

APPLICATION OF ARTIFICIAL NEURAL NETWORKS TO
GEOACOUSTIC INVERSION

by

JEREMY BENSON

B.E., University of Saskatchewan, 1996


A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

We accept this thesis as conforming
to the required standard


Dr. A. Antoniou, Supervisor, Dept. of Elect. & Comp. Eng.


Dr. N. R. Chapman, Supervisor, School of Earth & Ocean Sciences


Dr. N. Dimopoulos, Departmental Member, Dept. of Elect. & Comp. Eng.


Dr. S. Dosso, Outside Member, School of Earth & Ocean Sciences


Dr. D. Thomson, External Examiner, Defence Research Establishment Atlantic

© JEREMY BENSON, 1998

University of Victoria

*All rights reserved. This thesis may not be reproduced in whole or in part by
photocopy or other means, without the permission of the author.*

Q476.87
B46

Supervisors: Dr. A. Antoniou and Dr. N. R. Chapman

ABSTRACT

Artificial neural networks (ANNs) were used to estimate ocean bottom parameters from experimental data collected at sea during the TRIAL SABLE experiment. Both backpropagation ANNs (BPNNs) and radial basis function ANNs (RBFNNs) were investigated. These networks use supervised training to learn the behavioural mapping from the data to the bottom parameters. The input/output examples of the training data were generated using the geoacoustic forward model ORCA, which predicts the experimental data from the geoacoustic properties of the ocean and ocean bottom. The network was trained to approximate the inverse of the ORCA model. The TRIAL SABLE experiment involved a vertical line array of hydrophones which sampled the pressure field generated by a towed multi-tone continuous wave source. From the *a priori* information available for the experiment site we were able to construct a set of nominal ocean parameters with an upper and lower bound for each parameter. Using ORCA we show that the field is strongly dependent on the ocean parameters which are not known precisely. We train two types of neural networks to estimate these parameters from the data. The BPNN achieves good estimation accuracy for several of the most important parameters, but other parameters are poorly estimated. The RBFNN estimates the single most important parameter very well but the other parameter approximations are inaccurate. We propose that this performance difference is due to the choice of training algorithm. The problem in estimating the less important parameters is overcome by simplifying the task to one of bottom classification. The bottom classification is done with the two neural networks using supervised training. The BPNN performs very well with only a 4% chance of misclassification. The RBFNN has a 20% probability of misclassification. The BPNN classifies the data as resulting from a gravel bottom, and the RBFNN chooses sand. The bottom type is known *a priori* to consist of a mixture of sand and gravel.

Examiners:



Dr. A. Antoniou, Supervisor, Dept. of Elect. & Comp. Eng.



Dr. N. R. Chapman, Supervisor, School of Earth & Ocean Sciences



Dr. N. Dimopoulos, Departmental Member, Dept. of Elect. & Comp. Eng.



Dr. S. Dosso, Outside Member, School of Earth & Ocean Sciences



Dr. D. Thomson, External Examiner, Defence Research Establishment Atlantic

Table of Contents

Abstract	ii
Table of Contents	iv
List of Figures	vii
List of Tables	xii
Notation	xiii
Acknowledgement	xv
1 Introduction	1
2 Ocean Acoustics Introduction	7
2.1 Acoustic Modeling	9
2.1.1 The Wave Equation	9
2.1.2 Numerical Solutions	11
2.1.3 The Normal Mode Solution	13
2.1.4 ORCA	17
2.1.4.1 The SVP file	17
2.1.4.2 The OPT file	19
2.1.4.3 Implementation and Testing	19
2.2 Source Localization in a Waveguide	20
2.2.1 Matched-Field Processing	20
2.2.2 Simulation Example	22
2.2.3 The Mismatch Problem	25
2.3 Matched Field Inversion	27
2.4 Conclusions	28

3	Neural Inversion	29
3.1	System Identification	29
3.1.1	Problem Formulation	29
3.1.2	Neural System Identification	31
3.1.3	Neural Classifiers	34
3.2	Artificial Neural Networks	35
3.2.1	Backpropagation Neural Networks	38
3.2.2	Radial Basis Function	44
3.2.3	Pre-processing	48
3.2.4	Identifiability Conditions	50
3.3	Conclusions	52
 4	 TRIAL SABLE Inversion	 53
4.1	Introduction	53
4.2	Environment Parameterization	54
4.2.1	Source locations	54
4.2.2	Receiver Locations	54
4.2.3	Bathymetry	57
4.2.4	Sound Speed Profile	58
4.2.5	Bottom Properties	60
4.2.6	Source Signature	63
4.2.7	The ORCA Environment	70
4.3	Simulation Study	70
4.4	Source Localization with Experimental Data	75
4.5	Conclusions	80
 5	 Results and Discussion	 81
5.1	Backpropagation Results	81
5.1.1	Range Finding	82
5.1.2	Range Finding with Mismatch	92
5.1.3	Inversion for Group II–IV Parameters	93
5.1.4	Bottom Classification	97
5.2	Radial Basis Function Results	102

5.2.1	Range Finding	103
5.2.2	Range Finding with Mismatch	104
5.2.3	Inversion for Group II–IV Parameters	105
5.2.4	Bottom Classification	106
5.3	Discussion of Results	108
5.3.1	Application of New Environments to Source Localization . . .	111
5.3.2	Sources of Error	114
5.4	Conclusions	114
6	Conclusions and Recommendations	115
6.1	Summary and Conclusions	115
6.2	Suggestions for Further Research	117
	Bibliography	120
	Appendix A ORCA Input Files	123

List of Figures

Figure 2.1	A typical ocean acoustic experiment.	8
Figure 2.2	A circularly symmetric parallel plate waveguide has constant properties over any change in bearing or range. The properties vary only in depth.	8
Figure 2.3	The calibration environment for the matched-field inversion workshop.	18
Figure 2.4	A comparison of the pressure calculated by ORCA and the calibration data set (SAFARI).	23
Figure 2.5	The transmission loss in range and depth for a 35 Hz signal in the calibration environment. The horizontal lines indicate the water-sediment and sediment-basement interfaces. The sidebar indicates the loss in -dB.	24
Figure 2.6	The Bartlett ambiguity surface for a 35 Hz signal in the calibration environment. The source is correctly localized at a depth of 20 m and a range of 1 km.	26
Figure 2.7	The Bartlett ambiguity surface for a 35 Hz signal with a mismatch in the water depth. The actual water depth is 100 m, and the modeled depth is 90 m. The predicted source depth is ≈ 30 m, and the predicted range is ≈ 1.5 km.	26
Figure 3.1	An artificial neural network (ANN) being trained with a forward model (ORCA). The data vector \mathbf{d} is the output of the model for the environment vector $\boldsymbol{\xi}$. The difference between the environment predicted by the inverse model G and the actual environment is used to update the weights in G	31
Figure 3.2	A neural identification system.	32
Figure 3.3	A general feedforward neural network.	35

Figure 3.4	A summing neuron. The activation function $f(s)$ can have many forms, although nonlinear differentiable functions are preferred. The bias b is usually included as a variable in the training algorithm.	36
Figure 3.5	Several activation functions commonly used in neural networks. Note that $s = \sum_{i=1}^n w_i x_i$.	37
Figure 3.6	A radial basis function neuron. Note that $e = \ \mathbf{x} - \mathbf{c}\ _2^2$.	37
Figure 3.7	A generalized layer for the backpropagation algorithm.	38
Figure 3.8	An undertrained, well-trained, and overtrained network. The overtrained network is said to have memorized the training data rather than the functional mapping. The middle network <i>generalizes</i> well for data in the test set.	43
Figure 3.9	A training record for cross-validation noting the optimal stopping point.	44
Figure 3.10	A radial basis function neural network. The hidden layer is composed of radial basis function neurons, and the output layer is linear.	45
Figure 4.1	The ship track overlaying the bathymetry of the experimental location shows a range-dependent environment. The ship track is represented by '+'s. The lower track is the outgoing leg and the upper track is the incoming leg. The first portion of the outgoing leg is used in this thesis. The array location is noted by an asterisk in the lower left-hand corner.	55
Figure 4.2	The outgoing ship track and the location of the array are shown. Each successive circle is one minute further up the track.	57
Figure 4.3	The depth below the ship track shows that for the first kilometer the dip is less than 1.25 m, thus the slope is less than 0.125 %.	58
Figure 4.4	The measured sound speed profile at the array and its parameterization.	59
Figure 4.5	Spectrogram, averaged periodogram and time domain signal for the first hydrophone for a source range of approximately 0.5 km.	64
Figure 4.6	The received signal power over range generated by ORCA for ten frequencies. The total range spanned is 300 m.	66

Figure 4.7	The computation of spectral estimates using the averaged periodogram.	67
Figure 4.8	Experimental spectral components estimated with a resolution of ≈ 3 m. The total range spanned is one-half that of Figure 4.6. . .	69
Figure 4.9	The parameterized ORCA environment.	71
Figure 4.10	Simulation study for the nominal parameter environment. Each parameter is varied while the other parameters are held constant at their nominal value and the Bartlett mismatch of Equation 2.43 is calculated.	73
Figure 4.11	The Bartlett surfaces for nine separate frequencies and the incoherent average of the surfaces (with the exception of the 110 Hz surface) show that for the nominal parameter values source localization is unsuccessful.	77
Figure 4.12	The Bartlett power for all frequencies and the incoherent mean of these surfaces show that the source localization is improved with the new parameters. The 55 Hz surface is the most influential with a peak power of 0.94.	78
Figure 4.13	The Bartlett power averaged incoherently for all frequencies and for the low frequency band (35, 55, 85 Hz) shows that the most important parameters are at or near their peak values for the new parameters.	79
Figure 5.1	The effect of standardization. (a) the original signal, (b) the signal with the mean removed, (c) the signal with the standard deviation set to unity. The <i>shape</i> of the signal is preserved, whereas the effect of possibly unknown gains and offsets has been removed. The <i>x</i> -axis indicates the sample number.	83
Figure 5.2	The convergence history for the training, test and cross-validation data sets shows that the error for the cross-validation and test sets after 1000 epochs is the same. The training set error is much lower. The asymptotic error for the CV and test sets is determined by the network size, the starting point, and the sampling density.	85

Figure 5.3 The output of a network for both the training and test sets. The test set shows an increase in error relative to the training set. This error can most easily be reduced by increasing the size of the training set to more than 200 samples. 87

Figure 5.4 Training with an undersampled function results in poor generalization and a high error for the test set. This figure was generated with only 100 range samples between 0.1 and 1.1 km. 88

Figure 5.5 The mean-squared error vs. the number of inputs shows that the optimal number of inputs is approximately 30. This corresponds to a mse of 0.0115 for the test set. 88

Figure 5.6 The mean-squared error vs. the number of hidden neurons shows that the optimal number of neurons is 8. This is a mse for the test set of 0.0136. 89

Figure 5.7 The mean-squared error vs. the number of hidden layer neurons in the presence of noise. Each point is the expected value of the final mse for 100 different starting weights. The noise has severely degraded the performance. 91

Figure 5.8 The mean-squared error vs. the number of inputs (principal components) in the presence of noise. Each point is the expected value of the final mse for 100 different starting weights. The noise has severely degraded the performance. 91

Figure 5.9 The output of an undertrained and slightly trained network overplotted with the target. The undertrained network has an output equal to the mean of the targets and a mse of 0.338. The slightly trained network has begun to approximate the targets and has a reduced error of 0.308. 95

Figure 5.10 The output of the neural classifier before and after the application of the winner-takes-all function. 99

Figure 5.11 The classification process with a two-element feature vector and three classes. (a) Initial feature vectors and classification. (b) Decision boundaries after training on the prototype feature vectors. (c) Decision boundaries after training on the complete data set. (d) Decision boundaries after training on the noisy data. 101

Figure 5.12 The output of the backpropagation neural classifier for the experimental data show that the bottom is likely a mixture of sand and gravel. The final classification is gravel. 102

Figure 5.13 The mean-squared error of the test set plotted vs. the spread with and without smoothing of the input with a logarithm. The compressed data achieves a lower error in the test set. 103

Figure 5.14 The test set mse is minimized for a spread of 5. The nonlinear changes in the spread indicate that the parameter space is poorly sampled with only 250 centres. 104

Figure 5.15 The test set mse is minimized for a spread of 10. The smoothing of the mse curve indicates that the parameter space is well sampled with 500 centres. 105

Figure 5.16 Scatterplot of the first two feature vectors for the first two classes. The small points correspond to the clay class and the larger points are for the silt class. The distribution of points within each class are not Gaussian. 108

Figure 5.17 The output of the neural RBF classifier for the experimental data is more ambiguous than with the BPNN. The probability of misclassification is 18.8%. The final classification is sand. 109

Figure 5.18 The ambiguity surfaces for the environment predicted by the BPNN. 112

Figure 5.19 The ambiguity surfaces for the environment predicted by the RBFNN. 113

