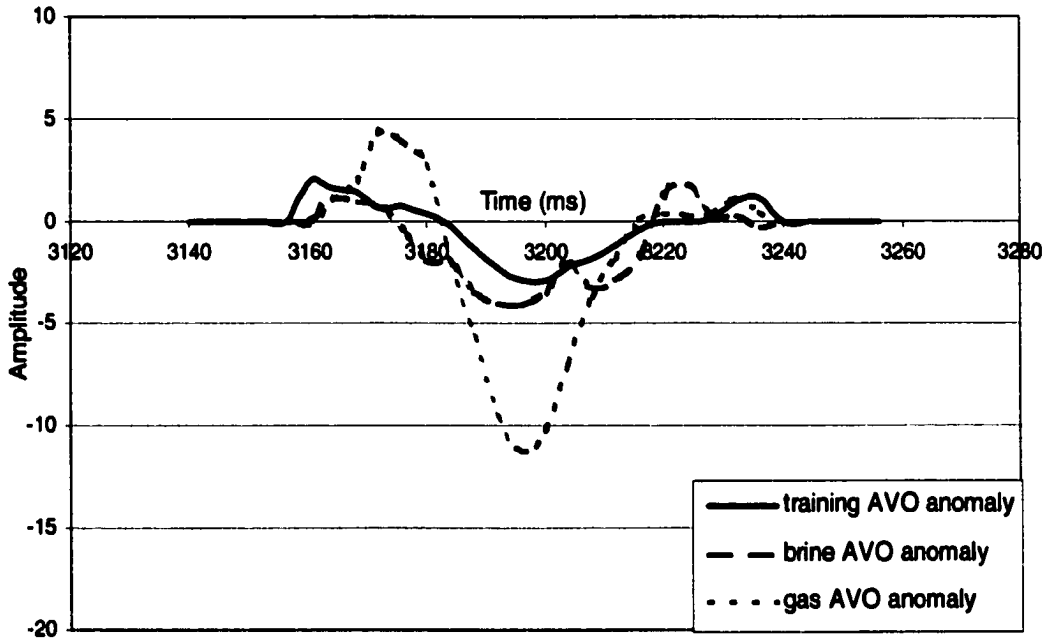


Figure 7-16. AVO anomaly with phase unwrapping

CHAPTER 8. CONCLUSIONS AND DISCUSSION

8.1 Conclusions

1. Using artificial neural networks, we can invert seismic data into petrophysical attributes, such as electric resistivity or porosity. This method requires the seismic data and relative petrophysical attribute data to train the neural network first. It does not require the forward model between the seismic data and petrophysical attributes. The neural network will find this forward relationship by the training.

2. The petrophysical data usually come from the well log. Since the well logs have wider frequency components than the seismic data, we should provide the neural network with low frequency interpolation data.

3. In the training of a neural network for seismic inversion, the learning pair should include one wavelength of seismic data as input and one correlated point in the well log as desired output. In this way, we can raise the resolution of inversion from tuning thickness to the sampling rate of the seismic data.

4. Usually, seismic data and well log are in different domains. Seismic data are in the time domain, and well log data are in the depth domain. Before the inversion, we should unify them into the same domain. I recommend converting the well log

data from depth domain to time domain. The depth-time conversion is very important for the accuracy of the inversion.

5. The depth-time converted well log data should be resampled into the same sampling rate as the seismic data. In order to keep as most information as possible, before the re-sampling, the well log data should be smoothed.

6. Comparing with conventional inversion (sparse spike, blocky), inversion using artificial neural networks can exhibit much higher apparent resolution and can be more accurate. Unlike conventional seismic inversion that can only invert seismic data into impedance or velocity, artificial neural networks can invert the seismic data into any petrophysical attribute that has a relationship with the seismic data without requiring a known forward physical model.

7. In order to avoid over-training, the training samples should be divided into two groups, the training group and validation group. The training process should be controlled by observing the error of both training data and validation data. When the validation error begins to increase, we should stop the training of the neural network.

8. With 2D seismic data, when selecting seismic traces near the well for training, we should pick the seismic traces in both sides of the well. The number of traces I

recommend is 5~10. With the 3D seismic data, when selecting seismic traces near the well for training, we should avoid only picking the seismic trace in the inline and cross line that cross at the point of the well location. The near well seismic traces should be selected in a circular area. The center of the round is the location of the well. The radius of the circular area should be large enough to include 5~10 traces in the area. In this way, we can guarantee that the training data set include the seismic traces in all the directions around the well. Otherwise, the seismic data will be improperly inverted in some direction.

9. The neural network can be trained to ignore real random noise, but will be influenced by regular noises. Sometimes, adding random noises can modify the result of the inversion.

10. The difference between the observed far trace and artificial neural network predicted far trace can be used as an AVO anomaly detector. The neural network is trained previously in the brine window. The AVO anomaly can be a hydrocarbon indicator. Both synthetic data and real data were used to test this method. The method was found to correctly identify AVO anomalies for both data sets studied.

11. AVO anomaly calculated by neural network is insensitive to constant NMO error. The constant NMO error means the NMO errors in the brine window and gas window are the same.

12. With AVO inversion using ANN, one way to suppress the variable NMO error is cross-correlation between the observed far trace and ANN predicted far trace. Because the cross-correlation is performed between predicted and observed traces, it is not confounded by differential tuning, stretch, frequency content, interference, etc. that would occur if an attempt were made to correct the NMO by cross-correlating with the near trace for example.

13. Another method to suppress the variable NMO error for the AVO inversion by ANN is phase unwrapping. With this method, we assume that the phase change is the only factor that will cause the NMO error. So, the phase spectrum of the observed far trace and amplitude spectrum of the predicted far trace are combined into a new far trace. We think the NMO error should be erased in the new far trace. This method was tested on both synthetic data and real data. The result is very good. However, there is disadvantage of this method. The AVO behavior may include the phase change. After phase unwrapping, the phase spectra between the predicted far trace and observed far trace are enforced to be the same. So, it will ignore the AVO anomaly between the predicted far trace and observed far trace caused by phase change. For this reason, when we interpret the ANN inverted AVO anomaly, the AVO anomaly calculated before the phase unwrapping should be used with the AVO anomaly calculated after the phase unwrapping to get a more reasonable result.

8.2 Discussion and future work

1. Low frequency of the petrophysical attribute is a kind of input for training the neural network. The reason we provide this data to the neural network is that the seismic data does not have the low frequency component. To prepare this data, I do the interpolation between wells and filter the interpolation by a low pass filter. The high cut of the filter should be between 5 and 15 Hz, and should not be higher than 15 Hz. The low frequency data is just a very coarse back ground for the training of the neural network. The neural network will handle the detail information between the seismic and the well log data. When the low frequency data contain higher frequency components, the higher frequency component will dominate the detail information.
2. For the 2D seismic inversion by ANN, it is relatively simple to select learning pairs and train the neural network. For the 3D seismic inversion, the situation may be more complex. In my research, I find that faults will significantly influence the inversion. So, if the faults are very developed in the 3D area, it is better to divide the whole 3D area into some smaller areas according to the main faults and use several neural networks to handle the areas; every network corresponds to one area. After inversion by the multi-neural networks, combine the results back to the 3D area. In this way, the accuracy of 3D inversion will be improved.

3. For seismic inversion using neural networks, the selection of learning pair is very important. The learning pair is composed of input data and desired output data. Usually, the desired output is one point of well log data. In my research, I used one wavelength of post-stacked seismic data and low frequency data as the input. It is also possible to add spatial and temporal constraints in the input data. Actually, we can add any useful information into the input data set if we think it will help to modify the inversion result. I have added the convolutional model into the input data. This process is show in the Figure 8-1. The result of inversion after adding the convolutional model is more stable, however, loses some detail information and decreases the resolution. I give this example to explain that we should optimize the input data. The input data set I used may not be the best combination. There is some future work needed into establishing procedures to choose the input data. In the future, we may wish to consider extracting some attribute from the seismic data, and putting them into the input data set. This is common practice today, but it remains to be seen if additional information is obtained by extracting various attributes from the same seismic trace. These attributes may not only come from the post-stack seismic data, but also form the pre-stack seismic data, in which case additional information would be provided to the neural network.

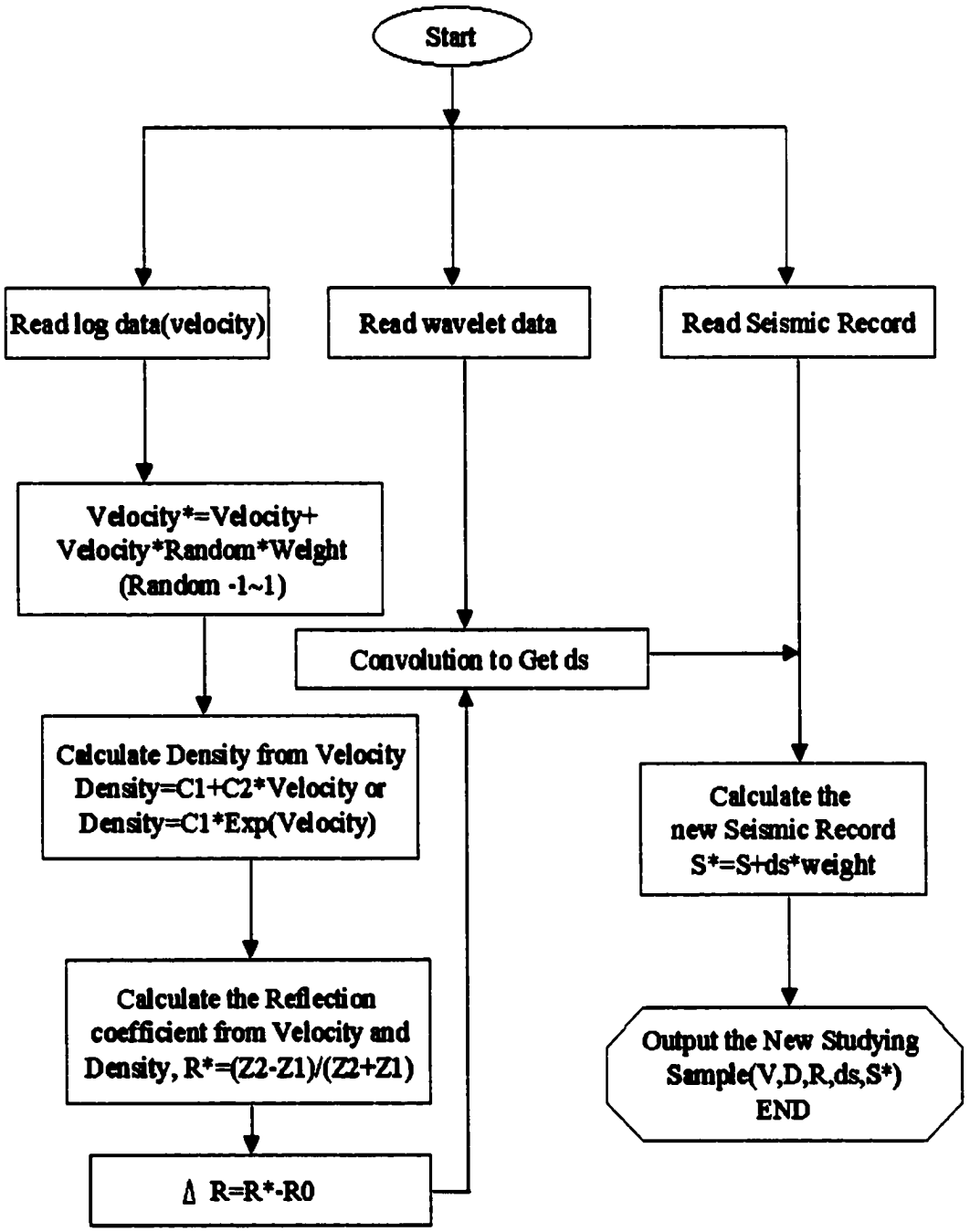


Figure 8-1. Flow chart of adding convolution model

4. In order to suppress the NMO error in the AVO inversion by ANN, I tried another method, which is AVO inversion in the frequency domain. Figure 8-2 is the flow chart of this method. Near trace and far trace stacks were converted to the frequency domain by Fourier transformation. Two neural networks will handle the real part and imaginary part respectively. The predicted real part and imaginary part will convert back to the trace in the time domain by reversed Fourier transformation. In order to let the neural network adapt the NMO error, several CDP gathers with different NMO errors were used in training the neural network. However, the result showed that the neural network is over trained. I tried several training schemes to avoid the over training. Those schemes include increasing exit error, decreasing training time, and adding some random noise to make more learning pairs. However, the situation of over training was not modified. In the time domain we can use one wavelength of seismic data correlating one point of desired output data and slide the sampling window to make a learning pair. In the frequency domain, we cannot use slicing windows to make learning pairs, so the number of learning pairs is much less than those of time domain. This may be the reason for the over training. I did not find a good way to select the learning pairs to avoid over-training. But I think if we make a much more complex synthetic model, the over-training may be avoided.

5. For the AVO inversion by ANN, in addition to using near trace as input and far trace as output, I tried the reverse. In that way, far traces were used as input and near traces were used as desired output to train the neural network in the brine window. Then, the trained neural network was used to predict the near traces by inputting the far traces. After that, the AVO anomalies were calculated by subtracting the predicted near traces from the observed near trace. This method worked well in the synthetic model. However, in the real data set, it did not work well. Figure 8-3 shows the AVO anomalies calculated in this way for the 2D seismic data set of case#1 in the chapter 7. The result is very noisy. Comparing with that in Figure 7-3, it very difficult to tell the prospect of the gas reservoir in the Figure 8-3. The reason for the failure is that the partially stacked near traces are not so stable as the partially stacked far traces.
6. For the AVO inversion by ANN, in addition to using the CDP gathers in the brine window to train the neural network, I used the CDP gathers in the gas window to train the neural network. Using this training scheme, I wanted to let the neural network learn the relationship between the near traces and far traces in the gas area. The trained neural network was used to predict the far traces by inputting the near traces. After that, the AVO anomalies were calculated by subtracting the predicted far traces from the observed far traces. I expect that these anomalies should be small in the gas window, and larger in the brine window. This method worked well in the synthetic model. However, it did not show such behavior in the real data set. Figure 8-4 shows the AVO anomalies calculated in this way for the

2D seismic data set of case#1 in the chapter 7. From the figure we can see that the AVO anomalies are not small in the gas window and not large in the brine window. The reason for the failure is that the seismic responses in the gas window are not very stable, and varies due to many factors such as porosity change and water saturation change.

In conclusion, the use of neural networks is a promising seismic inversion method that does not rely on a known forward physical model. This dissertation has demonstrated such promise on various real and synthetic, post-stack and pre-stack inversion problems. As much research is needed into how to set up the neural networks for specific problems, as is needed in improving the inversion algorithms themselves. Existing algorithms are adequate for many potentially profitable applications if appropriate methodologies can be developed and employed.