List of Tables

Table 4.1	Relative Hydrophone Locations	56
Table 4.2	Array Element Parameterization	57
Table 4.3	Bathymetry Data	59
Table 4.4	Sound Speed Profile at the Array	60
Table 4.5	Parameterized Sound Speed Profile and Range of Uncertainty .	61
Table 4.6	Nominal Bottom Parameter Values and Range of Uncertainty .	62
Table 4.7	Doppler Shift Due to Source Movement at 3.2 m/s	63
Table 4.8	Spectral Estimation Parameters	70
Table 4.9	Nominal <i>a priori</i> Parameter Values and Range of Uncertainty .	72
Table 4.10	Relative Importance of Parameters	74
Table 4.11	Result of Grid Search of Group I	76
Table 5.1	Initial Training Parameters and Conventions	84
Table 5.2	Estimation Results (BPNN): Range, Group II–IV	94
Table 5.3	Estimation Results (BPNN): Group II–IV	96
Table 5.4	Bottom Types and Nominal Parameter Values	98
Table 5.5	Estimation Results (RBFNN): Range, Group II–IV	106
Table 5.6	Estimation Results (RBFNN): Group II–IV	107

Notation

p	lower case italics, scalar
P	upper case italics, scalar upper bound (e.g., $m = 1, 2, \dots, M$) or a transfer function (e.g., $y = H(x)$)
\mathbf{p}	lower case bold, vector
\mathbf{P}	upper case bold, matrix
p^*	complex conjugate
\mathbf{M}^T	vector or matrix transpose
\mathbf{M}^\dagger	vector or matrix complex conjugate transpose
\mathbf{M}^+	Moore-Penrose pseudoinverse
ANN	artificial neural network
BPNN	backpropagation artificial neural network
BP	backpropagation algorithm
RBFNN	radial basis function neural network
MFP	matched-field processing
MFI	matched-field inversion
B	Bartlett power
B_{inc}	incoherent average Bartlett power
B_{sup}	Bartlett power calculated with the super-vector of Michalopoulou
RF	radio frequency
GPS	global positioning system
CSM	cross-spectral matrix
λ	wavelength
FFT	fast Fourier transform
I	importance parameter
SVD	singular-value decomposition
PCA	principal components analysis
SD	steepest-descent optimization
CV	cross-validation
mse	mean-squared error

TL	transmission loss
cps/cpb	compressional speed in the sediment/basement
ps/pb	density in the sediment/basement
aps/apb	compressional-wave attenuation in the sediment/basement
ass/asb	shear-wave attenuation in the sediment/basement
t	thickness of the sediment
tlt	array tilt
ah	array height above the ocean bottom
wd	water depth
sd	source depth
sr	source range

Acknowledgement

I would like to thank the following people: Dr. Antoniou for supporting me through my thesis work, and for providing his most thorough comments; Dr. Chapman for introducing me to the topic of ocean acoustics, and following it through with excellent discussions and help; Dr. Dimopoulos for starting me out with neural networks; Dr. Lu for showing me that even the most complex things start out simple; Reza Dizaji for his energetic answers to my questions; Dr. Garry Heard for help with the DREA data set; Mark Fallat for discussions and help with the data; John Dorocicz for proving that software administrators can be friendly; Catherine Chang for helping me finish remotely; Vicky Smith for her help with all of the details; Ron Kessel for his help with SAFARI; my parents for their support; and Ashleen Downes for helping me with everything.

Chapter 1

Introduction

Acoustic wave propagation continues to be the most practical form of remote sensing in the ocean. Electromagnetic waves are significantly attenuated after traveling only a few hundred meters in the water, whereas, at very low frequencies, an acoustic wave may only be subjected to 0.004 dB/km of attenuation in sea water [1]. This makes detection, classification, and identification by acoustic methods possible over long ranges [2].

In shallow water the geoacoustic properties of the sea bottom are required in order to accurately model acoustic wave propagation. Acoustic modeling of shallow water sites is useful for many military applications such as underwater communications, source localization and source tracking [2]. Civilian applications such as resource exploration and geotechnical assessment are also making use of advanced modeling techniques [3].

The parameters which affect sound wave propagation in fluid media are compressional wave speed and attenuation, and density. In viscoelastic media (lossy solids) the addition of shear wave speed and corresponding attenuation are required. A complete geoacoustic model also includes the spatial dependence of these properties for a given experiment [1].

Geoacoustic properties can be determined using core samples and other direct measurements [2]. In general these methods are costly and time-consuming. Recently methods have been proposed which extract these bottom parameters using a model inversion scheme [4] [5]. These techniques involve indirectly determining the bottom parameters from a measured acoustic field. Typically a sound source and one or more receivers are placed in the water. The transmitted signal travels through the ocean environment and is transformed by it before being recorded at the receiver.

Consequently, the received signal contains specific information about the medium through which it has traveled.

The acoustic field in a waveguide can be described by $\mathbf{p} = H(\boldsymbol{\xi})$, where H is the forward model for calculating the field, \mathbf{p} is the theoretical acoustic field, and $\boldsymbol{\xi}$ is a vector of the model parameters which defines the environment. These model parameters are sometimes called the *geoacoustic model* or the *environment* since they form the basis for the numerical model of the environment. In this thesis we will use the term *model* to refer to the numerical method used to calculate the acoustic field, and *environment* will be reserved for the geoacoustic model parameters. The inversion problem is defined as: *find an estimate of the model parameters $\boldsymbol{\xi}$ given some measurement(s) of the acoustic field \mathbf{p} and a forward model H .* This can be extended to include the requirement that the accuracy of the parameter estimates be found as well.

Geoacoustic inversion initially evolved from the source localization problem. In source localization an array of hydrophones is used to record a signal from a source at an unknown location. In shallow water, conventional plane-wave beamforming cannot resolve the source accurately because of the multipath propagation of the signal [1]. In the absence of multipaths the plane-wave beamformer is only able of resolving the direction of arrival of a signal. A model-based method was proposed by Hinich [6] and Bucker [7] which involves comparing the received field with a field modeled at each possible source location. The linear correlation of the two fields is a measure of their similarity; in the two-dimensional range-depth case the linear correlation gives rise to what is known as an ambiguity surface. The peak of the ambiguity surface (maximum correlation) generally occurs at the true source location. This technique is termed matched-field processing (MFP). Several different matched-field processors have been proposed which offer improvements such as noise suppression and localization in unknown or poorly known environments. These processors, as well as many other issues surrounding matched-field processing, are summarized by Baggeroer et al. [8] and by Tolstoy [4].

One of the first problems that was encountered in MFP was that the propagation environment needed to be known exactly in order to localize the source. *Mismatch* in the environment caused the peak location to shift or to be lost entirely due to

the increasing sidelobes. In 1991 Collins and Kuperman proposed a technique to address this problem which they called *focalization* [9]. This method involves varying not only the range and the depth of the replica fields but also the parameters of the water column and sea bottom. They immediately saw it as a multi-dimensional optimization problem.

The situation then became one of efficiently finding the optimal set of parameters. In realistic cases the ambiguity hypersurface is multimodal, meaning that there are many local optima which are good, but are not guaranteed to be the global optimum (true peak). Various techniques based on guided Monte Carlo searches have since been proposed and implemented with varying degrees of success [5]. One problem that occurs with real data is that the global minimum may not correspond to the true parameter values due to noise or other factors. The limiting factor or bottleneck in virtually all of these methods is that the forward models used to generate the candidate fields for matching are very slow. Methods which minimize the number of function evaluations are the most preferred. A summary of the current state-of-the-art in this field is given in the special issue on geoacoustic inversion in the *Journal of Computational Acoustics* [10]. The majority of the techniques involve either genetic algorithms [11] or simulated annealing [3]. These methods are often used in concert with local optimization techniques such as the downhill simplex algorithm or variants on the conjugate-direction method. These local optimization techniques are preferred over gradient-descent and Newton-based methods because they do not require that the gradient or Hessian of the cost function be calculated; analytic expressions for the gradient of the acoustic field with respect to specific parameters are not generally available for practical acoustic models.

Determining the parameters by finding the maximum of a matching function has come to be known as matched-field inversion (MFI). One technique which stands apart from these methods is the inversion of the data using a backpropagation neural network. This method was proposed by Stephan et al. [12] as an alternative to matched-field methods. In 1996 Jesus and Caiti [13] proposed a radial basis function (RBF) based inversion which compares favorably with genetic algorithms. To date these techniques have only been applied to synthetic data sets.

Neural inversion involves the supervised training of a neural network with input-

output vector pairs. In the case of Stephan et al. the input vector consists of the received acoustic power at an array of sensors. The output is the environment vector ξ . A large number of fields are calculated from randomly chosen environment vectors, and the network is trained with this data using the backpropagation weight update algorithm. The final step of this inversion process is to use the trained network with the experimental data. Neural inversion with backpropagation neural networks does not eliminate iterative optimization. Instead of optimizing the model parameters directly with respect to an objective function based on the data, a set of inverse model weights are optimized to find a network which performs a nonlinear regression of the data. The advantage, as stated by Stephan et. al., is that typically only 1/20th the number of forward models need to be calculated to obtain a solution. Exact radial basis function neural networks avoid iteration altogether as their training consists of a matrix inversion.

In this thesis we apply the neural inversion technique to experimental data obtained from the TRIAL SABLE experiment. Both backpropagation and radial basis function neural networks are investigated. The goal is to obtain a set of environmental parameters which are more accurate than those obtained in previous studies [14]. The success of the inversion is quantified using an objective function which compares the experimental data with the modeled data as in MFP. In addition to the environmental parameter estimation, neural networks are designed to classify the measured data with respect to bottom type. The neural classifiers distinguish between different ocean bottom types and effectively avoid the problems of correlated parameters which occur in some parameter estimation methods. The training of both the parameter estimation and classification networks is done with a set of input/output pairs generated for the environment using an acoustic model known as ORCA.

In Chapter 2 the concepts of ocean acoustic modeling and matched-field parameter estimation are introduced. The normal-mode solution of the wave equation is examined in detail for a simple environment. The ORCA normal mode code for geoacoustic modeling is described and ORCA's suitability for use with parameter estimation algorithms is discussed. ORCA is used to illustrate the technique of matched-field processing (MFP) for source localization in shallow water. An example with mismatch in the water depth is given to show the importance of an accurate set of environmental

parameters. Matched-field inversion (MFI) for environment parameter estimation is introduced as a multi-parameter global optimization problem, which is then cast as one method of inverse system identification.

Chapter 3 deals with inverse system identification. The general inverse system identification problem is posed and neural inversion is offered as a specific solution approach. Two types of artificial neural networks, which can be used for parameter estimation or data classification, are described and their training algorithms are detailed. The role of the pre-processing of acoustic data for inclusion in the neural inversion scheme is discussed and an effective scheme for pre-processing both the experimental data and the output of the geoacoustic model is developed. Some refinements and extensions of the classical neural network training algorithms are introduced. The limitations of artificial neural networks subject to the identifiability conditions of a system closes the chapter.

The details of the TRIAL SABLE acoustic experiment are given in Chapter 4. The experiment itself, and all of the *a priori* information available for the site, are summarized and the environment is parameterized for use with ORCA. The estimation of the spectral components of the data is described and the results are compared with the output of ORCA using the nominal set of environmental parameters. A sensitivity study was performed using ORCA to determine which parameters are important for the calculation of the field. An *ad hoc* grid search was performed to quickly find the most important of these parameters and the estimated values are tabulated. The remaining unknown parameters are the focus of the inversion and classification in Chapter 5.

In Chapter 5 the results of the neural inversion of synthetic and experimental data are presented. Initially a backpropagation network was trained to estimate the range since this parameter is known *a priori* and is used to demonstrate the effectiveness of the neural inversion technique. The network size, training parameters and pre-processing steps which minimize the network error were determined for the range-finding case. A new network was designed to estimate the bottom parameters using the training information gleaned from the range-finding study and the results are presented and discussed. Bottom classification was performed using a backpropagation neural classifier. The bottom classification results are discussed and compared with

the parameter estimation results. The range-finding, bottom parameter estimation, and bottom classification processes were then repeated using a radial basis function neural network in place of the backpropagation network. The final bottom parameter estimates from both networks were then used to perform matched-field source localization.

Chapter 6 contains a discussion of the final results and some conclusions about the applicability of the neural inversion method to the problem of geoacoustic parameter estimation. The relative merits of the two different network types are discussed. The sources of error in the TRIAL SABLE experiment inversion and bottom classification are detailed followed by suggestions for future work in the area of neural inversion.

Chapter 2

Ocean Acoustics Introduction

Ocean acoustic experiments generally require large-scale equipment and manpower. The equipment is an assortment of sound projectors (underwater speakers), hydrophone receivers (underwater microphones), sensors to measure the environment, data acquisition equipment and a ship. A typical acoustic experiment is illustrated in Figure 2.1. In this case there is a single ship towing a sound source. The acoustic field is measured by a vertical array of receivers moored to the sea floor. The received signal is sampled and sent back to the ship using a radio frequency (RF) link. At the ship the data are processed and stored for later examination.

In Figure 2.1 the ocean environment is assumed to be range-independent, thus the properties that control the acoustic wave propagation are constant in range, and vary only with depth. In addition, the properties are assumed to be circularly symmetric so that the field is independent of the bearing of the ship relative to the array. This is represented by the parallel-plate waveguide model shown in Figure 2.2. The waveguide boundaries are stacked layers with constant acoustic properties in each layer. The sound from the source travels through the water, sediment, and basement, undergoing reflections and refractions at each of the interfaces before being received by the array. In shallow water the multipath propagation results in a highly structured pressure field. The exact structure of this field is dependent upon the acoustic parameters of the water column and the ocean bottom. This is in contrast to the simplest case of sound propagation in an unbounded homogeneous medium in which spherical wavefronts propagate out from the source. In the far-field this can be approximated locally as plane-wave propagation.

In this chapter we briefly describe the derivation of the differential equation for wave propagation in a fluid. The boundary conditions for this equation are then

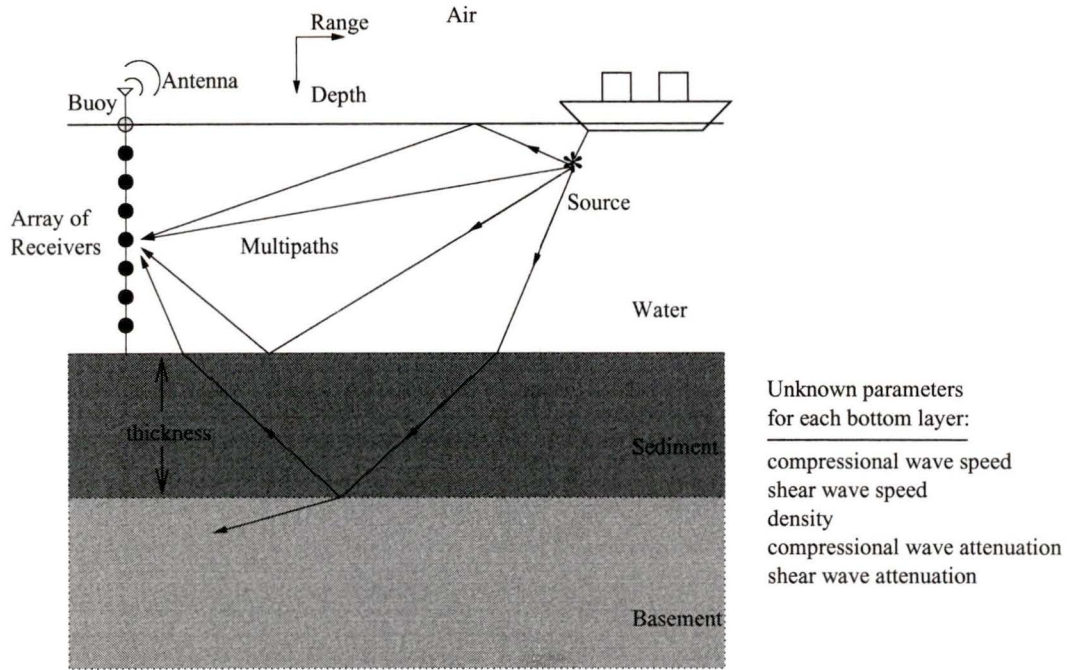


Figure 2.1. A typical ocean acoustic experiment.

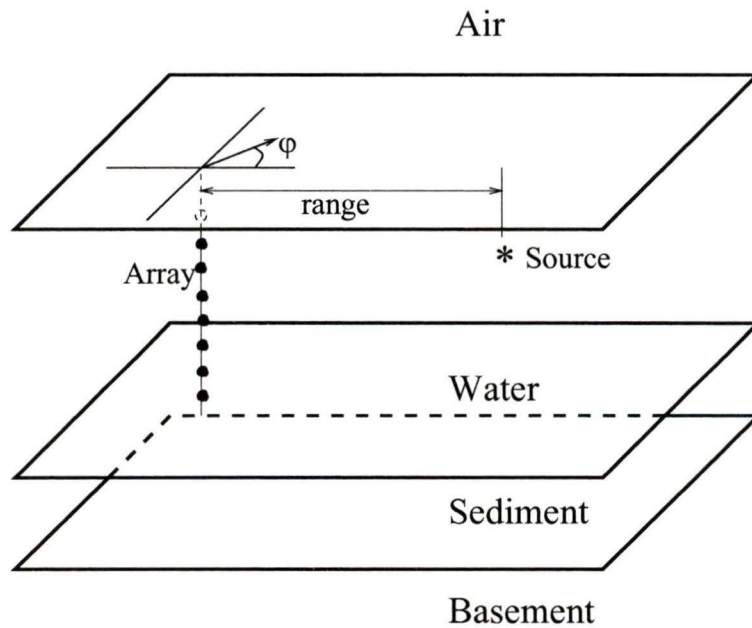


Figure 2.2. A circularly symmetric parallel plate waveguide has constant properties over any change in bearing or range. The properties vary only in depth.

described and the numerical solutions for realistic boundary conditions are discussed. The normal mode solution provided by the ORCA software is explained in detail. ORCA is used to generate transmission loss plots for a typical waveguide. The concept of matched-field processing is introduced with an example using simulated data. Finally mismatch and the matched-field inversion problem are illustrated with an example.

2.1 Acoustic Modeling

This section follows the notation and derivation of Jensen et al. [1]. The reader is referred to this book for a detailed account of computational ocean acoustics.

2.1.1 The Wave Equation

The goal of propagation modeling is to predict sound field measurements made in the ocean. Each hydrophone measures the pressure at a single point in the sound field. In a well designed experiment the hydrophone measurement will be primarily dependent on the sound source rather than noise, thus the mathematical relationship between the source signal and the received signal is the model of what will be seen in the actual experiment. This model is sometimes called the *forward model*.

The wave equation for a fluid is derived from the equation for conservation of mass

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot \rho \mathbf{v}, \quad (2.1)$$

Newton's Second Law

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\frac{1}{\rho} \nabla p(\rho), \quad (2.2)$$

and the adiabatic equation of state

$$p = p_0 + \rho' \left[\frac{\partial p}{\partial \rho} \right]_S + \frac{1}{2} (\rho')^2 \left[\frac{\partial^2 p}{\partial \rho^2} \right]_S + \dots \quad (2.3)$$

where ρ is the density, \mathbf{v} is the particle velocity, p is the pressure, p_0 is the quiescent pressure, and ρ' is a small perturbation in the density. Subscript S denotes that the partial derivatives are taken at constant entropy.

The linearized form of the wave equation is derived by using only the first-order terms in the above equations, i.e.,

$$\frac{\partial \rho'}{\partial t} = -\rho_0 \nabla \cdot \mathbf{v} \quad (2.4)$$

$$\frac{\partial \mathbf{v}}{\partial t} = -\frac{1}{\rho_0} \nabla p'(\rho) \quad (2.5)$$

$$p' = \rho' c^2 \quad (2.6)$$

where ρ_0 is the quiescent density and we have defined the speed of sound in an ideal fluid to be

$$c^2 \equiv \left[\frac{\partial p}{\partial \rho} \right]_s \quad (2.7)$$

If we assume that the speed of sound and the quiescent density are independent of time, then the above equations can be used to obtain the wave equation for pressure which is

$$\rho \nabla \cdot \left(\frac{1}{\rho} \nabla p \right) - \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} = 0 \quad (2.8)$$

Furthermore, if the density is constant in space, then the standard form of the wave equation becomes

$$\nabla^2 p - \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} = 0 \quad (2.9)$$

Equation 2.9 is a homogeneous differential equation; in the presence of a sound source we must include a *forcing function* $f(\mathbf{r}, t)$ to obtain the inhomogeneous equation

$$\nabla^2 p(\mathbf{r}, t) - \frac{1}{c^2(\mathbf{r})} \frac{\partial^2 p(\mathbf{r}, t)}{\partial t^2} = f(\mathbf{r}, t) \quad (2.10)$$

where \mathbf{r} in cylindrical coordinates is the location of the source in range, depth, and bearing. Typically the forcing function is taken to be a *Dirac delta function* positioned at \mathbf{r}_0 ($f(\mathbf{r}, t) = \delta(\mathbf{r} - \mathbf{r}_0)$). This is a good approximation for practical point sources as long as the receiver is more than a few meters away.

Equation 2.10 is a four-dimensional differential equation. Three of the dimensions are in space and the fourth is time. The solution of the equation can be simplified by eliminating the dimension of time through the use of a Fourier transform

$$f(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \quad (2.11)$$

where $\omega = 2\pi f$. Equation 2.9 coupled with Equation 2.11 leads to what is known as the *Helmholtz equation*

$$\left[\nabla^2 + k^2(\mathbf{r})\right] p(\mathbf{r}, \omega) = f(\mathbf{r}, \omega) \quad (2.12)$$

where $\omega = 2\pi f$ and

$$k(\mathbf{r}) = \frac{\omega}{c(\mathbf{r})} \quad (2.13)$$

is the wavenumber at the frequency ω

Equation 2.12 evaluated at a single frequency ω and including the Dirac delta forcing function results in the inhomogeneous Helmholtz equation

$$\left[\nabla^2 + k^2(\mathbf{r})\right] p(\mathbf{r}) = -4\pi\delta(\mathbf{r} - \mathbf{r}_0) \quad (2.14)$$

The Helmholtz equation is a three-dimensional elliptic partial differential equation. The solution to this equation can be analytic or numerical depending on the complexity of the boundary conditions which are developed from the environmental properties. A combination of the analytic and numerical solution is used for realistic problems to keep the computation time to a minimum and the accuracy high.

2.1.2 Numerical Solutions

There are a number of different methods for the practical solution of the wave equation. The most commonly used methods are

- Ray theory
- Wavenumber integration
- Parabolic equation
- Finite differences/finite elements
- Normal modes

Each of these methods has certain properties which make it suitable for different applications. Details of all five methods can be found in [1].

Ray theory is perhaps the most intuitive of the methods. Ray theory originated from optics and is easily visualized in the form of ray paths, or ray tracing. Ray models are derived by assuming an infinite frequency source and dividing the medium into a large number of very thin layers. At each layer the ray is reflected and/or refracted.

The reflection coefficients are determined by the properties of each layer. This method is useful for broadband or time-domain modeling since it is very fast to compute. Ray theory breaks down at low frequencies and is not generally useful when high accuracy is required. GAMARAY is an example of an available ray tracing code [15].

Wavenumber integration is often called the *exact solution* since it evaluates the integrals of the depth-separated Helmholtz equation directly. This method is narrow-band (single frequency) because of the use of the Helmholtz equation. Wavenumber integration models are more difficult for the neophyte because they require the specification of several integration limits which are best determined by an experienced user. This method is range-independent, although coupled range-dependent solutions are available. The main disadvantage of wavenumber integration is its long computation time and complexity of use. SAFARI and OASES are examples of available wavenumber integration codes [16].

The parabolic equation method is a parabolic approximation to the elliptic wave equation. This method is gaining in popularity because it can more easily accommodate range-dependent environments. Solutions for both narrowband and wideband cases exist which include range dependence and elastic media. This method can be computationally efficient. The main disadvantage of the parabolic equation method is that it requires significant expertise on the part of the user.

Finite differences and finite elements are most often used when backscattering from rough surfaces needs to be taken into account. They are also useful for environments with complex geometries. These methods are very computationally expensive; however, they can be formulated for virtually any conceivable situation and are thus required in situations where other, faster, models are not sufficiently precise.

The normal mode solution uses the same integral formulation as wavenumber integration. The solution is obtained by representing the integration as a sum of residues. The normal mode solution is a finite sum of depth dependent eigenfunctions analogous to the vibration modes of a string [1]. The modes are found subject to the boundary conditions imposed on the system. Normal modes are relatively fast to compute and can be adapted to accommodate both compressional and shear waves. The main disadvantage is that they are range independent and narrowband. Range dependence can be taken into account by using the method of coupled or adi-

abatic normal modes. Range-dependent formulations can be accurate but do require significantly more computation time.

We chose the normal mode code ORCA [17] as the primary model for this work because of its ease of use, versatility and computational speed. The next section describes the derivation of a simple normal mode solution and then goes on to describe the specifics of the ORCA model.