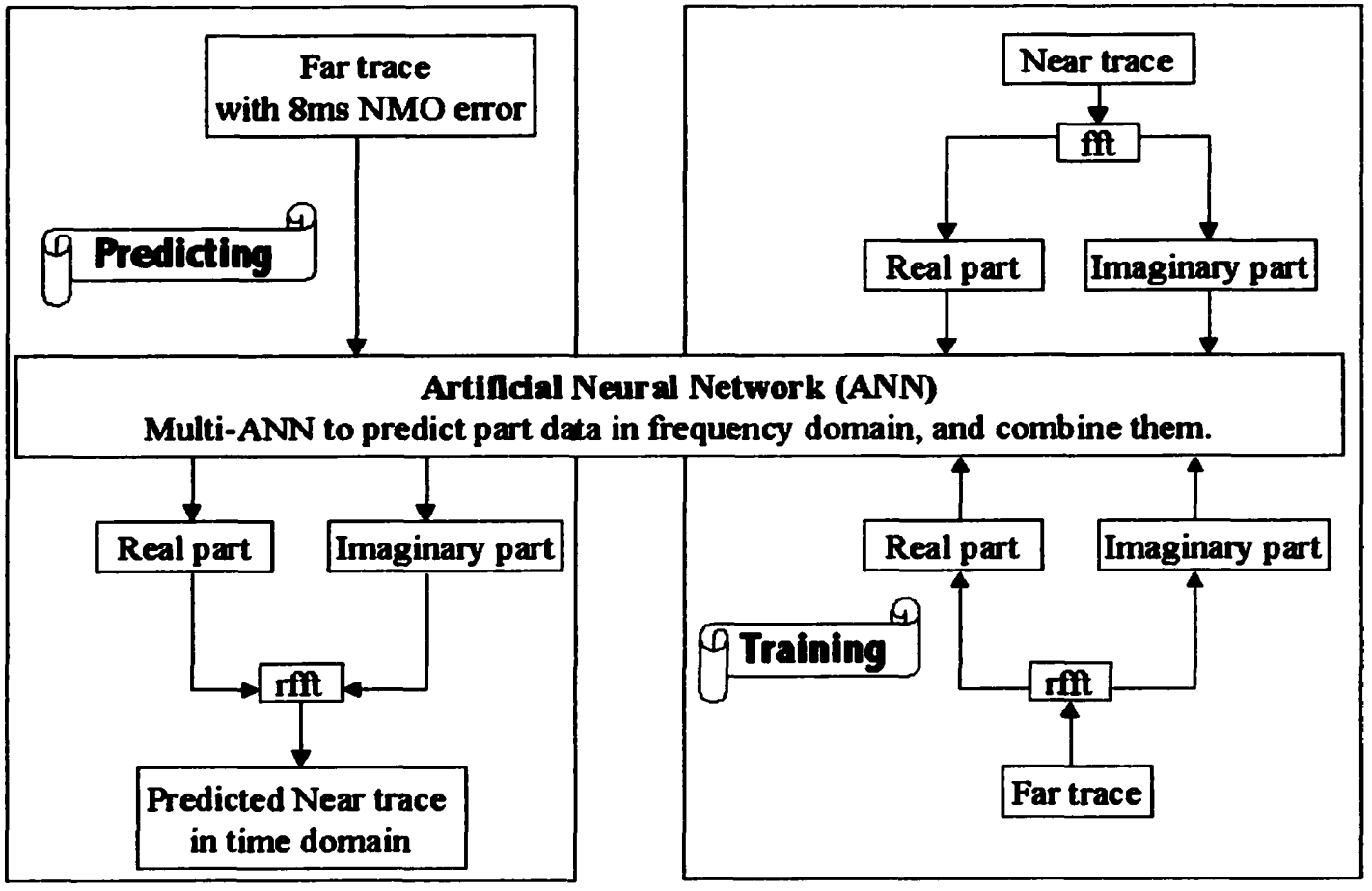


Figure 8-2. Flow Chart of AVO inversion in frequency domain

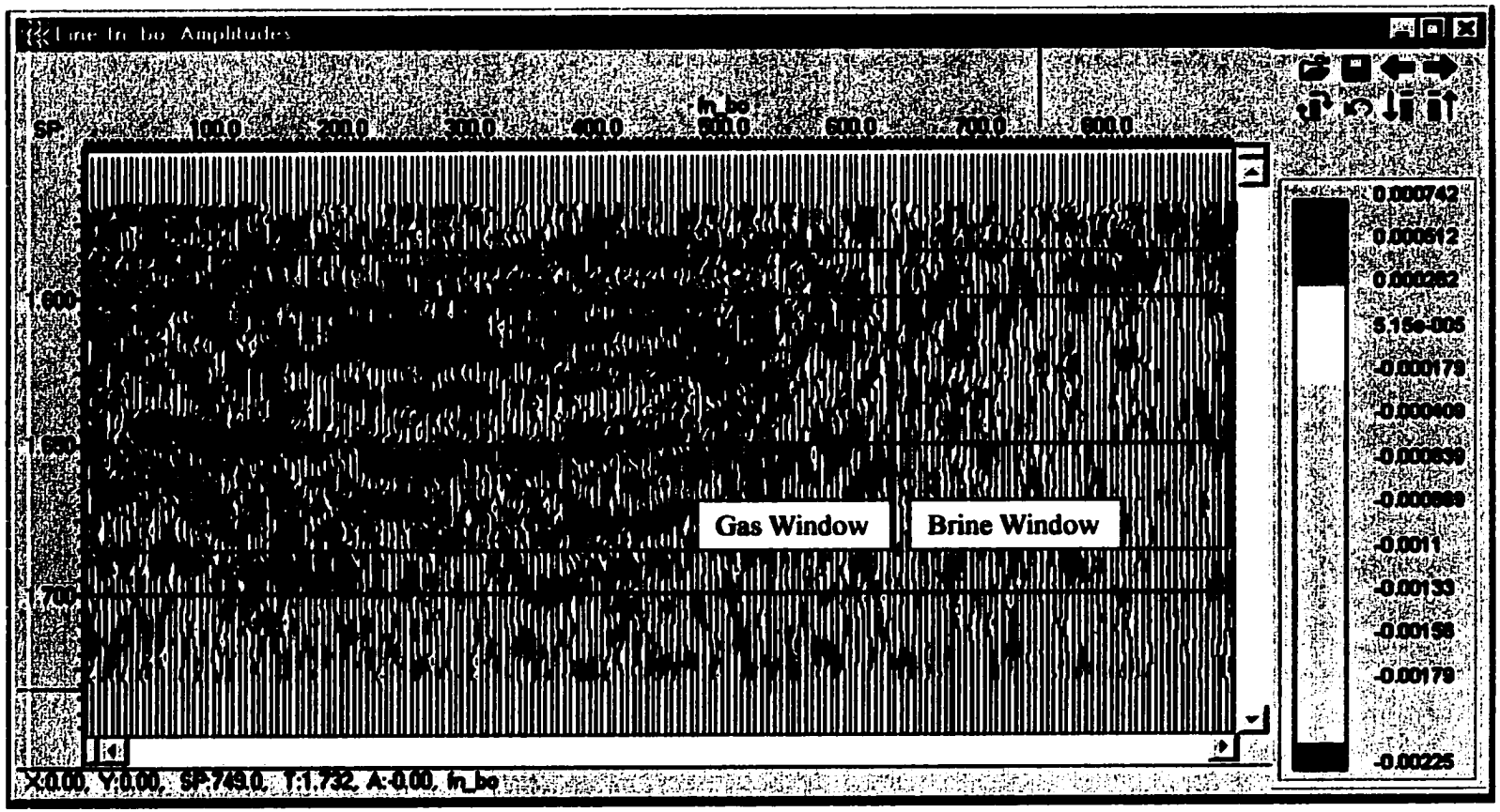


Figure 8-3. AVO anomaly, far traces were used as input.

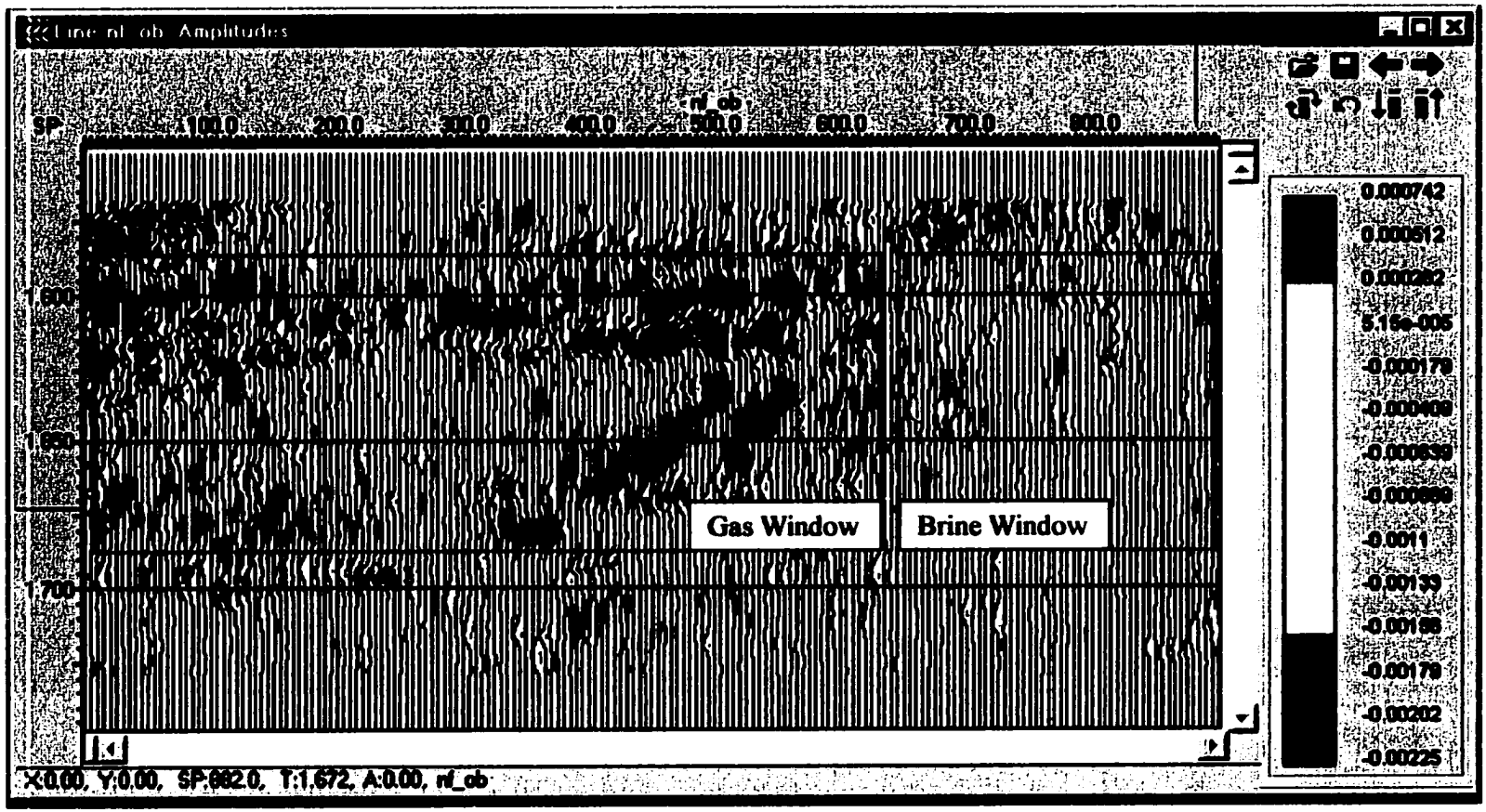


Figure 8-4. AVO anomaly, CDP gathers in the gas window were used in the training.

REFERENCE

Baan, M.V.D. and Jutten, C., 2000, Neural networks in geophysical applications: Geophysics, Vol. 65, No. 4, P. 1032-1047

Boadu, F.K., 1997, Rock properties and seismic attenuation: neural network analysis: Pure and Applied Geophysics, 149, P. 507-524.

Boadu, F.K., 1998, Inversion of fracture density from field seismic velocities using artificial neural networks: Geophysics, Vol. 63, No. 2, P. 534-545.

Buffenmyer, v., Poulton, M., and Johnson, r., 1999, Neural networks approach to seismic crew noise identification in marine surveys: 1999 SEG conference expanded abstract.

Calderon-Macias, C., Sen, M.K., Stoffa, P.L., 1997, Hopfield neural networks, and mean field annealing for seismic deconvolution and multiple attenuation: Geophysics, Vol. 62, No. 3, P. 992-1002.

Calderon-Macias, C., 1998, Automatic NMO correction and velocity estimation by a feed forward neural network: Geophysics, Vol. 63, No. 5, P. 1696-1707.

Castagna, J. P., 1993, AVO Analysis – Tutorial and Review: Offset-dependent reflectivity – theory and practice of AVO analysis.

Castagna, J. P., Swan, H.W. and Foster, D. J., 1998, Framework for AVO gradient and intercept interpretation: Geophysics, Vol. 63, P. 948-956.

Chakravarthy, S.V., Chunduru, R.K., Mezzatesta., A.G., and Fanini O., 1999, Detection of layer boundaries from array induction tool responses using neural networks: 1999 SEG conference expanded abstract.

Davies, P., and Warner, M., 1999, Non-linear cross-equation of seismic surveys acquired with different bandwidths: 1999 SEG conference expanded abstract.

Hampson, D.P., Schuelke, J.S., and Quirein, J.A., 2001, Use of multiattribute transforms to predict log properties from seismic data: Geophysics, Vol. 66, No. 1, P220-236

Hampson, D., Russell, B., 1999, Manual of Hampson Russell package, strata.

Hornik, K., Multilayer feedforward networks are universal approximators: Neural Networks, Vol. 2, No. 5 (1989), P 359-366.

Liu, Z.P. , Liu, J.Q., Castagna, J.P., and Pan, C.H. 1998, An iterative inversion of seismic data constrained by wells using artificial neural networks: Offshore Technology Conference.