2.1.3 The Normal Mode Solution

In this section we solve the homogeneous Helmholtz equation in a depth-dependent waveguide, for a rigid-bottom pressure-release surface using the method of separation of variables. It is useful at this point to adopt a coordinate system convention. For most shallow water propagation problems the wavefronts from a source change from spherical spreading to cylindrical spreading very quickly. When there is little variation in the water depth, or the ocean properties in range and bearing, it is advantageous to use the cylindrical coordinate system. For a cylindrical system the spatial vector is $\mathbf{r} = (r, \varphi, z)$ and the Laplace operator for the Helmholtz equation is

$$\nabla^2 = \frac{1}{r} \frac{\partial}{\partial r} r \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \varphi^2} + \frac{\partial^2}{\partial z^2} \quad (2.15)$$

Combining Equation 2.15 with Equation 2.14, and making the assumption that sound speed and density depend only on depth, we obtain the two-dimensional Helmholtz equation for a point source in cylindrical geometry

$$\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial p}{\partial r} \right) + \rho(z) \frac{\partial}{\partial z} \left[\frac{1}{\rho(z)} \frac{\partial p}{\partial z} \right] + \frac{\omega^2}{c^2(z)} p = -\frac{\delta(r) \delta(z - z_s)}{2\pi r} \quad (2.16)$$

The details of this step may be found in [1]. Substituting $p(r, z) = \Phi(r)\Psi(z)$ into the unforced form of equation 2.16 and then dividing by $\Phi(r)\Psi(z)$, we have effectively separated the variables of r and z

$$\frac{1}{\Phi} \left[\frac{1}{r} \frac{d}{dr} \left(r \frac{d\Phi}{dr} \right) \right] + \frac{1}{\Psi} \left[\rho(z) \frac{d}{dz} \left(\frac{1}{\rho(z)} \frac{d\Psi}{dz} \right) \frac{\omega^2}{c^2(z)} \Psi \right] = 0 \quad (2.17)$$

The condition for this equation to hold is that the second term must equal a separation constant k_{rm}^2 . This condition results in what is known as the *modal equation*

$$\rho(z) \frac{d}{dz} \left[\frac{1}{\rho(z)} \frac{d\Psi_m(z)}{dz} \right] + \left[\frac{\omega^2}{c^2(z)} - k_{rm}^2 \right] \Psi_m(z) = 0 \quad (2.18)$$

The boundary conditions for a pressure-release surface and a rigid bottom located at depth D are

$$\Psi = 0, \quad \left. \frac{d\Psi}{dz} \right|_{z=D} = 0 \quad (2.19)$$

The modal equation is in the form of a *Sturm-Liouville* eigenvalue problem with homogeneous boundary conditions; the properties of this problem are well known [18]. The modal equation has an infinite number of eigenfunctions $\Psi_m(z)$. Each eigenfunction has a distinct eigenvalue in the form of a horizontal propagation constant k_{rm} . The eigenvalues are real and ordered as $k_{r1}^2 > k_{r2}^2 > \dots > k_{rm}^2 > \dots$. The m th mode has m zeros on the interval $[0, D]$. An important feature of the modes is that they are *orthogonal* and thus satisfy the relation

$$\int_0^D \frac{\Psi_m(z)\Psi_n(z)}{\rho(z)} dz = \begin{cases} 0 & n \neq m \\ 1 & n = m \end{cases} \quad (2.20)$$

These modes form a complete set, thus any continuous function can be represented as an infinite sum of normal modes. The pressure of Equation 2.18 can be written as

$$p(r, z) = \sum_{m=1}^{\infty} \Phi_m(r)\Psi_m(z) \quad (2.21)$$

We can substitute Equation 2.21 into Equation 2.16 to obtain

$$\begin{aligned} \sum_{m=1}^{\infty} \left\{ \frac{1}{r} \frac{d}{dr} \left(r \frac{d\Phi_m(r)}{dr} \right) \Psi_m(z) + \Phi_m(r) \left[\rho(z) \frac{d}{dz} \left(\frac{1}{\rho(z)} \frac{d\Psi_m(z)}{dz} \right) + \frac{\omega^2}{c^2(z)} \Psi_m(z) \right] \right\} \\ = -\frac{\delta(r)\delta(z-z_s)}{2\pi r} \end{aligned} \quad (2.22)$$

The modal equation is then applied to yield

$$\sum_{m=1}^{\infty} \left\{ \frac{1}{r} \frac{d}{dr} \left[r \frac{d\Phi_m(r)}{dr} \right] \Psi_m(z) + k_{rm}^2 \Phi_m(r) \Psi_m(z) \right\} = -\frac{\delta(r)\delta(z-z_s)}{2\pi r} \quad (2.23)$$

Applying the following integration

$$\int_0^D (\cdot) \frac{\Psi_n(z)}{\rho(z)} dz \quad (2.24)$$

to Equation 2.23 results in the standard Bessel equation

$$\frac{1}{r} \frac{d}{dr} \left[r \frac{d\Phi_n(r)}{dr} \right] + k_{rn}^2 \Phi_n(r) = -\frac{\delta(r)\Psi_n(z_s)}{2\pi r \rho(z_s)} \quad (2.25)$$

Since the time dependence is of the form $\exp(-i\omega t)$ we can use the Hankel function of the first kind to form a solution where

$$\Phi_n(r) = \frac{i}{4\rho(z_s)} \Psi_n(z_s) H_0^{(1)}(k_{rn}r) \quad (2.26)$$

The asymptotic approximation to the Hankel function is

$$H_0^{(1)}(k_{rn}r) \approx \frac{e^{ik_{rn}r}}{\sqrt{rk_{rn}}} \quad \text{for } k_{rn}r \gg 1 \quad (2.27)$$

The final solution for the pressure is then

$$p(r, z) \approx \frac{i}{\rho(z_s)\sqrt{8\pi r}} e^{-i\pi/4} \sum_{m=1}^{\infty} \Psi_m(z_s) \Psi_m(z) \frac{e^{ik_{rm}r}}{\sqrt{k_{rm}}} \quad (2.28)$$

This equation represents the pressure due to a point source in cylindrical geometry. In order to solve for the pressure, the normal modes Ψ_m must be known. They are found by applying the appropriate boundary conditions. For a rigid-bottom, pressure-release surface, and an isovelocity medium (see Equation 2.19), they are simply sinusoids in depth and can be found analytically [1] as

$$\Psi_m(z) = \sqrt{\frac{2\rho}{D}} \sin(k_{zm}z) \quad (2.29)$$

The pressure as a function of range and depth becomes

$$p(r, z) = \frac{i}{D\sqrt{2\pi r}} e^{-i\pi/4} \sum_{m=1}^M \sin(k_{zm}z_s) \sin(k_{zm}z) \frac{e^{ik_{rm}r}}{\sqrt{k_{rm}}} \quad (2.30)$$

This equation can be further simplified if we write it as a transmission loss which is

$$\text{TL} = -10 \log_{10} \left| \frac{p(r, z)}{p_0(\sqrt{r^2 + (z - z_s)^2} = 1)} \right|^2 \quad (2.31)$$

where the pressure for the source in free space is

$$p_0(r) = \frac{e^{ik_0\sqrt{r^2 + (z - z_s)^2}}}{4\pi\sqrt{r^2 + (z - z_s)^2}} \quad (2.32)$$

Combining Equations 2.30- 2.32 gives the transmission loss in range and depth as

$$\text{TL}(r, z) = -10 \log_{10} \left| \frac{1}{D} \sqrt{\frac{8\pi}{r}} \sum_{m=1}^M \sin(k_{zm}z_s) \sin(k_{zm}z) \frac{e^{ik_{rm}r}}{\sqrt{k_{rm}}} \right|^2 \quad (2.33)$$

where

$$k_{zm} = \frac{\pi}{D} \left(m - \frac{1}{2} \right) \quad m = 1, 2, \dots, M \quad (2.34)$$

and

$$k_{rm} = \sqrt{\left(\frac{\omega}{c} \right)^2 - k_{zm}^2} \quad m = 1, 2, \dots, M \quad (2.35)$$

are the vertical and horizontal wavenumbers for mode m , respectively, where $\omega = 2\pi f$. The number of propagating modes can be estimated as [1]

$$M < \frac{2fD}{c} \quad (2.36)$$

The number of significant modes increases with either frequency or water depth, and decreases with the homogeneous sound speed.

From Equation 2.33 we see that the position of the source in depth z_s determines how much a particular mode is excited based on the first sinusoidal term. If the source is at a mode peak ($\sin(k_{zm}z_s) = 1$), that mode is excited to a maximum. If the source is at a mode null ($\sin(k_{zm}z_s) = 0$), that mode is not excited in the waveguide. The mode is similarly received at different levels by the hydrophone at z . Each mode has a different excitation in the waveguide, and is detected differently at specific depths.

As the modes propagate radially some decay strongly relative to the others, and the remainder propagate as outgoing waves. The decay is governed by whether or not the eigenvalues k_{rm} are imaginary. If k_{rm} is imaginary, then the mode decays rapidly in range and is called an evanescent mode. If it is real, then the result is a propagating mode. The upper bound M , in Equation 2.36, is the number of propagating modes, this equation is only valid in the far-field (greater than 10 water depths away from the source). At very short ranges the evanescent modes may be significant and this upper bound must be changed to include them.

Equation 2.33 shows that the relationship between the pressure and the source location is highly nonlinear even for this simple case. In fact the water depth and the sound speed also have a nonlinear relationship to the acoustic field. Other simple boundary conditions have analytic solutions but for the general range-independent, multi-layered, case the eigenfunctions and eigenvalues must be found numerically. This makes the relationship between the parameters and the field even more complicated.

2.1.4 ORCA

The ORCA normal mode model uses a general derivation similar to that of the previous section [19]. It is more sophisticated in the following respects:

- Compressional-wave attenuation is included by making the wavenumber complex.
- ORCA is applicable to viscoelastic (lossy solid) media, thus shear waves and shear wave attenuation are accounted for. Shear waves can be a significant source of loss in viscoelastic bottoms.
- ORCA solves the wave equation in an arbitrary number of layers. Variations in compressional speed, density, shear speed, and attenuation are included for each layer.
- ORCA is relatively easy to run for the novice user.

ORCA (version 1.6) is available as FORTRAN source code via anonymous FTP from ftp.arlut.utexas.edu. Once compiled the code works in the following manner. A number of input ASCII text files are prepared which contain the information about the environment and about the geometry of the experiment. The type of output file, and what outputs are required are also specified. ORCA is run and a set of output files is generated. Usually these output files are specified to contain the acoustic field at an array, and information about the modes in the waveguide. The input file format is outlined in the following section. A user's manual [17] that provides more information is available from the FTP site.

2.1.4.1 The SVP file

The `_svp` file contains the geoacoustic description of the environment. This file can have any name as long as it has the extension `_svp`. The parameters of the upper halfspace (above the water), the water, and the bottom are all specified. The upper halfspace is parameterized using the compressional and shear speed, the density, and the attenuations. Gradients in these parameters can also be accommodated.

The properties of the water column are the homogeneous density and attenuation, and the compressional speed profile. Each point in the sound speed profile is specified by a depth and a corresponding sound speed. Any number of points can be specified.

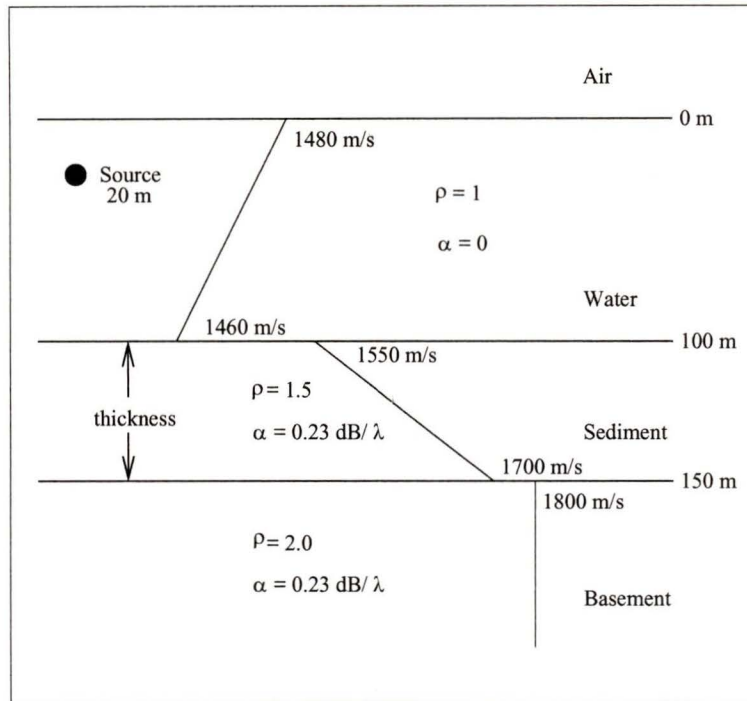


Figure 2.3. *The calibration environment for the matched-field inversion workshop.*

Each point indicates the bottom of a separate layer. ORCA automatically removes points that are not required for accurate calculation of the field in order to minimize computation time. The first point is the speed at the water surface; the last point is the speed at the bottom.

The bottom parameters and geometry are specified last. The bottom parameters are the compressional and shear speed, the density, and the attenuations. For each layer, except the bottom halfspace, a top and a bottom parameter value is specified. This is to allow for gradients in the parameters. For a homogeneous layer both values are the same. The distance of the layer interface from the ocean bottom is used to parameterize the layer thicknesses. The lower halfspace is specified by a single set of parameter values, though a gradient in the parameters within a layer is also supported. A sample `_svp` file can be found in Appendix A. This file corresponds to the environment specified in Figure 2.3.

2.1.4.2 The OPT file

The `_opt` file contains the specifications for the type of computations required, mode computation parameters, frequencies, output options, and source/receiver geometry. Other options are also supported for more advanced users.

The type of computation required refers to a number of outputs which can be generated. A most useful feature is that ORCA supports MATLAB `.mat` binary, HDF binary, or ASCII text files.

The mode computation parameters specify a number of different options available within the algorithm. We can choose to calculate only the modes which lie on the real k -axis. This simplification speeds up the computation considerably but is not valid for short ranges or for viscoelastic environments. In many cases we must find all of the significant modes in the complex k -plane. Other options specify the maximum number of modes to calculate or the thresholds for the modal attenuations. Some computational benefit can be gained by specifying these parameters but the parameters must be selected specifically for each environment, or there is a risk of error in the solution.

ORCA accepts any number of frequencies and almost any combination of source and receiver positions. When the total number of points requested becomes large, ORCA slows down because it begins using virtual memory. This situation can be avoided through the use of loops within the controlling function. The output of ORCA can be either the transmission loss or the complex pressure at any point in the waveguide. There are also options to output the mode functions and mode characteristics, such as phase speed and group velocity. A sample `_opt` file can be found in Appendix A.

2.1.4.3 Implementation and Testing

In this implementation, ORCA is used as a function call from MATLAB. This was done by writing a MATLAB function to generate the input data files. The function then runs ORCA, which in turn generates the output file. The output file is finally read into MATLAB. Writing data files for each run is somewhat inefficient but a field calculation for a single frequency and a single depth still took less than one second on a Sun ULTRA 1 with 64 MB of RAM running Solaris 2.4. This is 10 times faster than

computing the same field using SAFARI or OASES. The calibration of the ORCA model to confirm its validity is done in the following section.

2.2 Source Localization in a Waveguide

Now that we have a working model to calculate the acoustic fields it is instructive to use it to perform some simple simulations. One of the main applications is source localization. Matched-field processing for source localization can be simulated in the following manner. The model (ORCA) is used at a default source location to calculate the signal at an array for a fixed environment. These data are the ‘experimental’ data for the simulation. The simulated data are then used in a source localization routine to estimate the ‘actual’ source location. A number of different methods exist to locate a source in a waveguide. The most widely used technique is matched-field processing with the Bartlett processor.

2.2.1 Matched-Field Processing

As the name implies matched-field processing (MFP) finds the best source location by matching the field measured at an array with a set of candidate fields calculated for various source locations. The most common matching function is the linear correlator or Bartlett processor. It can be written as

$$B = \left| \sum_{n=1}^N p_n^* \hat{p}_n \right|^2 \quad (2.37)$$

where N is the number of sensors in the hydrophone array, \hat{p}_n is the measured complex pressure at sensor n , p_n is the modeled complex pressure, and $(\cdot)^*$ indicates complex conjugation.

Let us consider a field in which the source location in range and depth is unknown. The processor then becomes

$$B(r_s, z_s) = \left| \sum_{n=1}^N p_n^*(r_s, z_s) \hat{p}_n(\hat{r}, \hat{z}) \right|^2 \quad (2.38)$$

where (r_s, z_s) indicates the candidate source location and (\hat{r}, \hat{z}) represents the true source location. It follows, and can be shown, that this processor will display a peak when the condition $(r_s, z_s) = (\hat{r}, \hat{z})$ is met [4] [7] [8].

We can restate the problem using vector notation. Let $\hat{\mathbf{p}} = [p_1 \ p_2 \ \dots \ p_n]^T$ be the measured complex pressure vector at the array and \mathbf{p} be the modeled pressure vector. Furthermore define

$$\mathbf{R} = \langle \hat{\mathbf{p}} \hat{\mathbf{p}}^\dagger \rangle \quad (2.39)$$

to be the cross-spectral density matrix (CSM). The notation $\langle \cdot \rangle$ indicates the time average, and $(\cdot)^\dagger$ is the complex conjugate transpose operation. The quadratic form for the Bartlett power then becomes

$$B(r, z) = |\mathbf{p}^\dagger \mathbf{R} \mathbf{p}| \quad (2.40)$$

Matrix \mathbf{R} is Hermitian since it is formed by a sum of outer products [8]. Averaging is essential to reduce the variance of the cross-spectral estimate obtained from real data [20]. The calculation of \mathbf{R} in practice is discussed in Chapter 4.

It is generally necessary to normalize the Bartlett processor using

$$B(r, z) = \frac{|\mathbf{p}^\dagger \mathbf{R} \mathbf{p}|}{\|\mathbf{p}\|_2^2 \|\mathbf{R}\|_2} \quad (2.41)$$

where $\|\cdot\|_2$ denotes the L_2 -norm. When this normalization is used the maximum value for the Bartlett power is unity and the lowest possible value is zero.

Equation 2.41 is valid for a single frequency. The simplest way to include multiple frequencies is to sum up the Bartlett surfaces incoherently. We can write

$$B_{inc} = \frac{1}{F} \sum_{f=1}^F B_f \quad (2.42)$$

$$= \frac{1}{F} \sum_{f=1}^F \frac{|\mathbf{p}_f^\dagger \mathbf{R}_f \mathbf{p}_f|}{\|\mathbf{p}_f\|_2^2 \|\mathbf{R}_f\|_2} \quad (2.43)$$

where F is the number of frequencies, \mathbf{p}_f is the replica vector calculated for frequency f , and \mathbf{R}_f is the cross-spectral density matrix estimated from the data for the frequency f . The output of this processor is also on the interval $[0,1]$.

There are methods which attempt to use the relationship between the frequencies to aid in the localization [4] [21]. These processors differ from the incoherent Bartlett average of Equation 2.43 in that the correlations among the different fields at different frequencies are also used. One such processor uses the super-vector of Michalopoulou and Porter [22] which is defined as

$$\mathbf{p}_{sup} = [\mathbf{p}_1^T \ \mathbf{p}_2^T \ \dots \ \mathbf{p}_F^T]^T \quad (2.44)$$

The data covariance matrix is defined as

$$\mathbf{R}_{sup} = \langle \hat{\mathbf{p}}_{sup} \hat{\mathbf{p}}_{sup}^\dagger \rangle \quad (2.45)$$

and the coherent processor is obtained as

$$B_{sup} = \frac{|\mathbf{p}_{sup}^\dagger \mathbf{R}_{sup} \mathbf{p}_{sup}|}{\|\mathbf{p}_{sup}\|_2^2 \|\mathbf{R}_{sup}\|_2} \quad (2.46)$$

Michalopoulou and Porter [22] [23] show that this processor is more resistant to mismatch than the incoherent Bartlett processor in poorly known environments. One disadvantage to this processor is that the source spectrum must be known *a priori*. This problem can be avoided by normalizing each \mathbf{p}_f individually to remove the relative source levels.

Typically the range and depth of the source are found by calculating $B(r, z)$ for all r and z over some range of uncertainty for these parameters. This surface is then plotted and the maximum is found. This maximum, in the ideal case, corresponds to the true source location and has a Bartlett power of unity.

2.2.2 Simulation Example

Using the calibration environment from the matched-field processing workshop [10] and the 35 Hz, 1 km data, we can simulate matched-field source localization. The environment for the calibration data set is shown in Figure 2.3. Since no shear wave speeds are specified, a fluid bottom is indicated.

In order to verify that the ORCA model is in agreement with the calibration data set, we plot the transmission loss (TL) and the phase over the range for a 75 m deep receiver, and over the depth for a 1 km range receiver. Figure 2.4 shows that there is virtually no difference between the two models. The mean-squared percentage errors over range and depth are 0.4% and 1.5%, respectively. The pressures at numerous other frequencies were also calculated and similar errors were obtained.

Figure 2.5 shows the TL generated using ORCA from a range of 100 m to a range of 2100 m for the calibration environment. The TL is calculated in the water column and in the sediment and basement. Note that the signal penetrates the bottom and has significant interaction with the sediment-basement interface. This indicates

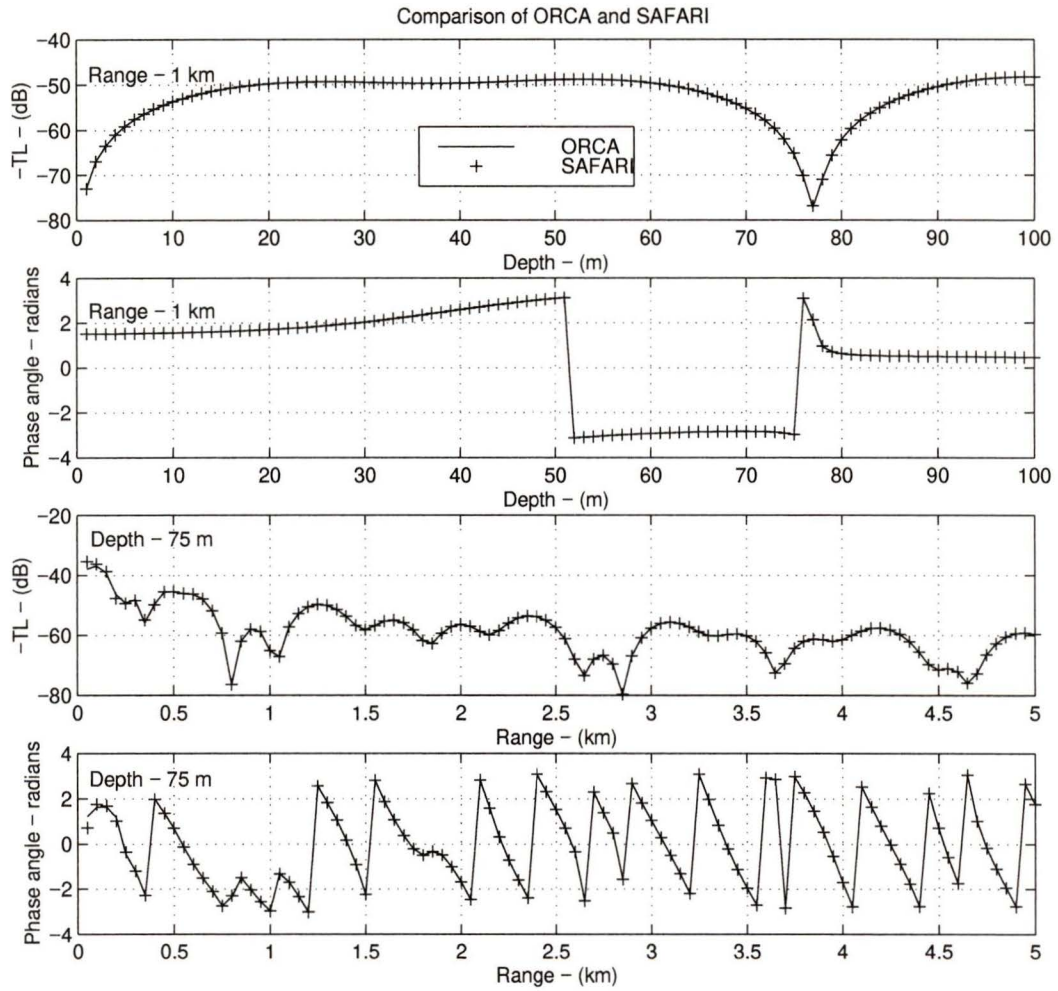


Figure 2.4. A comparison of the pressure calculated by ORCA and the calibration data set (SAFARI).

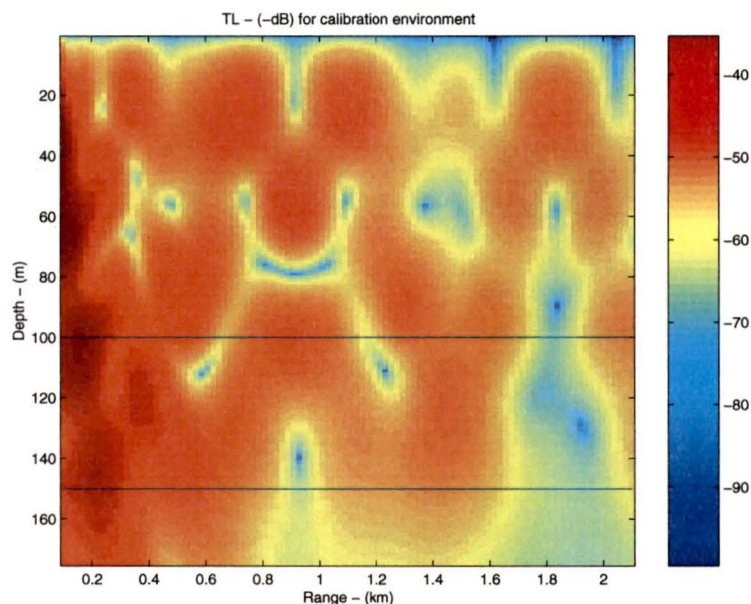


Figure 2.5. *The transmission loss in range and depth for a 35 Hz signal in the calibration environment. The horizontal lines indicate the water-sediment and sediment-basement interfaces. The sidebar indicates the loss in -dB.*

that the parameters of the bottom are significant for the field calculation. The TL shows the interference pattern typical of a shallow water environment. Matched-field processing depends upon the complexity of the field at a set of measurement points in the ocean. The more complex the field is, the more likely it is that this measured field will be unique for a given environment.

In order to use the quadratic form for the Bartlett processor we need to have a cross-spectral density matrix. The array is comprised of ten sensors spaced evenly between 10 and 100 m. The outer product of this vector with its complex conjugate transpose is used as the cross-spectral matrix. No time averaging needs to be included since the calibration data is noise-free.

The ambiguity surface shown in Figure 2.6 was calculated using Equation 2.41. The replica vectors were calculated using ORCA. A clear peak is evident at a range of 1 km and a depth of 20 m. This is the expected location. The Bartlett power at this point is 0.995. The peak value is not unity solely because of the mismatch between the ORCA and SAFARI models. If SAFARI had been used for the forward calculations then the Bartlett power would have been unity. The sidelobe level is approximately

0.7. This can be lowered through the inclusion of more sensors, or through the use of a so-called high-resolution processor [8]. The features of the ambiguity surface are similar to the interference pattern of the TL in Figure 2.5. Using higher frequencies results in a higher resolution for the solution, thus the peak (main lobe) would be sharpened but the sidelobes may remain at the same level.