Liu, Z.P. and Liu, J.Q., 1998, Seismic-controlled nonlinear extrapolation of well parameters using neural networks: Geophysics, Vol. 63, No. 6, P. 2035-2041

Meldahl, P., Heggland, R., Bril, B., and Groot, P., Identifying faults and gas chimneys using multiattributes and neural networks: The leading EDGE, Vol. 20, No. 5, P. 474-482

Pandya, S.A., Macy, B.R., Pattern recongnition with neural networks in C++, 1996, P87~96

Russell, B., Hampson, D., Schuelke, J., Quirein, J., Multiattribute seismic analysis: 1997, Leading Edge, Vol. 16, No 10, P. 1439-1443

Russell, B., Introduction to seismic inversion methods, 1988

Sun, Q., Castagna, J., Liu, Z. P., AVO inversion by artificial neural networks (ANN), SEG 2000, expanded abstract

Sun, Q., Eissa, M., Castagna, J., Cersosimo, D., Sun, S. J., Decker, C., Porosity from Artificial Neural Network Inversion for Bermejo Field, Ecuador, SEG 2001, expanded abstract

Zhang, L., Poulton, M., Zhang, Z.Y., Mezzatesta, A., and Chakravarthy, S., 1999, Fast forward modeling simulation of resistivity well logs using neural networks: 1999 SEG conference expanded abstract.

Zurada J.M., Introduction to Artificial Neural Systems, 1992

Appendix A

SEISMIC INVERSION BY ARTIFICIAL NEURAL NETWORKS

BY ZHENGPING LIU and QIANG SUN

ABSTRACT

In this report, we address a new seismic inversion method by Artificial Neural Networks (ANN). Rather than using a theoretical model, we let the BP neural network learn the relationships between the seismic input and the desired output, (the well logs), and perform the inversion. We tested the method by using several seismic synthetic models which include a wedge model, a wedge model with noise and a wedge model with multiple, and then compared the inversion results with conventional methods, blocky and sparse spike. The results show that ANN worked better in many fields. Finally, some real data are used to test the inversion by ANN, and a technology which creates new learning pairs is introduced.

INTRODUCTION

Conventional inversion strategies look for an optimal geophysical model, which can invert the seismic traces to well data with minimum residual. These methods can produce pseudo-acoustic impedance logs from seismic records. These methods include the bandlimited, blocky, sparse spike, etc. There exist two limitations for the conventional strategies. First, we must use a proper forward model. Second, the inversion can not directly invert for well data other than impedance, velocity and density.

Artificial Neural Networks (ANN) are an attractive alternative to conventional inversion methods due to their ability to "learn" and "estimate". Since ANN can learn the relationship between well data and seismic records, a correct forward modeling algorithm is not required. Also, as long as there is some kind of relationship between the well data and seismic records, we can perform the inversion. We need not know what the exact relationship is, because to find the relationship is the ANN's job. In this study, we let an ANN invert seismic data for the Gamma Ray (GR) log response. The result is pretty encouraging.

Among the ANN models, the back ward propagation (BP) model is one of most effective. It is a feed forward network trained by error back propagation. When it has one hidden layer, its approximated function is universal. In the learning stage of a BP neural network, the output error is propagated back ward, and the link weights of the network are modified with the propagation of error. When the total error is decreased to a given level, the network arrives at the relationship between the input and the desired output. This relationship is saved in weights, which are encoded in the network.

To test inversion by ANN, we used a seismic modeling package (GXII) to create some synthetic models, which include a wedge model, a wedge model with noise and a wedge model with multiple. We use both conventional methods and ANN to invert these models, and compare the results. The ANN worked better in certain aspects. We also use some real data to test the ANN, the results are also very encouraging.

STRATA (Hampson-Russell software) was used to perform the conventional inversions, which include bandlimited, sparse spike and blocky algorithms.

In order to stabilize inversion by ANN, we also introduce a method to create some new learning samples by a differential convolution model.

INVERSION OF SYNTHETIC MODELS

The inversion result of synthetic models and the comparison with conventional inversion are discussed here. The models are created with GXII.

1. Wedge without noise

This is a very sample model, but it is useful to test resolution. When a bed embedded in a medium of different impedance is about ¼ wavelength in thickness, the reflection from the top and the base of the bed interfere constructively and the amplitude increases. This "tuning" thickness is a practical resolution limit (the theoretical limit is 1/8 wavelength). In the wedge model (Fig. 1 and Fig. 2) we want to know if this effect in seismic record will also influence the resolution of inversion.

The maximum thickness of the wedge is 100 ms, the background velocity is 4000 m/s, and the wedge's velocity is 1000 m/s. For the ray tracing, we chose a 20 Hz Ricker wavelet (wavelength in the bed is 50m). So the #35 trace location is near a thickness of ¼ wavelength. (Fig. 3).

The results of conventional inversions, both of blocky (Fig. 4) and sparse spike (Fig 5), show that tuning influences the resolution seriously. The real thickness is not

be obtained when the trace number is greater than 33, so the practical resolvable thickness is greater than ¼ wavelength.

The inversion by ANN (Fig. 6) has better resolution. The real thickness is obtained down to trace #40. Even between traces #41 and #49 traces, there is decreasing thickness trend. In this inversion, only one trace was used in the training.

2. Noise Processing

In order to test the stability of ANN, we add 10% random noise in the ray tracing of the wedge model (Fig. 7). The results of blocky (Fig. 8) and sparse spike (Fig. 9) show that noise influenced the inversion, but not very seriously. However, the result of ANN (Fig. 10) is greatly degraded by noise. To get this result, we used 5 traces to train ANN. So, the ANN is apparently not stable in the presence of noise.

This result is misleading, when GXII adds noise, there is some regularity in the supposed random noise. The ANN may learn this regularity. To solve this problem, we use several different random systems to make the same level of noise, and use the traces with noise from different random systems in the training of the ANN. The result (Fig 11) is much better than the conventional inversion methods.

3. Multiple Processing

Multiples are supposed to be attenuated by stacking and other processing methods. However, multiples are often inadequately suppressed. Such multiples will influence inversions very seriously. A wedge synthetic with a multiple (Fig. 12) from its base

boundary is used to study the affects of multiples on ANN and conventional inversion methods.

The sparse spike (Fig. 13) and blocky (Fig. 14) inversions are affected by the multiple. However, the ANN (Fig 15) trained with three traces was not perturbed by the multiple. This is because that the ANN does not assume a primary only forward model, as do the conventional inversion methods tested.

INVERSION OF REAL DATA

In addition to the synthetic model, we also use real data to test the ANN inversion. The data we used are from Las Cavaos Oil Field, YPF. We select one line (Fig. 16) from its 3D seismic data set. There are three wells in the section. We use only the middle one, well 62, to train ANN. In the training, four traces near well 62 and a low frequency model are the input data; the velocity log of the well is the desired output. The result of inversion by ANN is shown as Fig 17. In the figure, the background wiggle is velocity by inversion. Near the well 62, the velocity log and the velocity by inversion match very well. This is expected for the training well. Near well 108 and well 143, the trend of velocity logging matched the velocity of inversion. So, the training results were applicable some distance from well 62. Comparing with that, Fig 17B is inversion by conventional method.

In training, well logs are the desired output and the seismic record is the input provided to the ANN. We call the input and desired output a learning pair. In order to

stabilize the inversion by ANN, we introduce a method that creates some additional learning pairs by a differential convolution model.

The simple convolution model is:

$$S = W*R+n \qquad (1)$$

Where S = the seismic trace,

W = a seismic wavelet,

R = earth reflectivity,

n = additive noise

$$R=(z2-z1)/(z2+z1) \qquad (2)$$

Where z = impedance

$$z=\rho V \qquad (3)$$

Where ρ = density

V = velocity

and,

$$\rho=f(V) \qquad (4)$$

In the training, the desired output is V and input is S. From equation (1)~(4), we can find that if we change V to V*, then we can get a correlative S* by the convolution model. So by changing V, we can get some new learning pairs.

The detailed process of creating new samples is show as the flow chart (Fig. 18). Using this technique, we created another six learning pairs. The original learning pair is well 62 and the seismic traces nearby. The inversion result (Fig. 19) is more stable but smoother than that of inversion without new learning pairs.

We selected some traces near by the wells from these inversion results, and compared them. Fig 20 shows that inversion by ANN can be better than conventional inversion. The correlation coefficient of inversion by ANN is much higher than that of conventional inversion. The preconditioned ANN (Inversion by ANN with new learning pairs) did not modify the correlation coefficient in well 143 (Fig 21). However, in well 108 (Fig. 22), the correlation coefficient of preconditioned ANN is higher than that of original ANN.

Since ANN's get the relationship of data by training, they can do inversion for any type of well log that has relationship with the seismic record. To prove this, we did the inversion for Gamma Ray (GR). The training process is pretty similar as inversion of velocity. The result (Fig. 23) shows that near well 108 and well 143 the inversion result and GR logging matched well in shallower layers.

CONCLUSION

1. Inversion by ANN can have higher resolution than the conventional inversion method.
2. Inversion by ANN can be stable in the presence of random noise.
3. Inversion by ANN can be done in the presence of multiples.
4. Inversion by ANN can be stabilized by creating some learning pairs with a differential convolution model.

REFERENCE

Jacek M. Zurada, 1992, Introduction to Artificial Neural Systems

Brian H. Russell, 1988, Introduction to Seismic Inversion Methods

Zhengping Liu, 1998, OTC, An Iterative Inversion of Seismic Data Constrained By Well Using Artificial Neural Networks

Abhijit S. Pandya, Robert B. Macy, 1995, Pattern Recognition with Neural networks in C++

THEORETICAL MODEL PARAMETERS

Model	wavelet			velocity of background (m/s)	velocity of the bed (m/s)	Noise	Maximum thickness of bed (ms)
	type	frequency	phase change				
Wedge1	Ricker	20	95	4000	1000	none	100
Wedge2	Ricker	20	95	4000	1000	10%	100
Multiple	Ricker	20	95	4000	1000	none	250

FIGURE NUMBER AND NAME

Fig. 1	Horizons and layer of initial model, the wedge without noise
Fig. 2	P-Velocity of initial model, the wedge without noise
Fig. 3	Seismic Traces of the wedge without noise
Fig. 4	Blocky Inversion of wedge without noise
Fig. 5	Sparse Spike Inversion of wedge without noise
Fig. 6	Inversion by ANN of wedge without noise
Fig. 7	Seismic Traces of the wedge with 10% noise
Fig. 8	Blocky Inversion of wedge with 10% noise
Fig. 9	Sparse Spike Inversion of wedge with 10% noise
Fig. 10	Inversion By ANN of wedge with 10% noise
Fig. 11	Inversion By ANN of wedge with 10% noise, six same noise level traces with different noise system join to the training.
Fig. 12	Seismic Traces of the wedge with the multiple
Fig. 13	Sparse Spike Inversion of wedge with the multiple
Fig. 14	Blocky Inversion of wedge with the multiple
Fig. 15	Inversion by ANN of wedge with the multiple
Fig. 16	Seismic Traces of real data, Las cavao, line 364
Fig. 17	Inversion by ANN of real data, P-Velocity
Fig. 17B	Conventional Seismic Inversion, P-Velocity
Fig. 18	Flow Chart of Learning Pair Pre-conditioning
Fig. 19	Inversion by ANN of real data, P-Velocity, with data pre-conditioning
Fig. 20	Well 62 inversion result
Fig. 21	Well 143 inversion result
Fig. 22	Well 108 inversion result
Fig. 23	Inversion by ANN of real data, GR