Matched-field processing succeeds when the field is sampled such that the pressure at the array is unique for every candidate location. If the field is not unique then the sidelobe level at the false location would be the same as the Bartlett power at the true location. Thus the complexity of the field calculation works to aid the localization. If the field were simpler, such as in the plane-wave case, or for only a few propagating modes, there would be much more ambiguity in the localization. In fact, matched-field processing is simply a generalization of plane-wave beamforming which seeks to locate the source in terms of direction of arrival, but is unable to localize in more than one dimension due to the relative simplicity of the plane-wave field.

2.2.3 The Mismatch Problem

In the near ideal case presented above there is little ambiguity in the location of the source. Ambiguity can be reduced through diligent experiment design, but will almost always be present. For example if we only had two hydrophones and one frequency the sidelobe levels would greatly increase. If the array was situated within a null of the field there would also be trouble in localization. Of course, noise from other sources, or simply measurement noise, can be enough to make localization very difficult [4] [8] [9].

Another source of error is mismatch. Here, we divide the mismatch into two types: model mismatch and environment mismatch [4]. These errors manifest themselves as lower peak values, higher sidelobes, and erroneous peak location. Model mismatch occurs when the numerical model used as a solution to the wave equation is inadequate for the experimental configuration. This can occur, for instance, when using a range-independent model in a range-dependent situation. The error between the ORCA and SAFARI models is the model mismatch for this example. Environment mismatch is due to an inadequate knowledge of the waveguide geometry and properties. This can include the actual layer structure, the parameter values, and the array element location. Figure 2.7 shows the Bartlett ambiguity surface with a mismatch in water

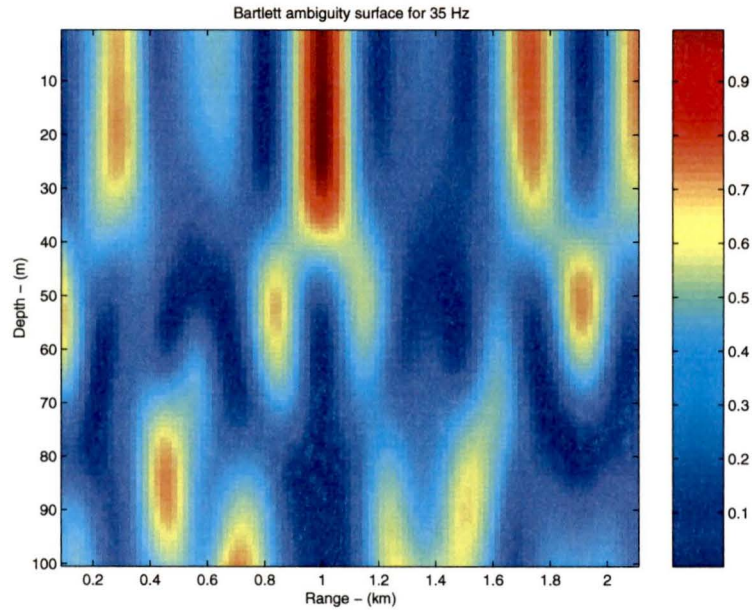


Figure 2.6. *The Bartlett ambiguity surface for a 35 Hz signal in the calibration environment. The source is correctly localized at a depth of 20 m and a range of 1 km.*

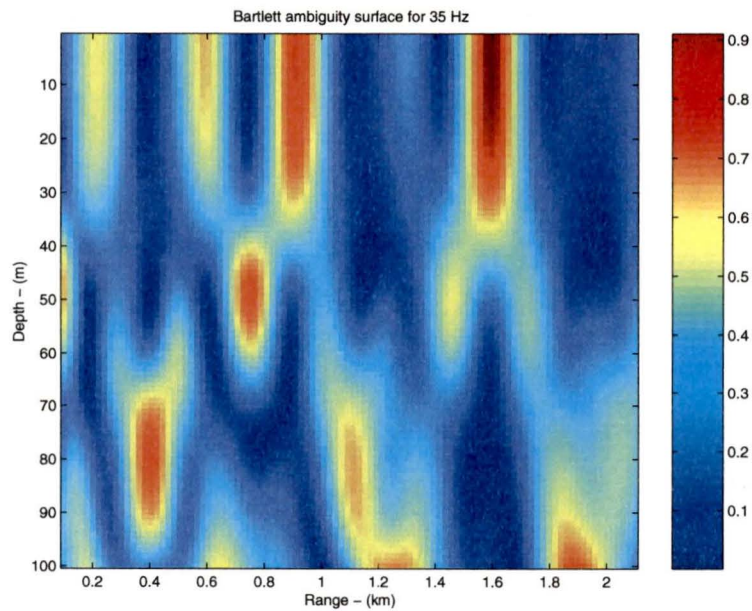


Figure 2.7. *The Bartlett ambiguity surface for a 35 Hz signal with a mismatch in the water depth. The actual water depth is 100 m, and the modeled depth is 90 m. The predicted source depth is ≈ 30 m, and the predicted range is ≈ 1.5 km.*

depth. The actual water depth is 100 m but the modeled water depth is 90 m. Here it is quite obvious that the peak has shifted from its original position and the peak value has lowered to 0.91. The sidelobe levels have also increased causing the localization to be lost entirely. The matched-field processor could be extended to include not only range and depth but also any parameter that is not known precisely. This idea of using the ambiguity function as a multi-dimensional cost function was first proposed by Collins and Kuperman [9], and has been successfully applied to matched-field inversion by Chapman and Ozard [3].

2.3 Matched Field Inversion

Matched-field inversion is a direct extension of matched-field processing for source localization. The acoustic model used for the inversion (e.g., ORCA) can be seen as a black box with the parameters as inputs and the acoustic field as the output. The parameters are here defined as any physically meaningful input to the model which must be specified to obtain an accurate result. The entire set of required parameters is termed the *environment*. Let us define the environment $\boldsymbol{\xi}$ as a vector of L scalar parameters ξ_l

$$\boldsymbol{\xi} = [\xi_1 \ \xi_2 \ \cdots \ \xi_L]^T \quad (2.47)$$

Matched-field inversion attempts to find the environment of the modeled field which provides the best match to an experimentally measured acoustic field. The process is generally cast as an optimization problem that combines matched-field processing with a global search method. The simplest and most commonly used cost function is the Bartlett processor

$$B(\boldsymbol{\xi}) = \frac{|\mathbf{p}^\dagger(\boldsymbol{\xi})\mathbf{R}(\hat{\boldsymbol{\xi}})\mathbf{p}(\boldsymbol{\xi})|}{\|\mathbf{p}(\boldsymbol{\xi})\|_2^2 \|\mathbf{R}(\hat{\boldsymbol{\xi}})\|_2} \quad (2.48)$$

where $\hat{\boldsymbol{\xi}}$ is the true environment. Equation 2.48 is maximized when $\boldsymbol{\xi} = \hat{\boldsymbol{\xi}}$, which only holds true if the parameters within $\boldsymbol{\xi}$ completely describe the experimental environment. Most optimization algorithms are set up to minimize functions. Thus the Bartlett *mismatch* is often defined as

$$M_B = 1 - B \quad (2.49)$$

$$= 1 - \frac{|\mathbf{p}^\dagger \mathbf{R} \mathbf{p}|}{\|\mathbf{p}\|_2^2 \|\mathbf{R}\|_2} \quad (2.50)$$

Finding the minimum of Equation 2.50 is one approach to the problem of system identification.

2.4 Conclusions

In this chapter we have seen that the mapping between the environmental parameter variables to the sampled pressure field is quite complex. The forward model ORCA is an efficient way to calculate the sampled pressure field for complicated environments. In a typical experiment many of the environment variables are difficult to measure directly. The problem then becomes one of determining the environment using the measured data and a forward model which can predict the data if the correct environment is known. In Chapter 3 the problem is posed within the paradigm of inverse system identification, and neural networks are proposed as a solution.

Chapter 3

Neural Inversion

Inverse theory is a methodology which attempts to extract useful information about a system from a set of measurements [24]. The measured observations are called the experimental data, and the useful information is the set of model parameters. These model parameters describe the environment and are used as the inputs to the forward model. The purpose of the forward model is to predict the experimental data using a general model structure and specific set of environmental parameters. The inverse problem is to predict the environmental parameters based on the forward model and a set of measurements. Neural inversion is the use of neural networks to extract useful information about the environmental parameters or the parameters themselves from measured data.

This chapter begins with a discussion of system identification and goes on to describe the role that neural networks can play. The specifics of each stage of an inversion are introduced. This is followed by a discussion of different types of neural networks, their design, and their applicability to practical inversion problems. The chapter concludes with a discussion of the identifiability conditions of a system.

3.1 System Identification

3.1.1 Problem Formulation

In Chapter 2 the environment was parameterized by the vector ξ , and the data were measurements of the acoustic field at the array. These data could be the time domain signal, the cross-spectral density of the received signal, or any other suitable representation of the acoustic field. Let us confine the measurement to be in vector

form

$$\mathbf{d} = [d_1 \ d_2 \ \cdots \ d_N]^T \quad (3.1)$$

where N is the dimension of the measurement vector. In order to maintain generality let us assume that the data vector \mathbf{d} is related to the environment vector by some system function

$$\mathbf{d} = H(\boldsymbol{\xi}) \quad (3.2)$$

The forward model \tilde{H} is a numerical approximation to the actual system function H . In this light, the inverse system identification problem is: *given an observation \mathbf{d} , and the forward model \tilde{H} , find the model parameters $\boldsymbol{\xi}$.*

The solution to this problem can be viewed in two ways. The first approach involves determining the environmental parameters from the data using the forward model, as in matched-field inversion. One of the most robust methods is the exhaustive search. In this method the forward model is evaluated for all possible environment vectors to find the output of the model which is sufficiently close to the data. The ‘closeness’ of the model output to the data is usually quantified by an error function which measures the error between the two vectors. This error is sometimes called the mismatch (see Equation 2.50). Matched-field inversion uses a wide variety of different matching functions which can be tailored to each experiment [8].

A second viewpoint for this problem arises if an inverse model G is known or can be approximated. Here we are estimating the inverse model directly rather than just the environmental parameters. The inverse model is defined as

$$\boldsymbol{\xi} = G(\mathbf{d}) \quad (3.3)$$

If the inverse model G exists and is known then it is a simple matter of evaluating this function for the data to obtain the model parameters. Neural system identification approximates G with a neural network trained using the forward model. This is depicted in Figure 3.1. In this figure the vector \mathbf{d} is the output of the numerical model corresponding to the environment $\boldsymbol{\xi}$. The difference between the environment vector predicted by the inverse model G and the actual environment vector is used to update the weights in G . This process is iterative when multiple data/environment pairs need to be learned.

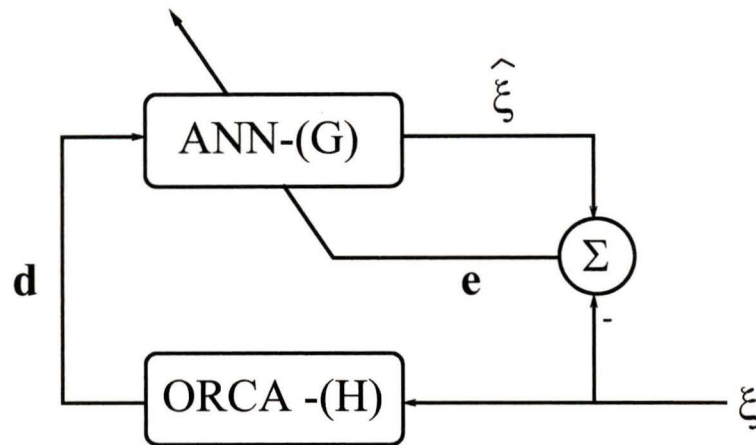


Figure 3.1. An artificial neural network (ANN) being trained with a forward model (ORCA). The data vector \mathbf{d} is the output of the model for the environment vector ξ . The difference between the environment predicted by the inverse model G and the actual environment is used to update the weights in G .

3.1.2 Neural System Identification

Figure 3.2 shows a neural identification system. Each block is important in achieving a successful inversion. The procedure begins with an experiment. The experiment must make measurements that are dependent on the parameters we wish to extract. After the data set is collected it should be analyzed and the portions of the data which can be modeled should be extracted. In the case of ORCA the spectral components are required; thus a spectral estimation stage is carried out. From this stage an estimate of the noise can be incorporated into the model. This can be done in a straightforward manner by adding the same amount of noise to the numerical model outputs as is present in the data. It is vital for the test data to be as close to the real data as possible so that the estimate of the error is accurate.