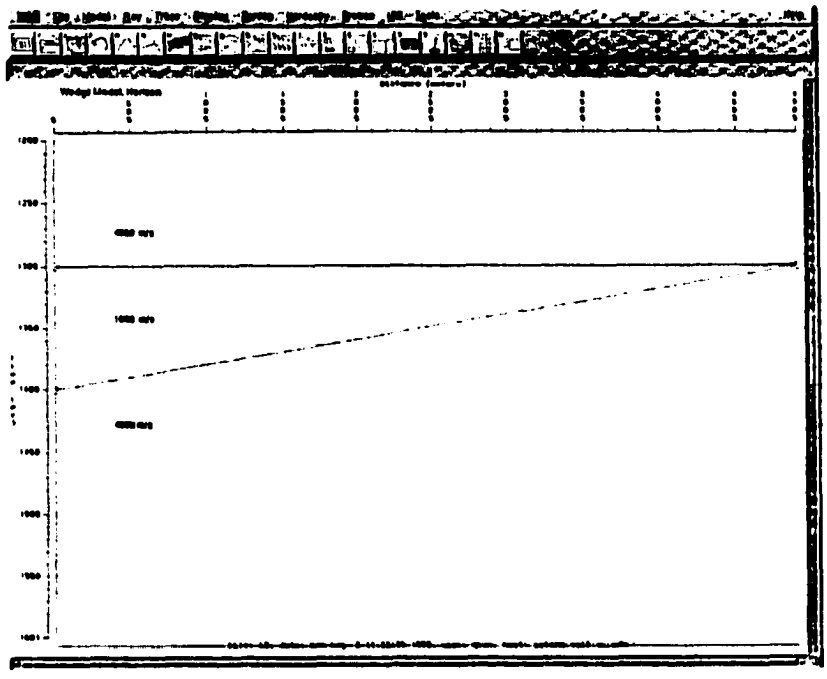


Fig. 1 Horizons and layer of initial model, the wedge without noise

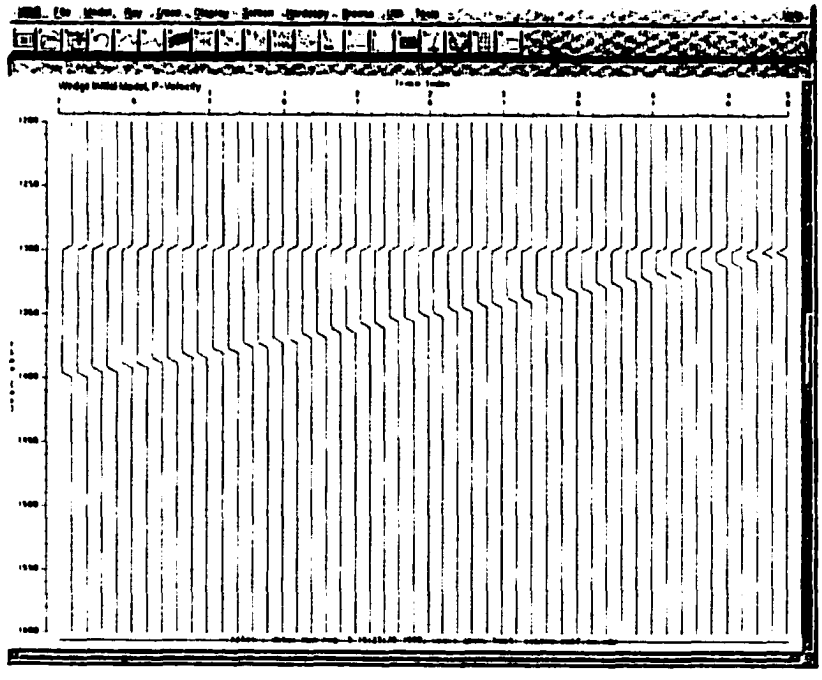


Fig. 2 P-Velocity of initial model, the wedge without noise

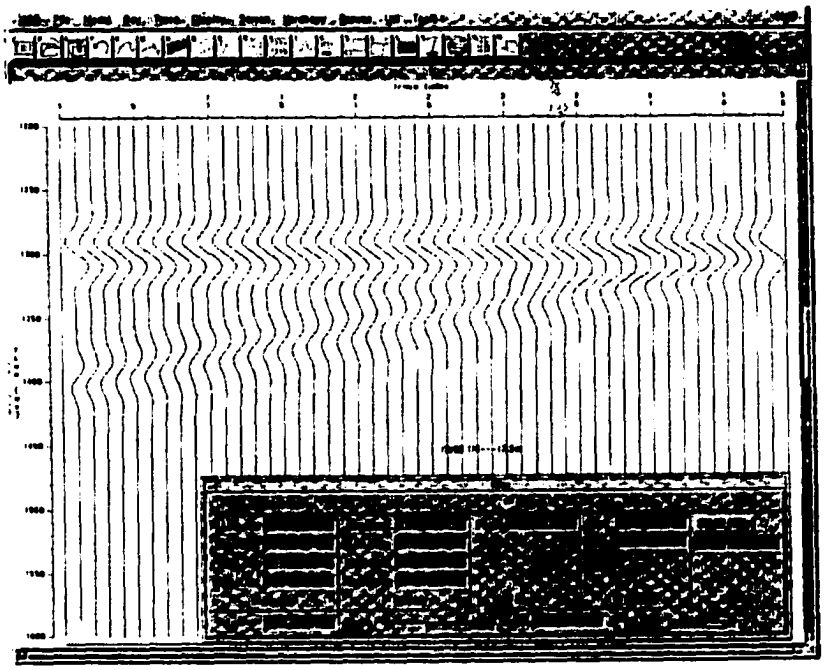


Fig. 3 Seismic Traces of the wedge without noise

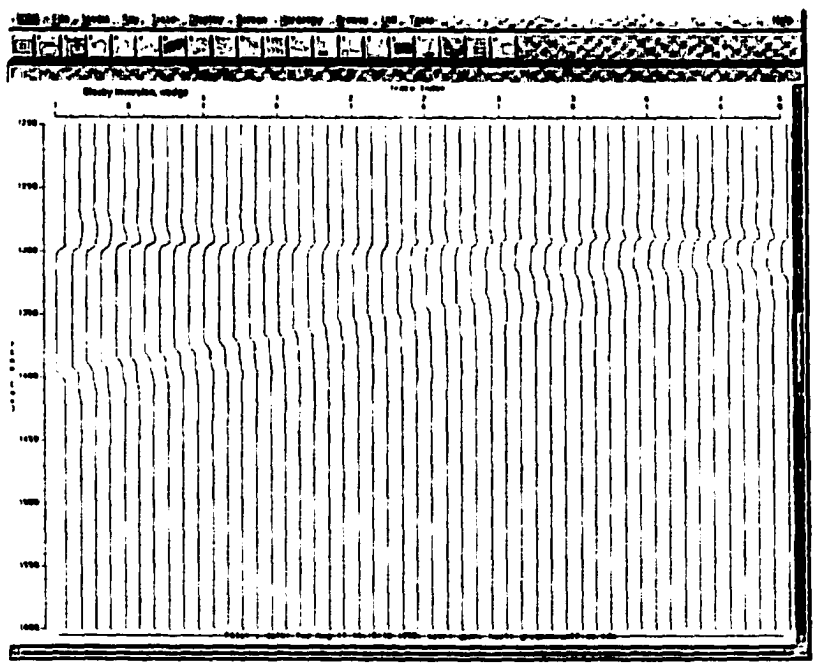


Fig. 4 Blocky Inversion of wedge without noise

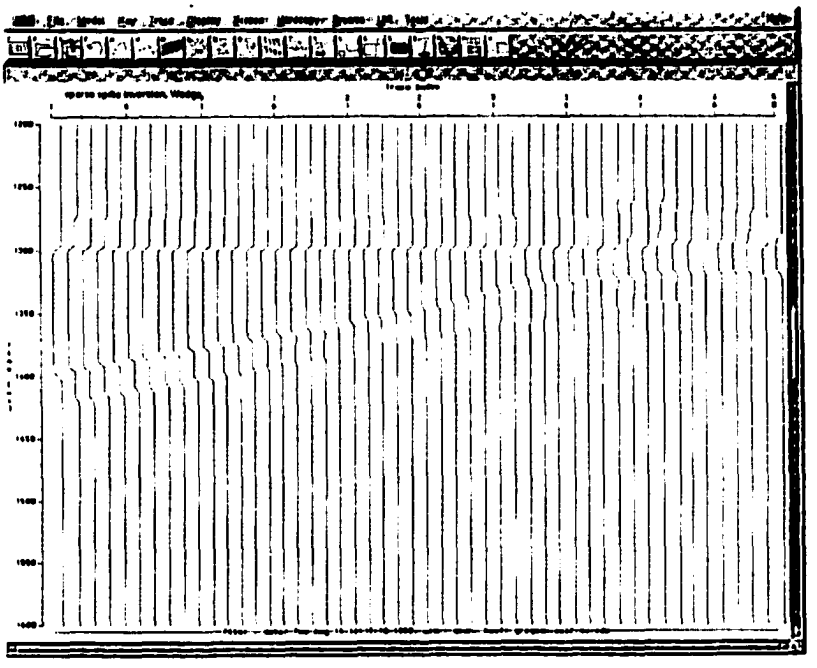


Fig. 5 Sparse Spike Inversion of wedge without noise

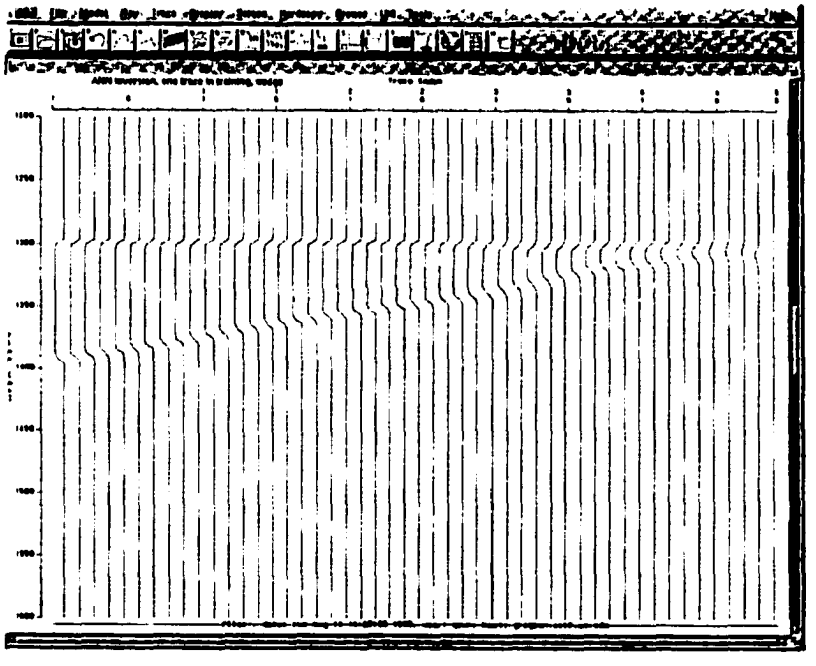


Fig. 6 Inversion by ANN of wedge without noise

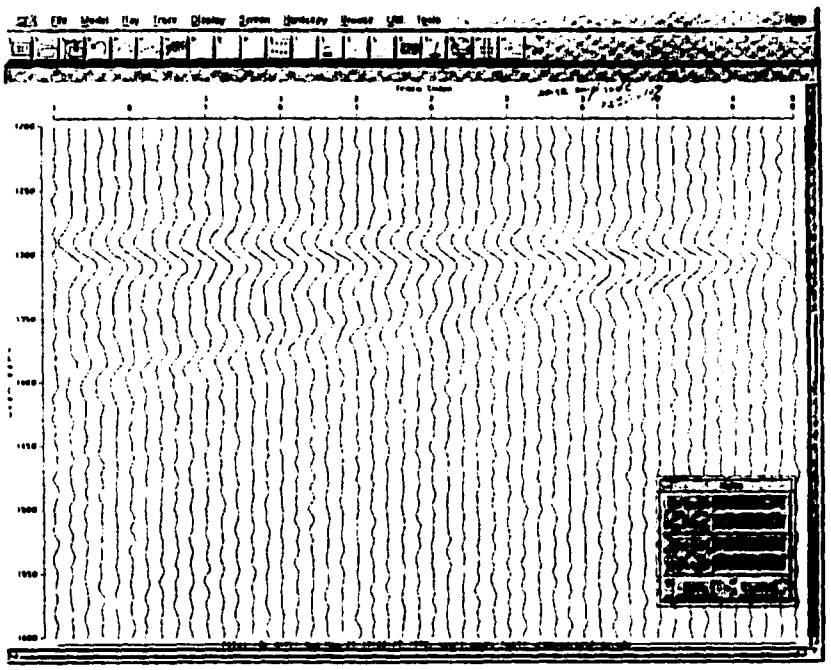


Fig. 7 Seismic Traces of the wedge with 10% noise

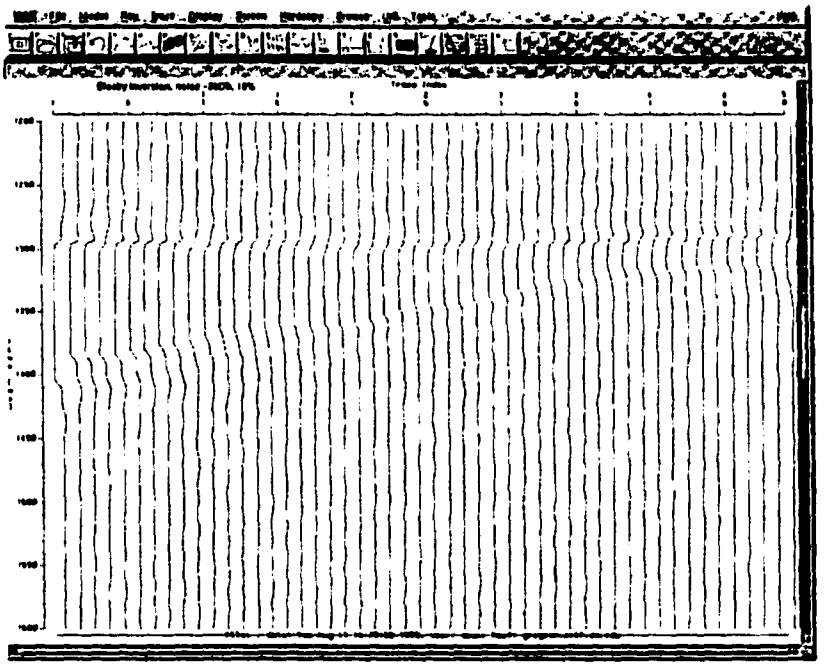


Fig. 8 Blocky Inversion of wedge with 10% noise

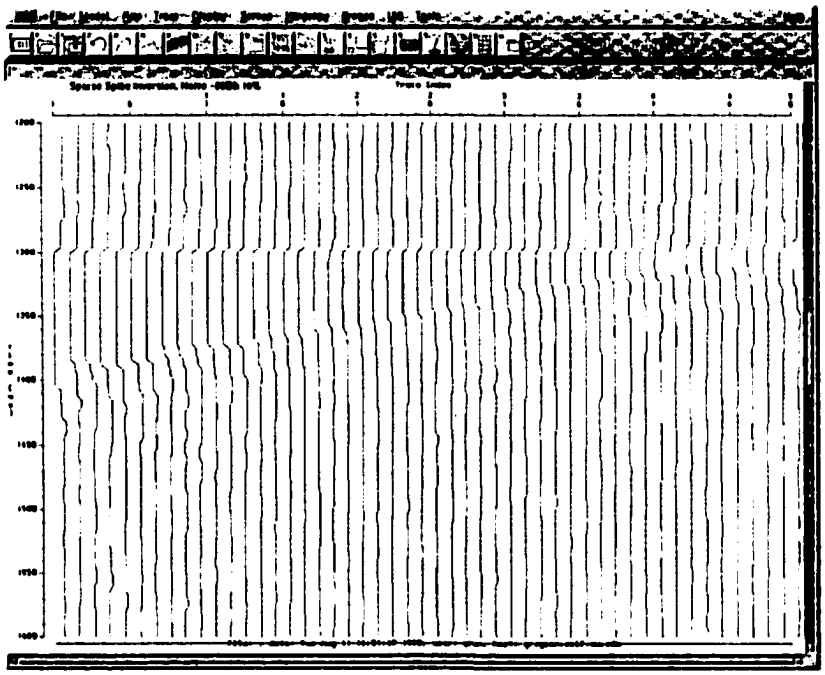


Fig. 9 Sparse Spike Inversion of wedge with 10% noise

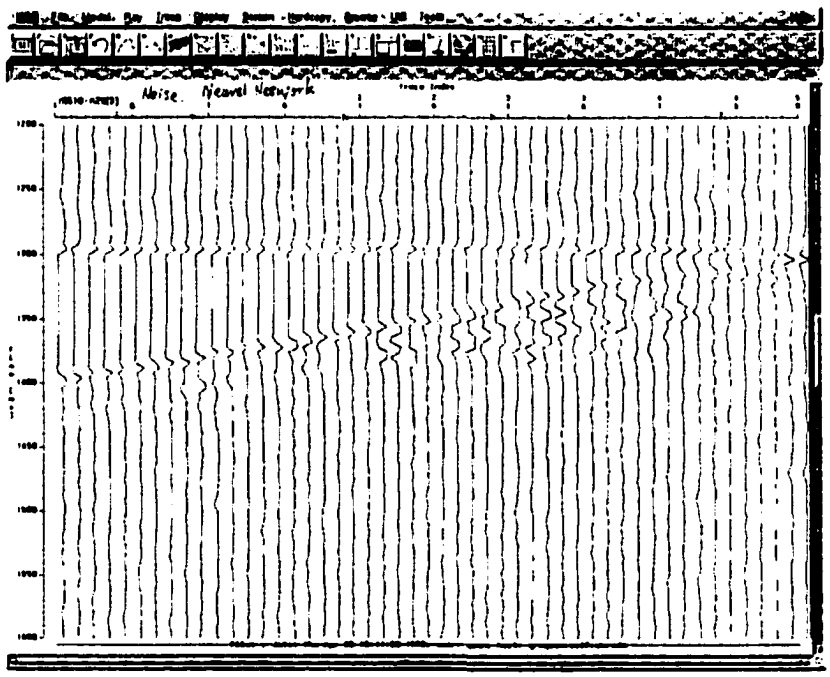


Fig. 10 Inversion By ANN of wedge with 10% noise

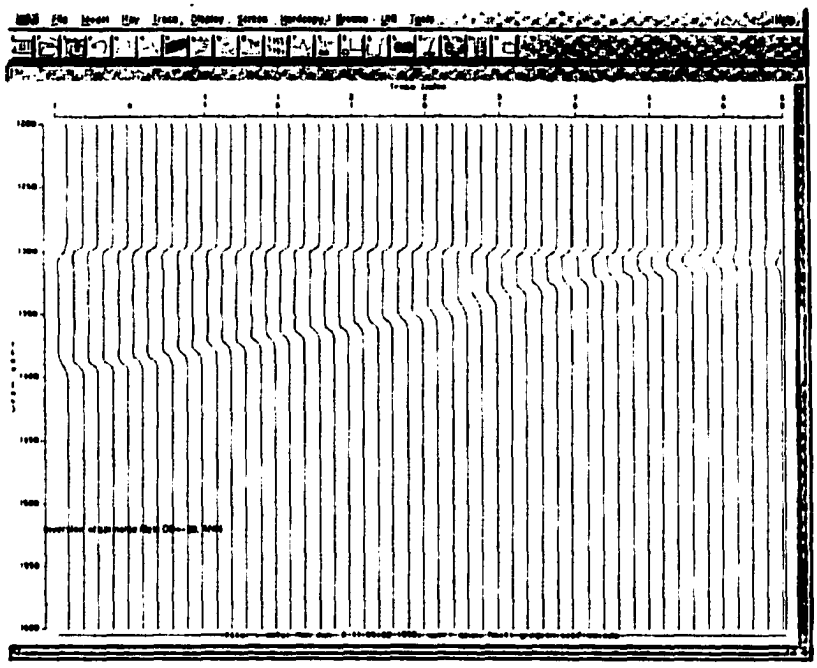


Fig. 11 Inversion By ANN of wedge with 10% noise, six same noi level traces with different noise system join to the training.

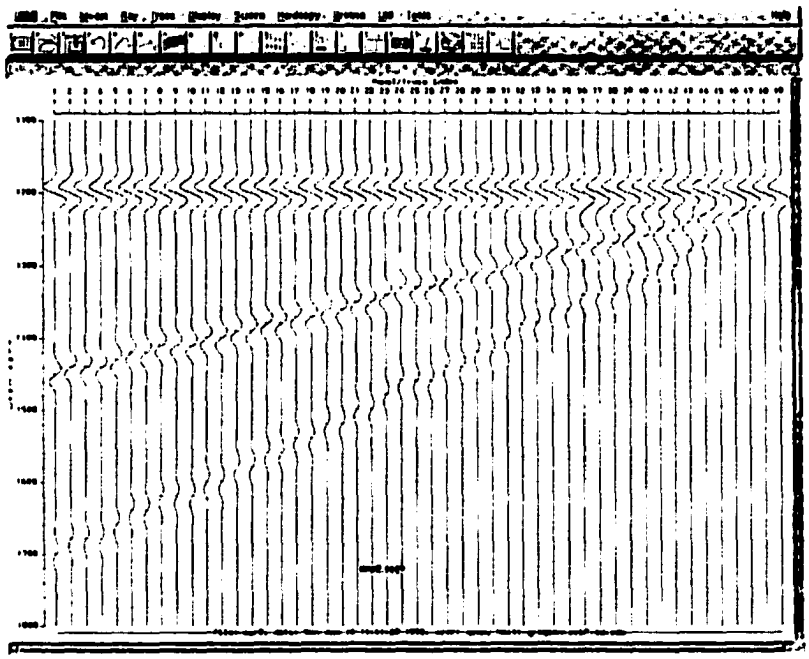


Fig. 12 Seismic Traces of the wedge with the multiple

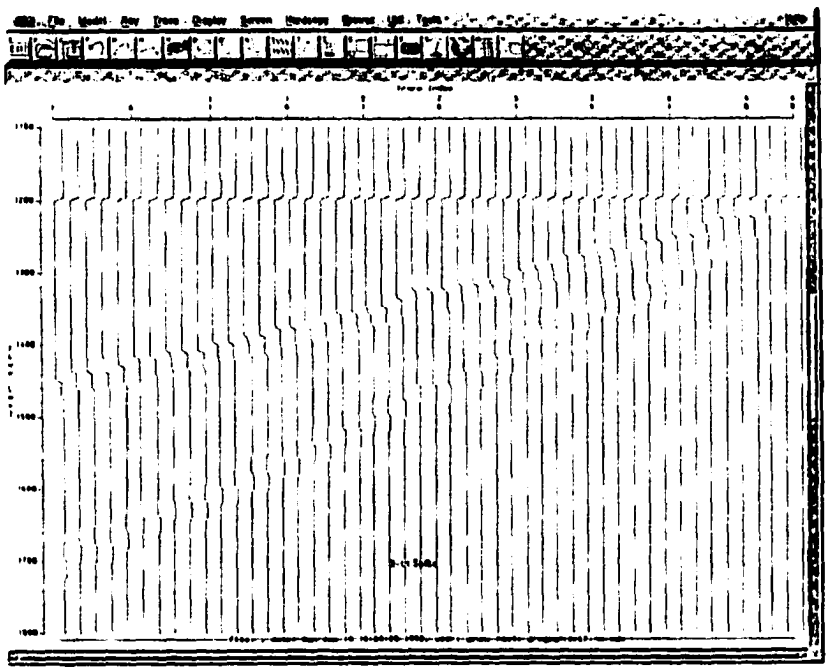


Fig. 13 Sparse Spike Inversion of wedge with the multiple

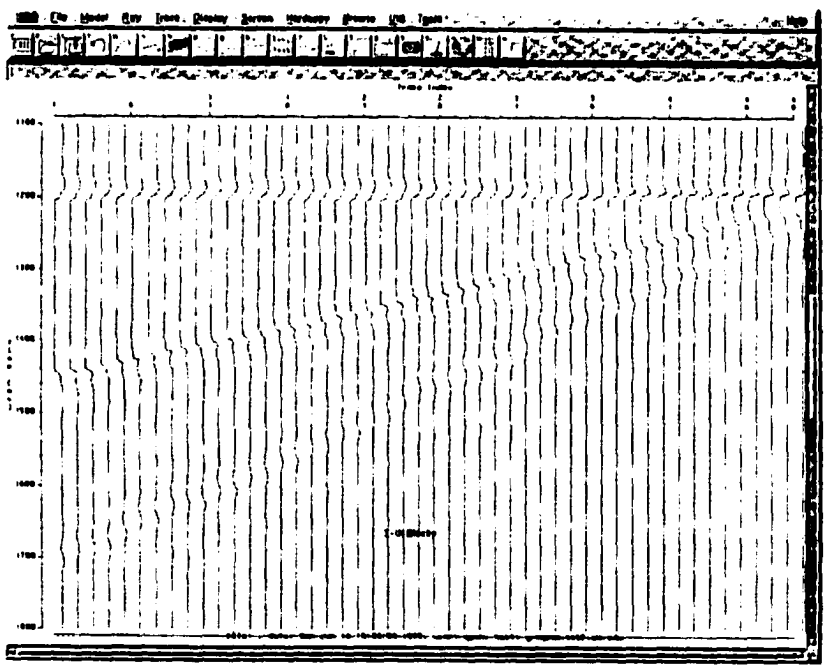


Fig. 14 Blocky Inversion of wedge with the multiple

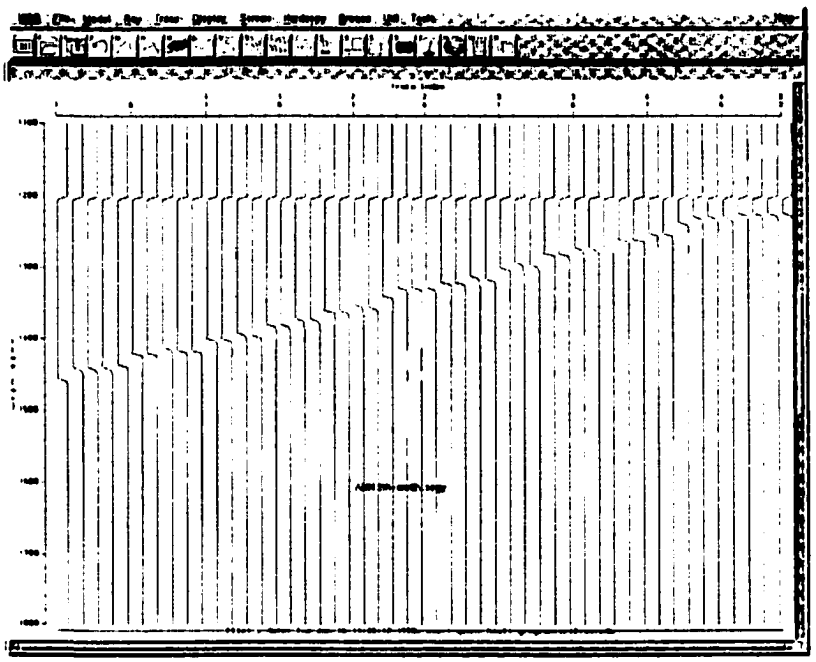


Fig. 15 Inversion by ANN of Wedge with the Multiple

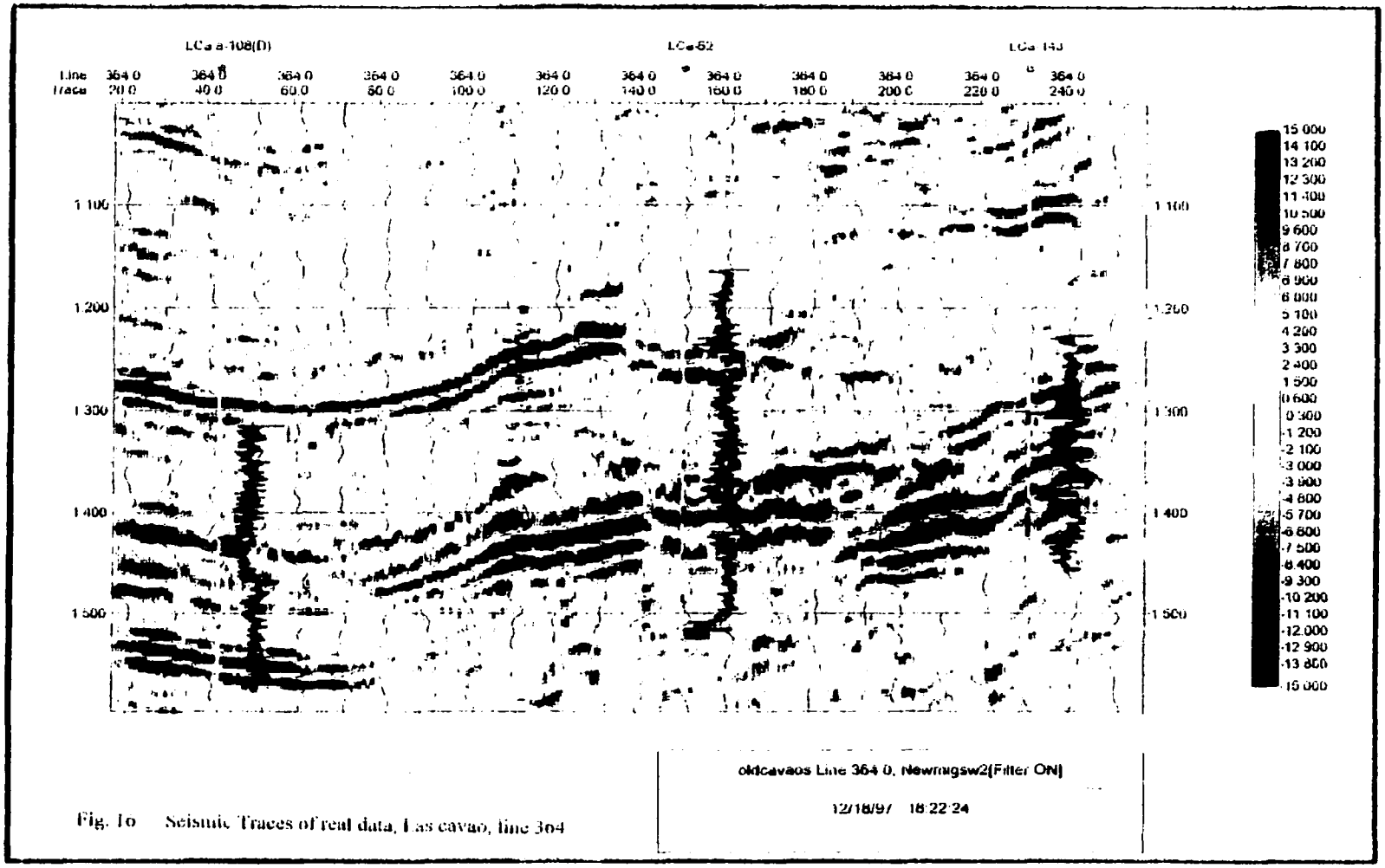