The next block is labeled pre-processing, which generally involves normalization of the data followed by feature extraction. For example, it may be important for each data vector to lie on the unit circle (or hypersphere), and normalization with respect to the L_2 -norm should be employed. Feature extraction attempts to tease information out of the data vector which is helpful in the identification of one or more environmental parameters. This is often done heuristically (as with empirical

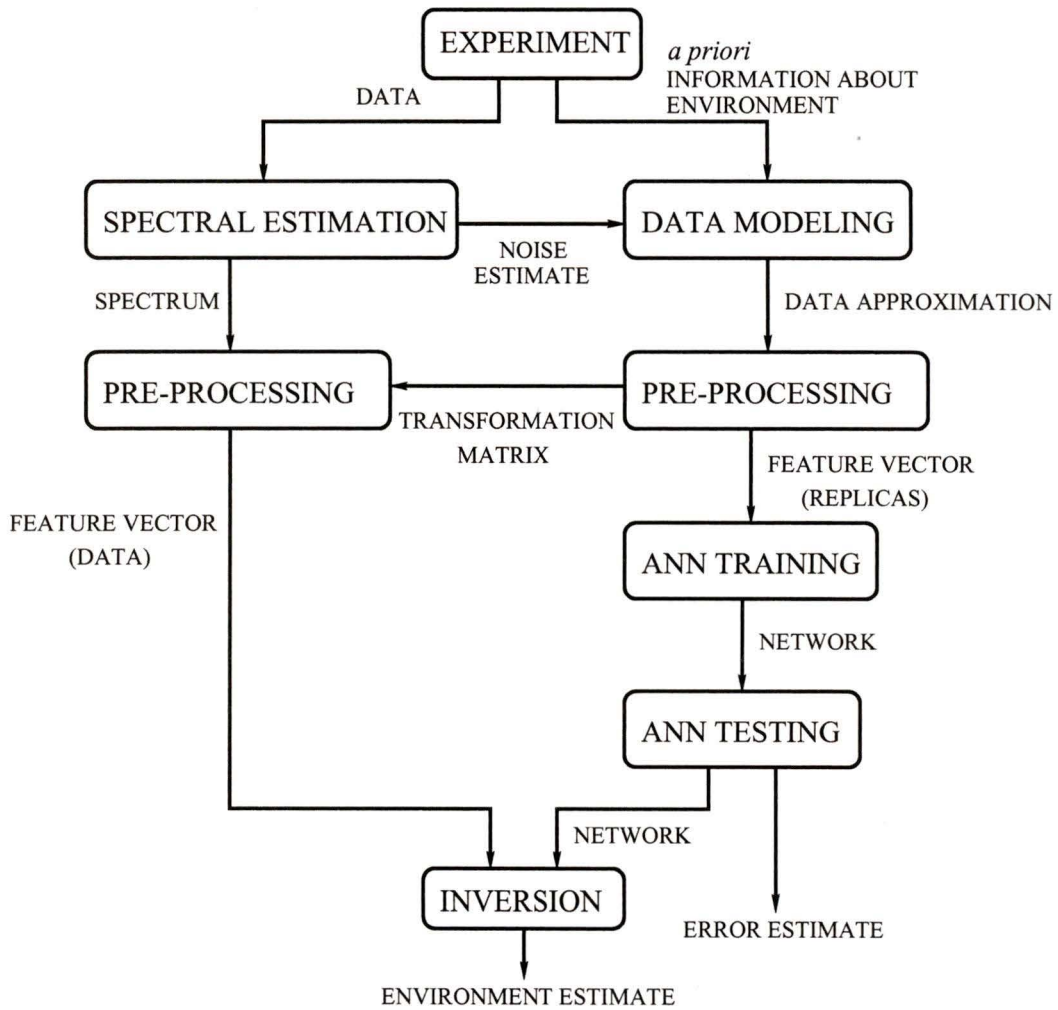


Figure 3.2. A neural identification system.

orthogonal functions) or using *a priori* knowledge of the system.

When nothing is known in advance about the relationship between the parameters and the environment, the forward model can be used to generate simulated data for training and testing. The forward model can generate the simulated data for a large number of different environments. The most important features of the data for the different environments can then be deduced empirically. Once a set of features has been found from the data, the algorithm to extract these features must also be applied to the experimental data. The feature extraction is often a linear transformation of the data. Feature extraction attempts to reduce the dimension of the data, decorrelate the inputs and remove noise. Methods of accomplishing this are described in Section 3.2.3.

Usually two simulated data sets are constructed using the forward model: the training set and the test set. Both sets comprise input/output pairs. In matrix notation, the input set is

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1M} \\ p_{21} & p_{22} & \cdots & p_{2M} \\ \vdots & \vdots & & \vdots \\ p_{Q1} & p_{Q2} & \cdots & p_{QM} \end{bmatrix} \quad (3.4)$$

where Q is the dimension of the input to the network, and M is the number of training examples. Each column of \mathbf{P} is a single input feature vector and is denoted by $\mathbf{p}(m)$ or \mathbf{p}_m . The corresponding desired output or target matrix is

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1M} \\ t_{21} & t_{22} & \cdots & t_{2M} \\ \vdots & \vdots & & \vdots \\ t_{R1} & t_{R2} & \cdots & t_{RM} \end{bmatrix} \quad (3.5)$$

where R is the dimension of the output vector. Each column of \mathbf{T} is a target output $\mathbf{t}(m)$ or \mathbf{t}_m , for the input in the same column of \mathbf{P} , thus $\mathbf{p}_m = H(\mathbf{t}_m)$.

The feature vectors are used to train an artificial neural network to perform the mapping from the data feature vector space to the environmental parameter vector space. Typically this training uses the same large data set which was used to determine the best features for extraction. Once the network weights are fixed after training, the network can be tested with another simulated data set. The training

data *must* have the same characteristics as the experimental data including the possibilities of mismatch and noise which may not have been included in the training data.

The expected value of the error in the test set is the estimated error for the inversion. The most common measurement of this error is the mean-squared error

$$mse = \frac{1}{MR} \sum_{m=1}^M \sum_{r=1}^R e_{rm}^2 \quad (3.6)$$

where $e_{rm} = t_{rm} - y_{rm}$ is the error between the target output and the network output. For the entire training set this is represented in matrix form as

$$\mathbf{E} = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1M} \\ e_{21} & e_{22} & \cdots & e_{2M} \\ \vdots & \vdots & & \vdots \\ e_{R1} & e_{R2} & \cdots & e_{RM} \end{bmatrix} \quad (3.7)$$

The mean-squared error for the r th parameter is given by

$$mse_r = \frac{1}{M} \sum_{m=1}^M e_{rm}^2 \quad (3.8)$$

Once the error in the test set has been established, the final stage is to invert the experimental data. This is done by presenting the experimental data feature vectors to the trained and tested network. The output of the network is the final estimate. The solution can be checked by modeling the data using the estimated environment. If the modeled data matches the actual data the inversion is a success. The success can be quantified by measuring the increase in the Bartlett power with the new environmental parameters.

3.1.3 Neural Classifiers

Neural networks can also be used to classify data as belonging to a certain group or cluster. This is often done when the parameters are correlated, and only a rough estimate of the ranges of the parameters is required. Rather than estimate each parameter of the environment in a continuous fashion, the possible environments are divided into classes. Each class has typical parameter values, and a range of

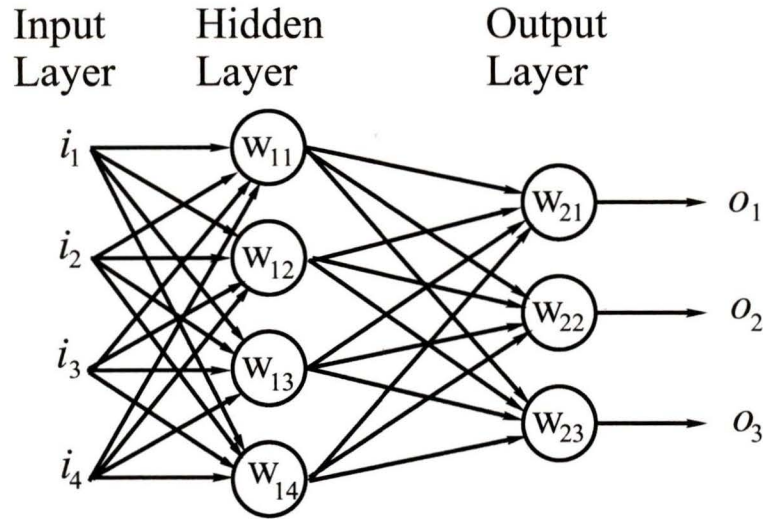


Figure 3.3. A general feedforward neural network.

uncertainty for each parameter. Good class selection takes advantage of the fact that some of the parameters may be correlated. For example high material densities are usually seen along with large sound speeds. The neural classifier examines the data and decides which class it belongs to. This type of network is trained in the same way as the continuous estimator. The advantage of the neural classifier is that it can be quite robust. Classifiers are generally much simpler and hence take less time to train than continuous estimators. The obvious disadvantage is that an explicit estimate of the individual ocean parameters is not obtained.

3.2 Artificial Neural Networks

In this section two types of artificial neural networks (ANNs) are described along with their training algorithms. A feedforward artificial neural network is shown in Figure 3.3. This network has one input layer, one hidden layer, and one output layer. Only the hidden layer and the output layer have neurons. The input layer serves only as a transfer point for the input pattern. Feedforward neural networks are sometimes called static neural networks in contrast with networks that have feedback. Networks with outputs that feed backward to previous layers are called recurrent, or dynamical networks. In this thesis only feedforward networks are considered because of their

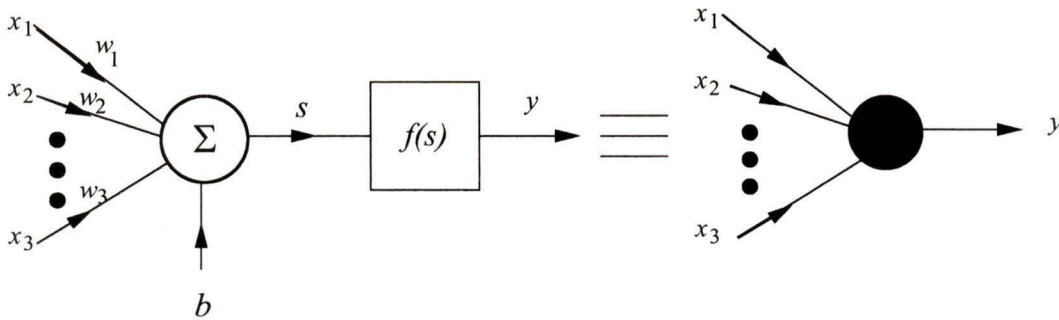


Figure 3.4. A summing neuron. The activation function $f(s)$ can have many forms, although nonlinear differentiable functions are preferred. The bias b is usually included as a variable in the training algorithm.

simplicity.

Each neuron in the network can be one of two types: a summing neuron or a radial basis function neuron. A summing neuron is depicted in Figure 3.4. The transfer function for this neuron is

$$y = f\left(\sum_{i=1}^n w_i x_i\right) \quad (3.9)$$

The *activation* function $f(s)$ can be one of several different kinds. Figure 3.5 shows five functions that are commonly used, where $s = \sum_{i=1}^n w_i x_i$. Feedforward networks which are used for function approximation, most often use a sigmoid function for the hidden layer neurons, and a linear function for the output layer. The sigmoid function is shown in Figures 3.5d and 3.5e. The network weights are usually trained using an algorithm called *backpropagation*. Networks of this type are called backpropagation neural networks (BPNN).

The other class of neurons that we will consider is the radial basis function. This neuron is depicted in Figure 3.6 This type of neuron has an activation function

$$f(\mathbf{x}) = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{c}\|_2^2\right) \quad (3.10)$$

where the vectors \mathbf{x} and \mathbf{c} are the input and weight vectors, respectively. The vector \mathbf{c} is commonly called a *centre*. The *spread* σ controls the width of each Gaussian. Typically radial basis function neurons are only used in the hidden layer. The most popular application for radial basis function neural networks (RBFNNs) is as function

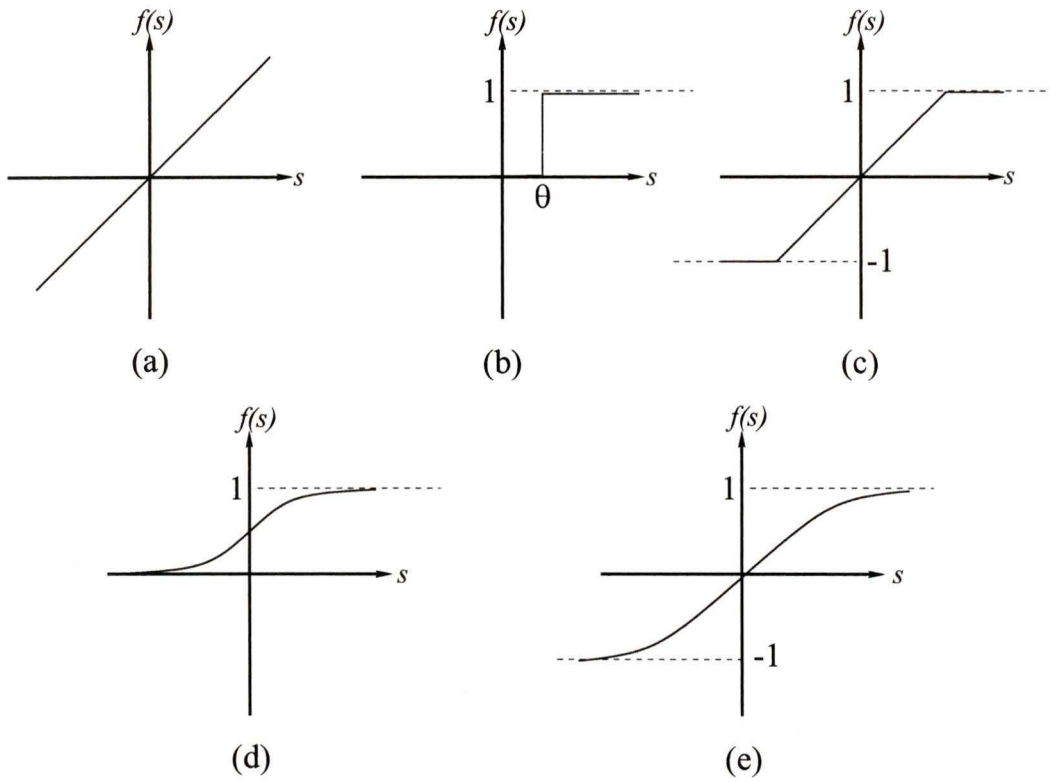


Figure 3.5. Several activation functions commonly used in neural networks. Note that $s = \sum_{i=1}^n w_i x_i$.