Fig. 1o Seismic Traces of real data, Las cavao, line 364

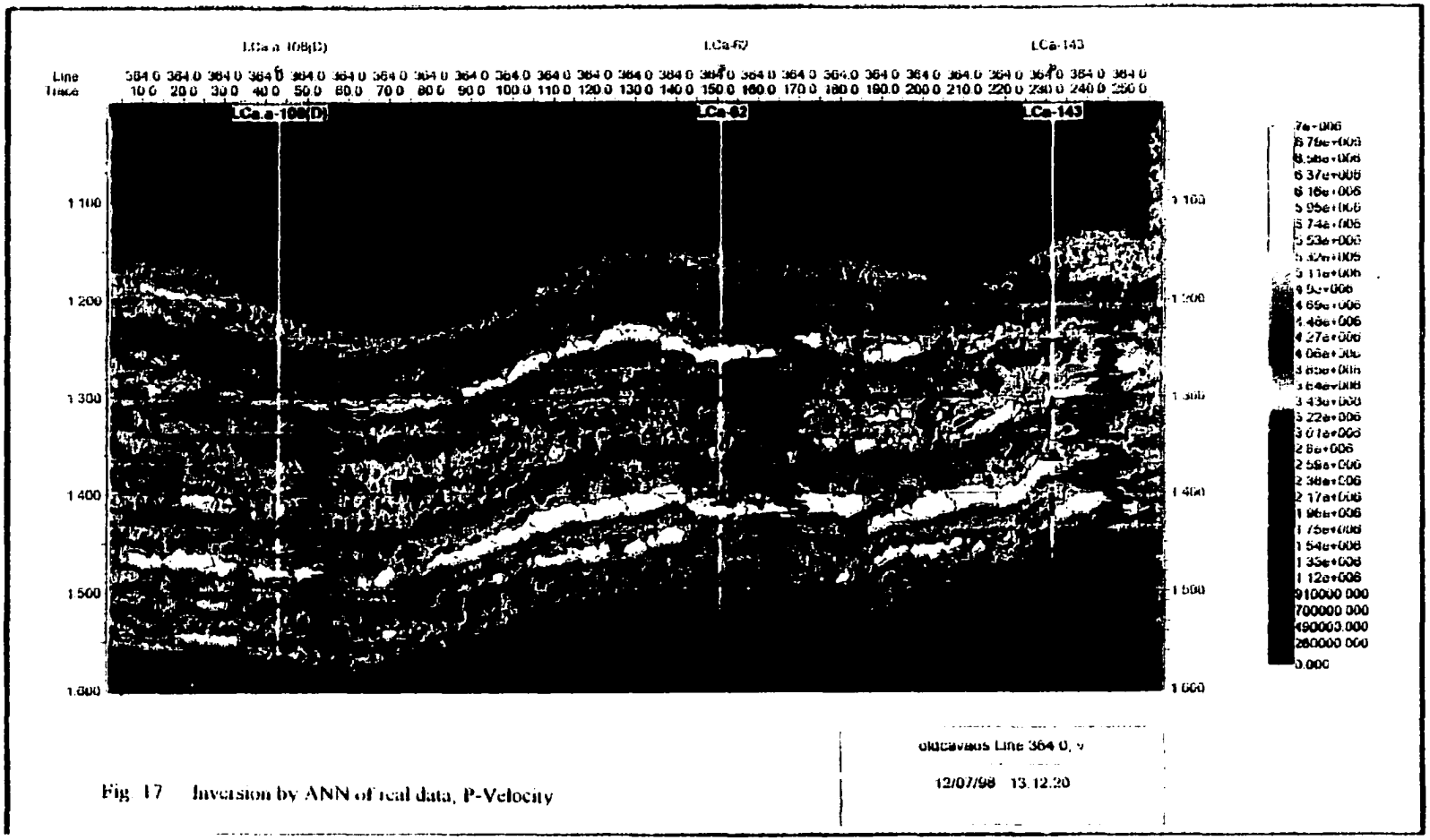


Fig 17 Inversion by ANN of real data, P-Velocity

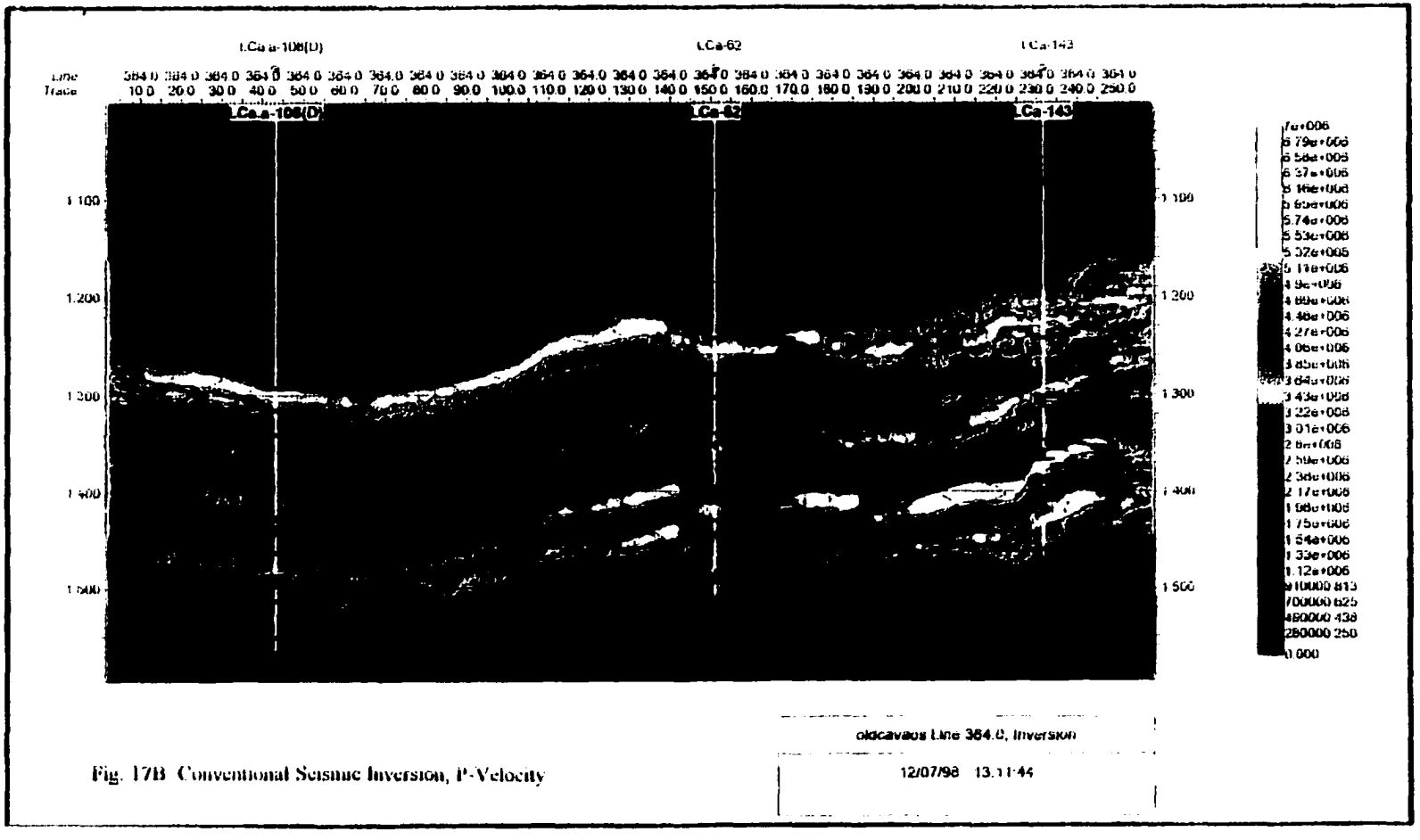


Fig. 17B Conventional Seismic Inversion, P-Velocity

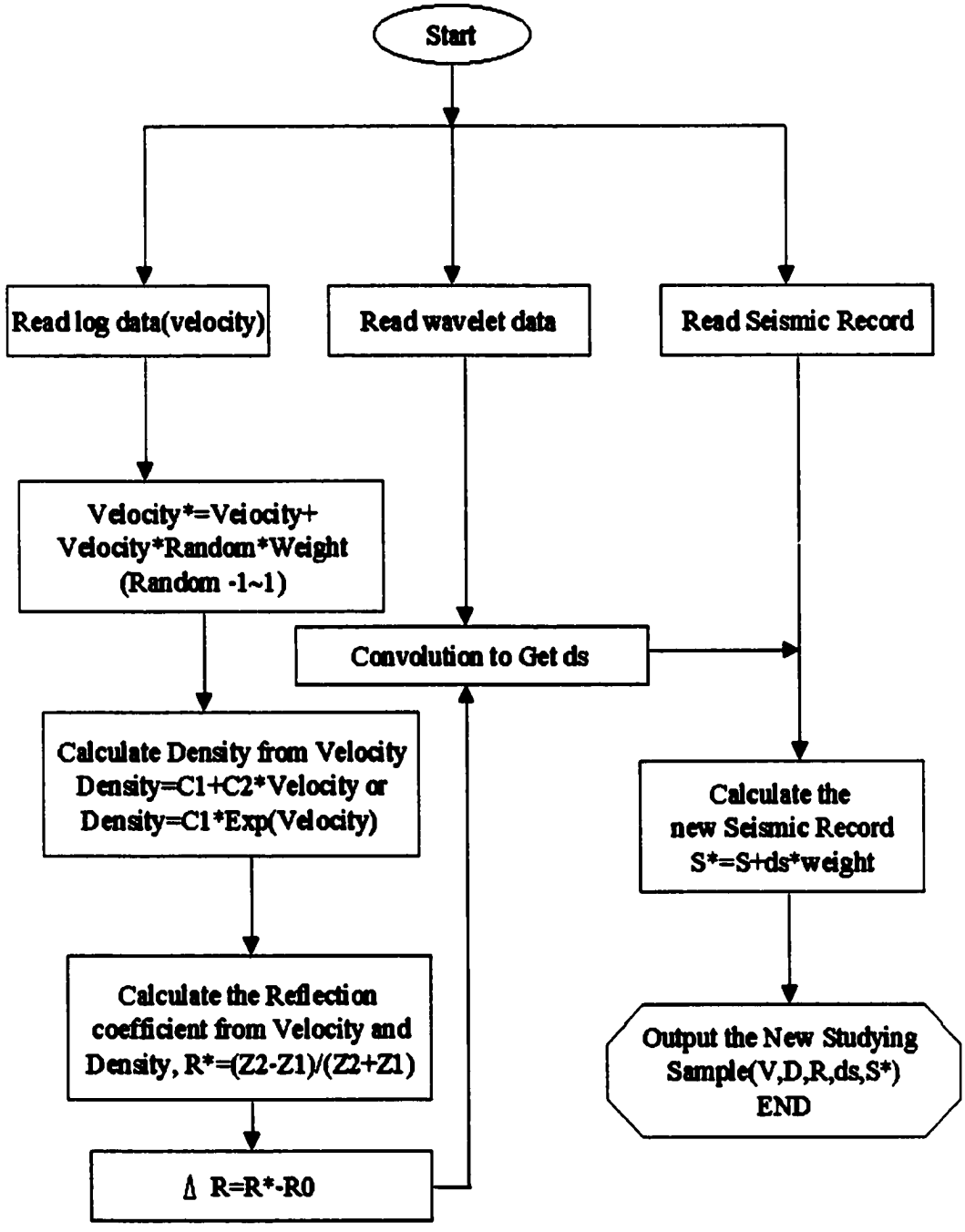


Fig. 18 Flow chart of studying pair pre-conditioning

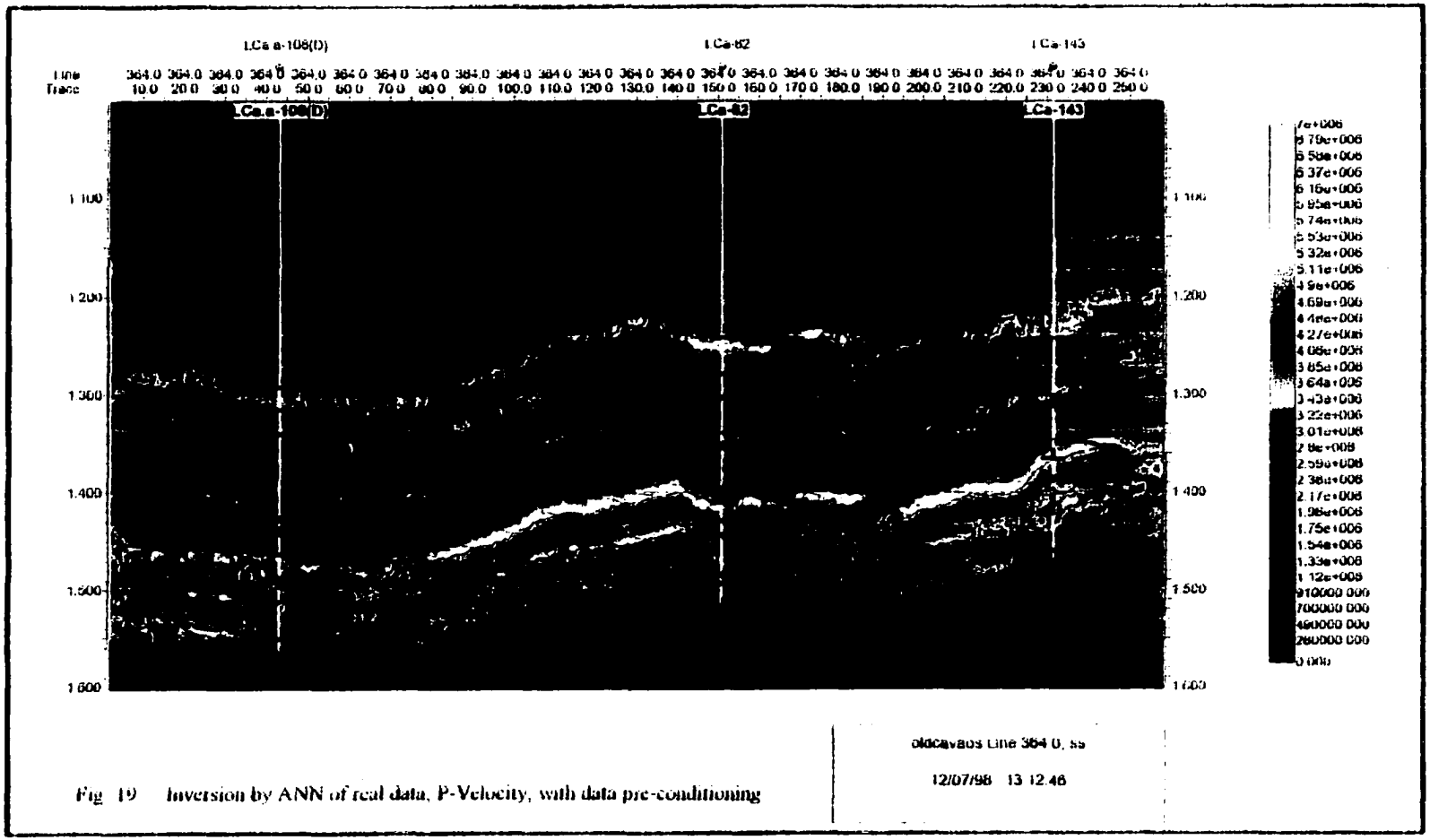


Fig. 19 Inversion by ANN of real data, P-Velocity, with data pre-conditioning

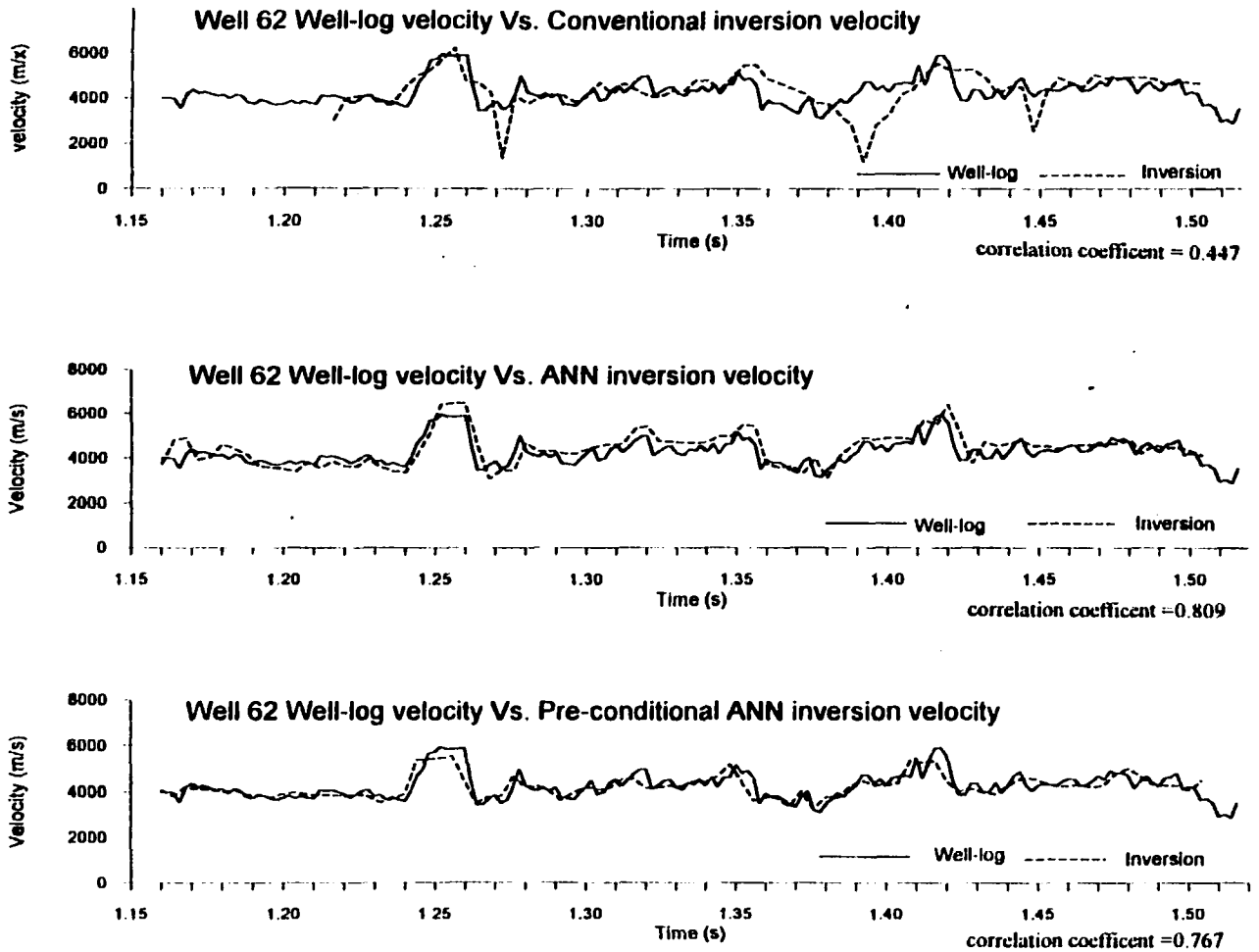
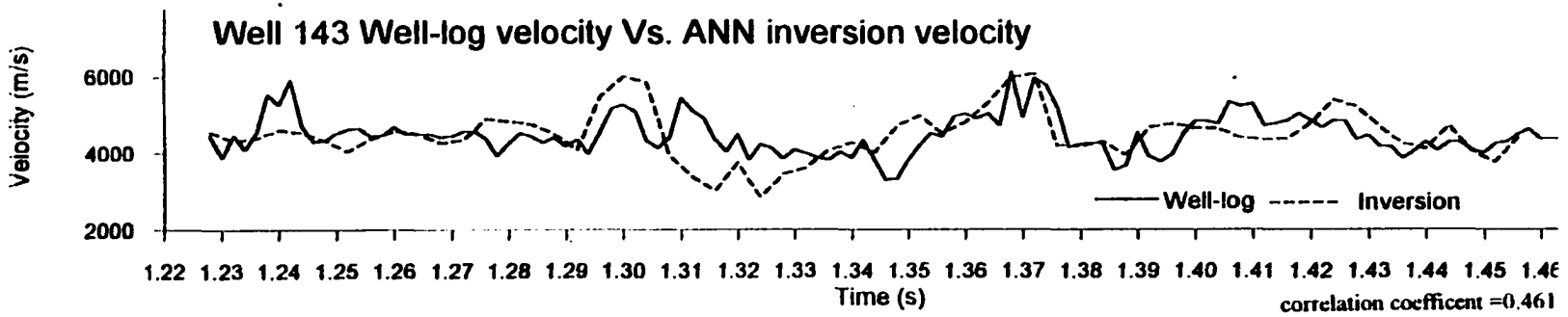


Fig. 20 Well 62 inversion result



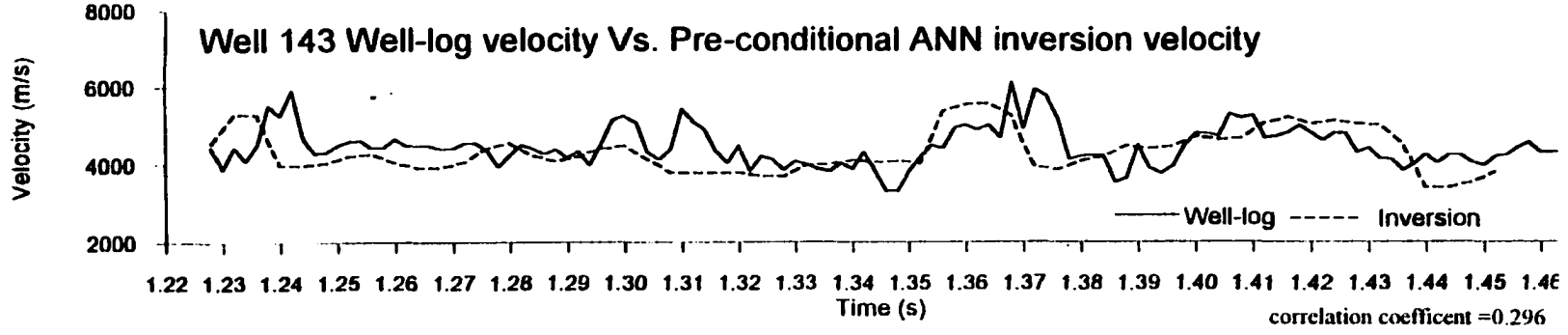


Fig. 21 Well 143 inversion result

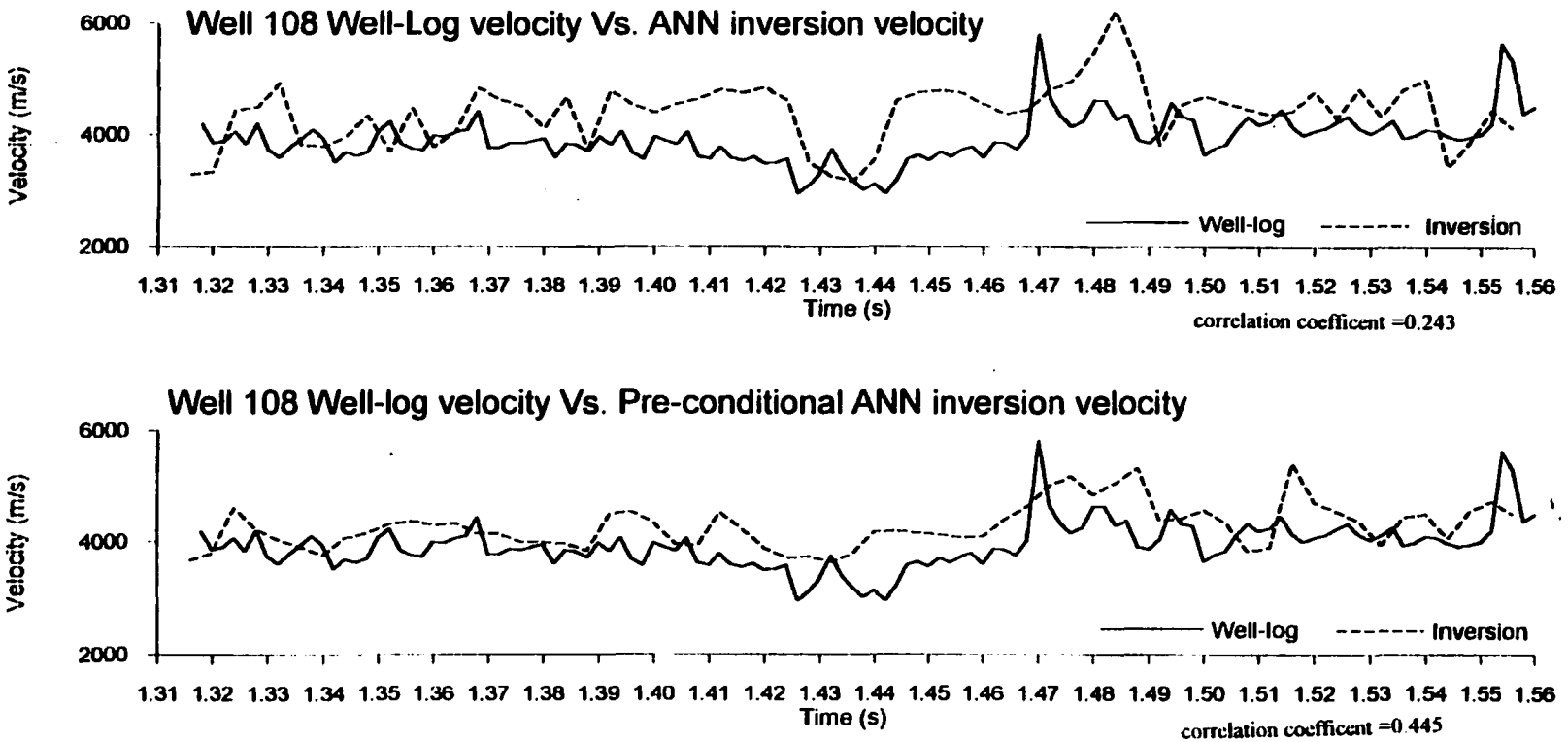


Fig. 22 Well 108 inversion result

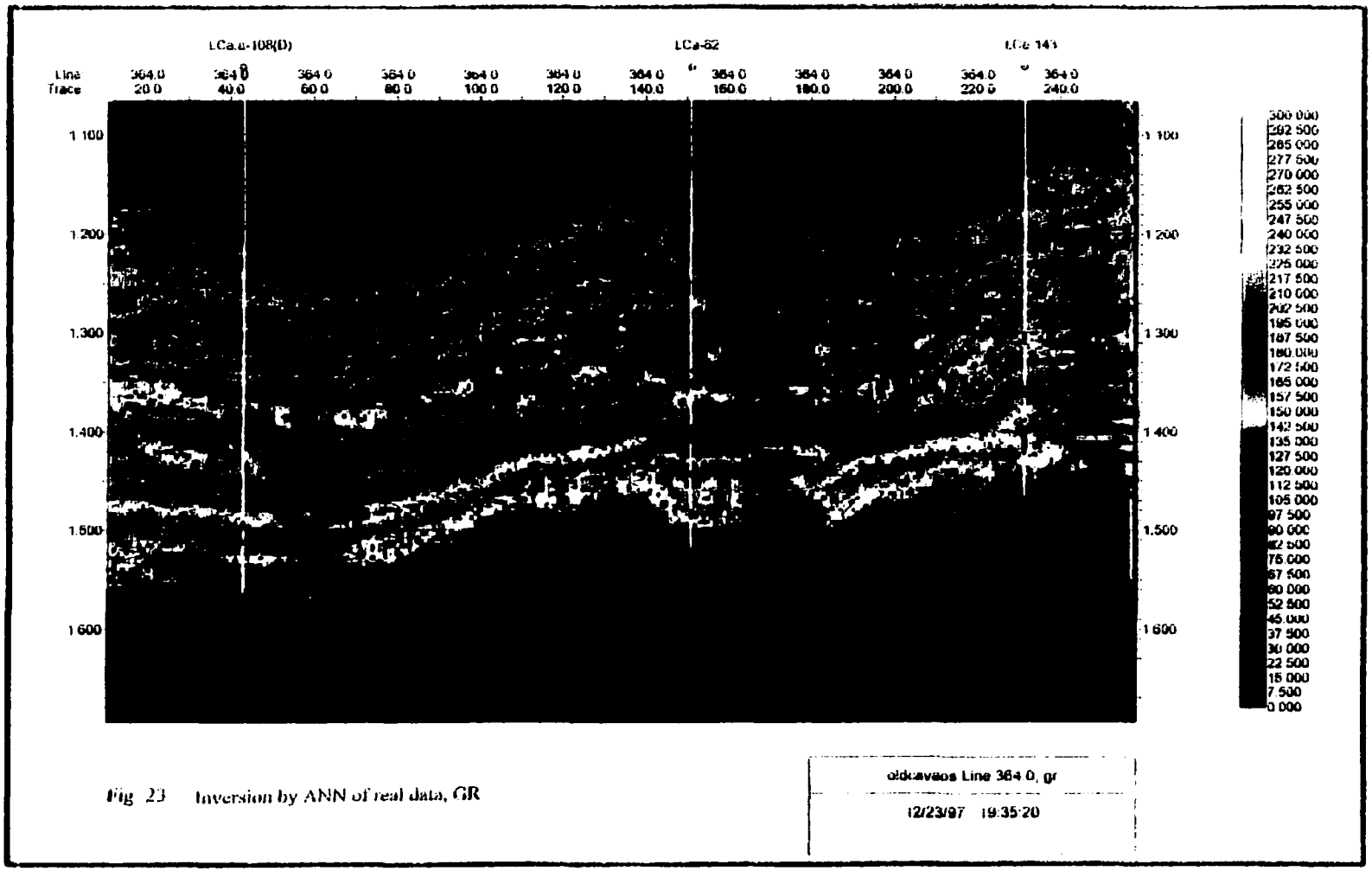


Fig. 23 Inversion by ANN of real data, GR

Appendix B

Resistivity and Lithology Fraction inversion for the YPF data set

Data available:

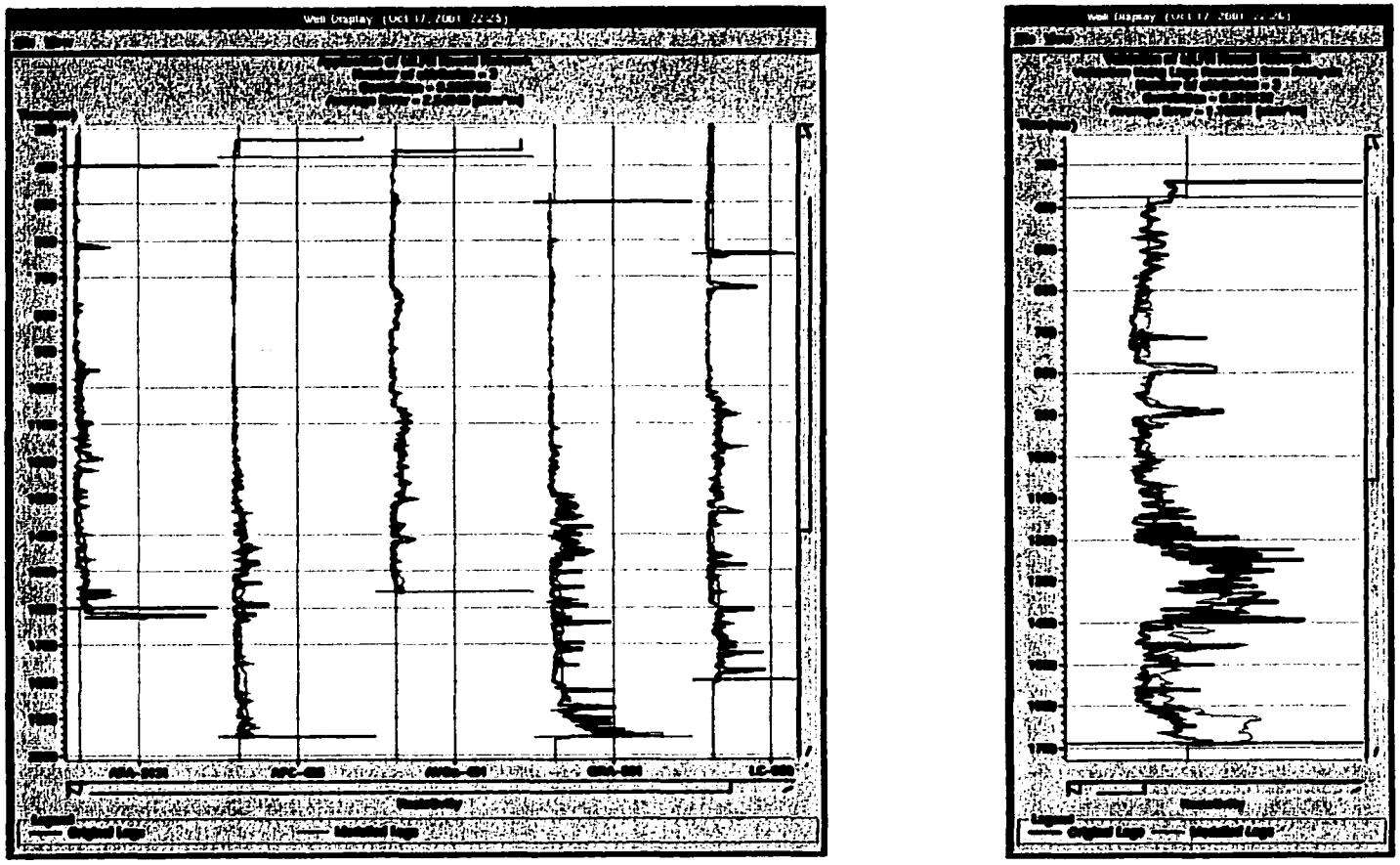
The data available are extracted S-impedance, P-impedance, and density for a 2D arbitrary line, and 6 wells with resistivity log and lithology fraction log on the 2D line.

Method:

Two BP ANN's were trained to learn the relationship between the seismic attributes and the well logs. The input data are S-impedance, P-impedance and density for both networks. The desired output data for the first network is the resistivity, and the desired output data for the second network is the lithology fraction. Training window length is 28 ms. The network had one hidden layer with 13 nodes. Five wells were used in training one well was used in validation. The software I used is Hampson-Russell Emerge, version 2.1. This new version has the neural network function.

Results:

The predicted well log match the true well log very well for both resistivity and lithology faction.



(a) (b)

Figure 1. The observed resistivity and the predicted resistivity for (a) the training wells and (b) the validation well.

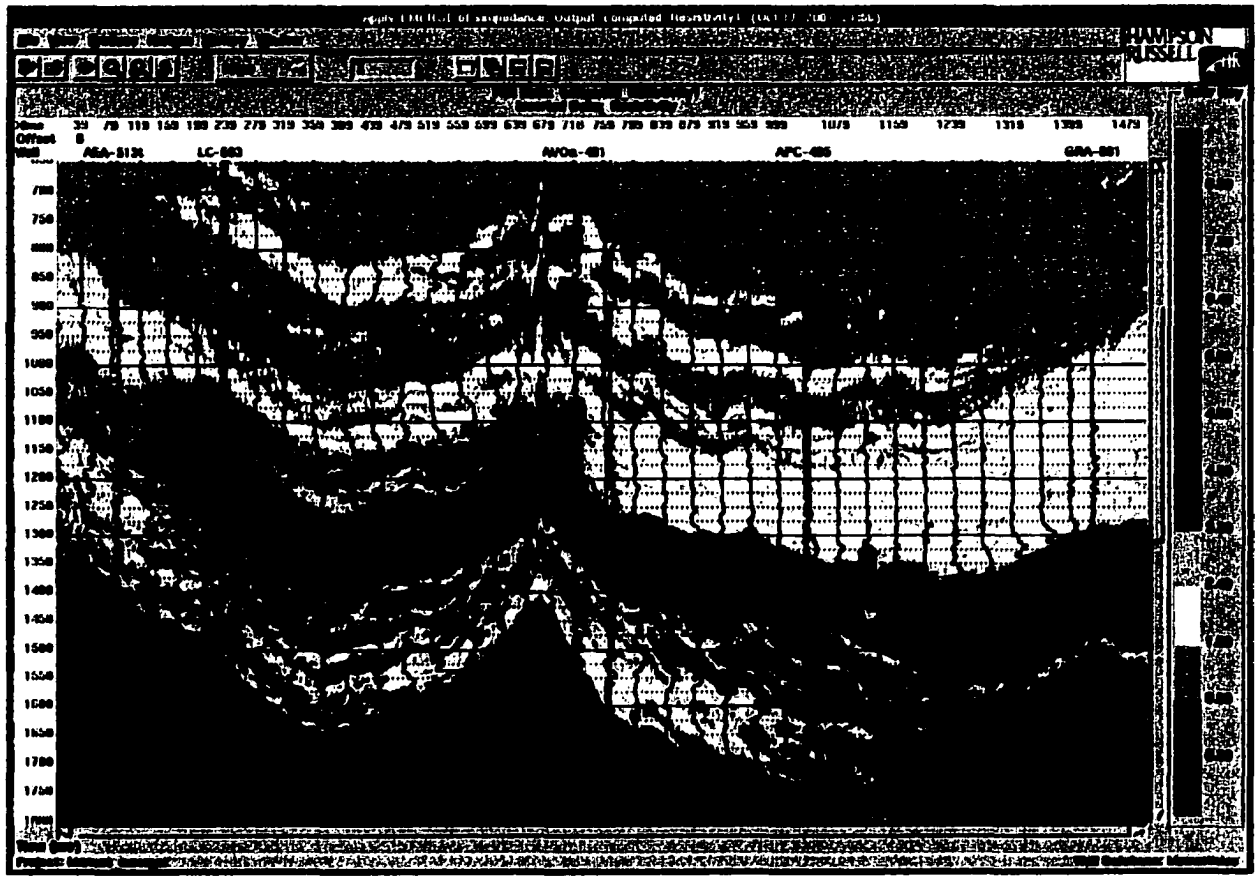
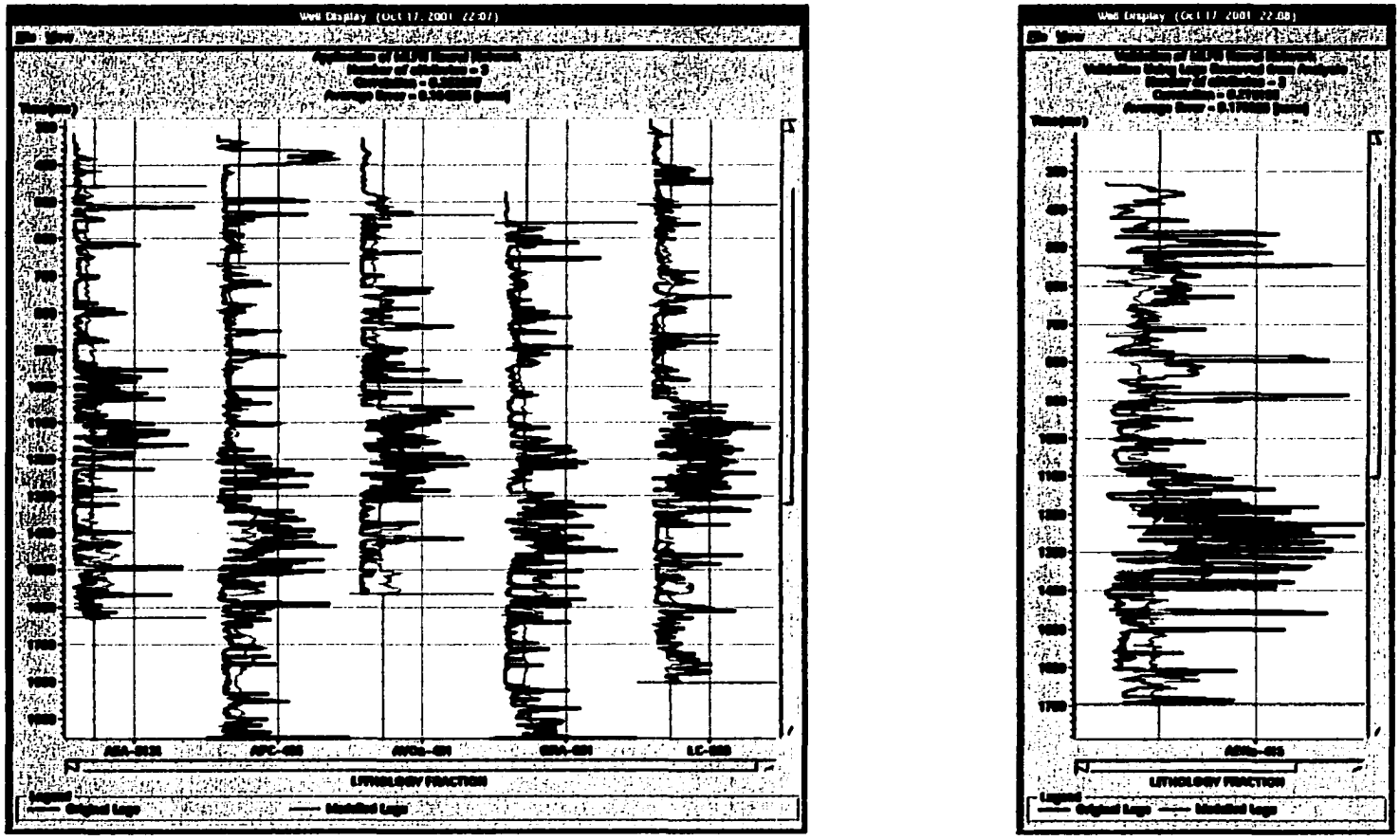


Figure 2. The ANN predicted resistivity for the whole 2D arbitrary line.



(a) (b)

Figure 3. The observed lithology fraction and the predicted lithology fraction for (a) the training wells and (b) the validation well.

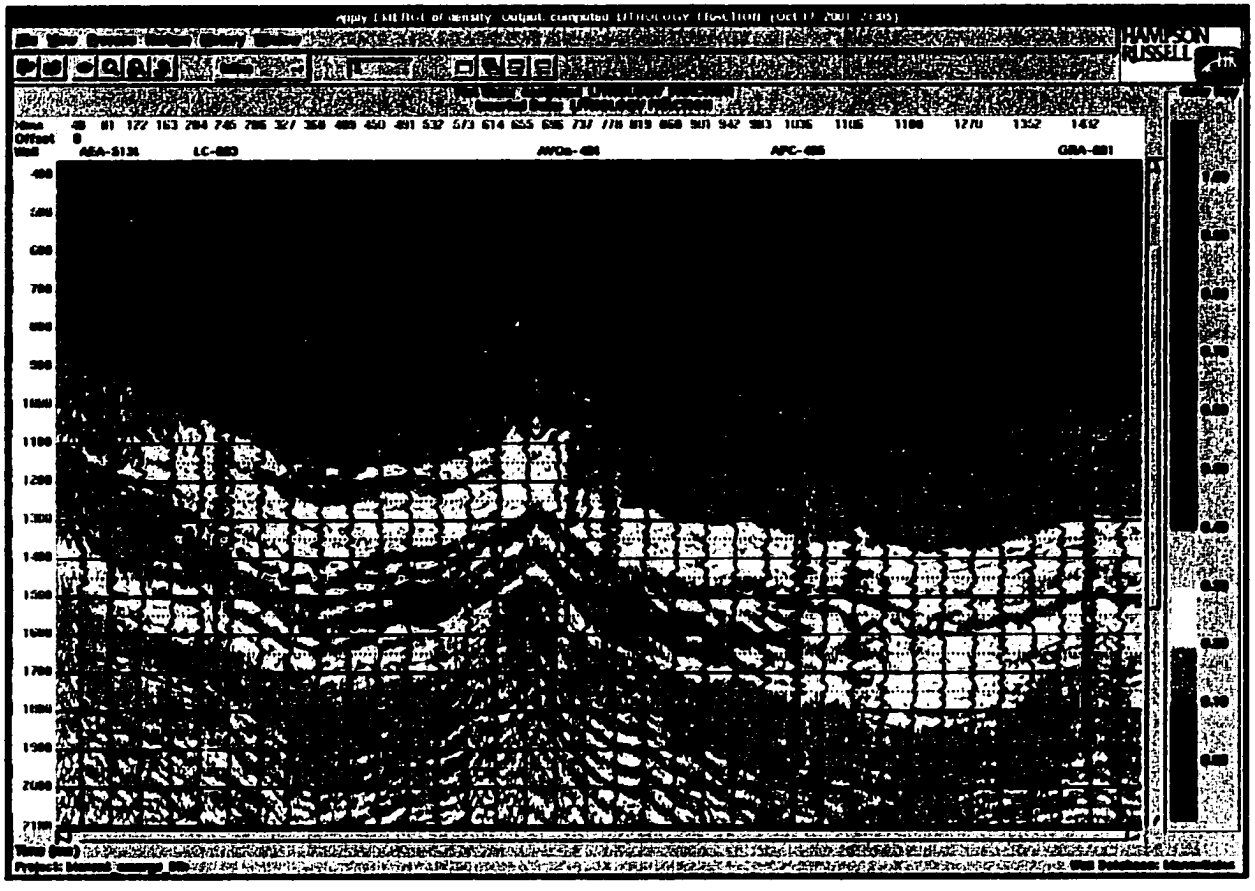


Figure 4. The ANN predicted lithology fraction for the whole 2D arbitrary line.