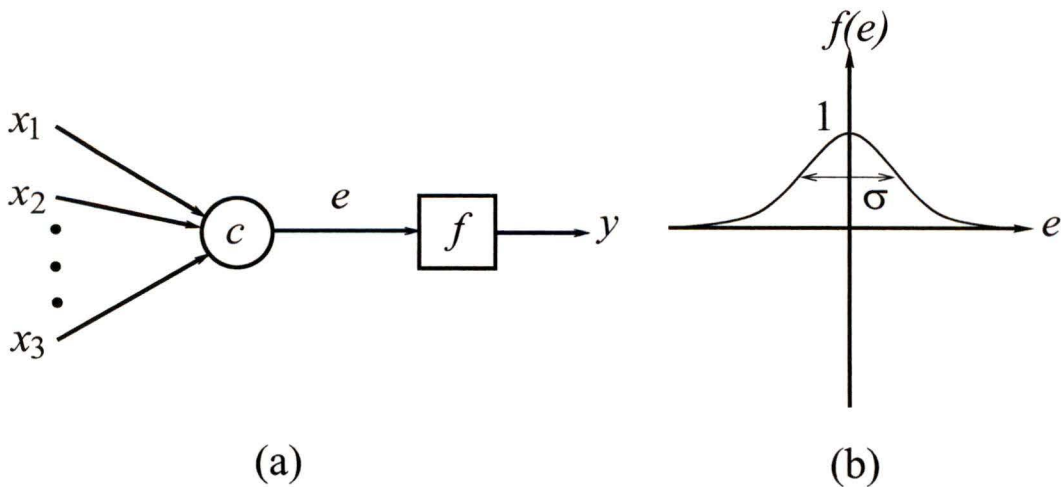


Figure 3.6. A radial basis function neuron. Note that $e = \|\mathbf{x} - \mathbf{c}\|_2^2$.

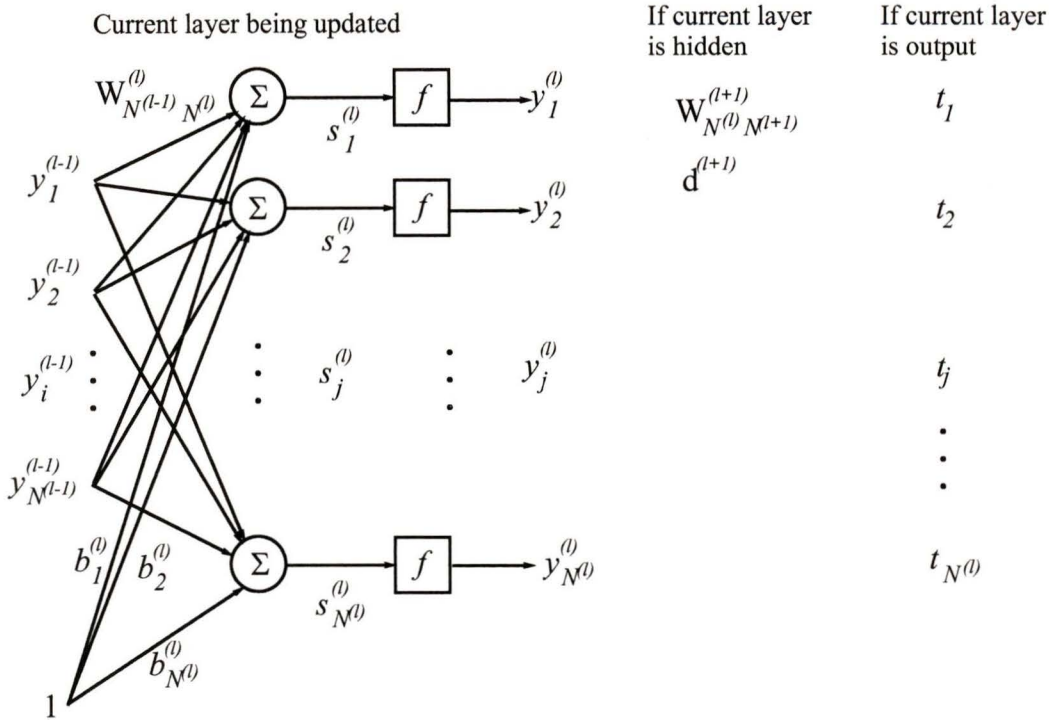


Figure 3.7. A generalized layer for the backpropagation algorithm.

approximators, thus the output layer is most often comprised of a layer of linear neurons. A special class of RBFNNs, called *exact* networks, require no iterative optimization for their design.

Both BPNNs and RBFNNs have been proven to be *universal function approximators* [25] [26] [27]. Any continuous mapping between one vector space and another can be approximated by these networks provided that enough neurons are used. In practice certain limitations are associated with these networks, which are detailed at the end of the chapter. The design of both networks is described in the following sections.

3.2.1 Backpropagation Neural Networks

The backpropagation algorithm (BP) is used for networks with more than one layer. A single layer of a multi-layered network is shown in Figure 3.7. This layer is general and applies to either a hidden or an output layer. The total number of layers is L . There are $L - 1$ weight matrices because the input layer outputs are simply the

inputs. The L th layer is the output layer. In element-wise notation the output of the l th layer is

$$y_j^{(l)} = f(s_j^{(l)}) \quad \text{where} \quad s_j^{(l)} = \sum_{i=1}^{N^{(l-1)}} w_{ij}^{(l)} y_i^{(l-1)} + b_j \quad (3.11)$$

If we let $y_{N^{(l-1)+1}}^{(l-1)} = 1$ and $w_{(N^{(l-1)+1})j}^{(l)} = b_j$, then the summation becomes

$$s_j^{(l)} = \sum_{i=1}^{N^{(l-1)+1}} w_{ij}^{(l)} y_i^{(l-1)} \quad (3.12)$$

In matrix notation we have

$$\mathbf{y}^{(l)} = f(\mathbf{s}^{(l)}) \quad \text{where} \quad \mathbf{s}^{(l)} = (\mathbf{W}^{(l)})^T \mathbf{y}^{(l-1)} \quad (3.13)$$

where

$$\mathbf{W}^{(l)} = \begin{bmatrix} w_{11}^{(l)} & w_{12}^{(l)} & \cdots & w_{1N^{(l)}}^{(l)} \\ w_{21}^{(l)} & w_{22}^{(l)} & \cdots & w_{2N^{(l)}}^{(l)} \\ \vdots & \vdots & & \vdots \\ w_{N^{(l-1)+1}1}^{(l)} & w_{N^{(l-1)+1}2}^{(l)} & \cdots & w_{N^{(l-1)+1}N^{(l)}}^{(l)} \\ b_1^{(l)} & b_2^{(l)} & \cdots & b_{N^{(l)}}^{(l)} \end{bmatrix} \quad (3.14)$$

The data input vector to the network is $\mathbf{x} = \mathbf{y}^{(1)}$. The output of the l th layer is $\mathbf{y}^{(l)}$. The structure of each input is $\mathbf{y} = [y_1 \ y_2 \ \cdots \ y_i \ \cdots \ y_{N^{l-1}} \ 1]^T$ where N^{l-1} is the size of the l th layer. The one appended to the input vector is to account for the bias term. The output $\mathbf{y}^{(L)}$ of a network with two hidden layers ($L = 3$) for an input $\mathbf{y}^{(1)}$ is

$$\mathbf{y}^{(3)} = f \left\{ (\mathbf{W}^{(3)})^T f \left[(\mathbf{W}^{(2)})^T f \left[(\mathbf{W}^{(1)})^T \mathbf{y}^{(1)} \right] \right] \right\} \quad (3.15)$$

The weights are trained using the input/output patterns \mathbf{P} and \mathbf{T} whose structure is described in Equations 3.4 and 3.5. Backpropagation involves propagating the error at the output for the m th pattern $\mathbf{t}_m - \mathbf{y}_m^{(L)}$ back through each layer of the network until each weight matrix has been updated. The weights are updated using the Delta Rule. For an output layer this weight change is calculated in three steps. First the error is propagated back through the non-linearity.

$$\mathbf{d}_m^{(L)} = (\mathbf{t}_m - \mathbf{y}_m^{(L)}) f'(\mathbf{s}_m^{(L)}) \quad (3.16)$$

where the subscript m indicates that this is for the m th pattern of the training set and $f'(\mathbf{s})$ is the derivative of the output layer's activation function with respect to \mathbf{s} . The output vector $\mathbf{y}_m^{(L)}$ does not contain the appended one that is required for the hidden layers. Next the change in the weight matrix is calculated

$$\mathbf{D}_m = \eta \mathbf{d}_m^{(L)} (\mathbf{y}_m^{(L-1)})^T \quad (3.17)$$

where η is the learning rate which controls the size of the weight change and $\mathbf{y}_m^{(L-1)}$ is the output of the penultimate layer; this vector does have the appended one, thus the biases are updated along with the rest of the weights. The learning rate η is typically around 0.9. The weight change

$$\mathbf{W}_{m+1}^{(L)} = \mathbf{W}_m^{(L)} + \mathbf{D}_m \quad (3.18)$$

is then applied.

Hidden layers do not have target outputs, thus the error must be backpropagated from the output layer. This is done using the following relation for the l th layer

$$\mathbf{d}_m^{(l)} = f'(\mathbf{s}_m^{(l)}) \circ (\mathbf{W}_m^{(l+1)} \mathbf{d}_m^{(l+1)}) \quad (3.19)$$

where $\mathbf{a} \circ \mathbf{b}$ denotes the Hadamard product (element-wise product) of vectors \mathbf{a} and \mathbf{b} . In this equation the backpropagated error $\mathbf{d}_m^{(l+1)}$ from the previous iteration (layer) is multiplied by the old weights of that layer $\mathbf{W}_m^{(l+1)}$ to obtain the backpropagated error for the current iteration $\mathbf{d}_m^{(l)}$. Equations 3.17 and 3.18 are then used to update the weights as with the output layer. The weights of each layer are updated for each pattern in the training set and the process is repeated until the error is acceptable.

An *epoch* is defined as a single iteration of weight updates for all of the M training patterns. Backpropagation, like the steepest-descent algorithm (SD), is notoriously slow and should include the maximum number of epochs as an alternative stopping criterion to meeting the error goal.

The learning rate η is the same as the step size for the steepest-descent algorithm. The convergence of SD can be improved significantly if the step size is varied, such as with a line search [28]. The same is true for BP. An adaptive learning rate speeds convergence by using a large step size when the weights are far from their optimal values, and using a smaller step size as the solution is approached.

One adaptation technique used with backpropagation is to vary the learning rate with the ratio of the error between epochs. The learning rate for the $(k + 1)$ th epoch is

$$\eta(k + 1) = \begin{cases} \eta(k) + \Delta_\eta & \text{if } a < \theta \\ \eta(k) - \phi_\eta & \text{if } a > \theta \end{cases} \quad (3.20)$$

where θ is a threshold on the interval $(0,1)$, Δ_η is a small increase in η , and ϕ_η is a small decrease in η . This method increases the learning rate when there are large decreases in the error, and decreases the learning rate when the error is only decreasing slowly, or is increasing.

Large problems and noisy data sets tend to generate error surfaces which are not unimodal. The presence of many local minima is mainly due to the nonlinearity of the neurons. In large problems the high dimensionality causes slow convergence and the increasing likelihood of becoming trapped in undesirable local minima. One method of avoiding and escaping local minima is by adding a momentum term to the delta rule. The new delta rule can be written as

$$\hat{\mathbf{d}}_{mk}^{(l)} = \mathbf{d}_{mk}^{(l)}(1 - m_c) + m_c \mathbf{d}_{m(k-1)}^{(l)} \quad (3.21)$$

where k is the epoch, m_c is the momentum constant and $\hat{\mathbf{d}}_{mk}^{(l)}$ is the error with momentum. A typical value for m_c is 0.1. The addition of the error from the previous epoch to the current epoch can push the algorithm out of local minima in which it might otherwise have become stuck. The backpropagation algorithm used to obtain the results of Chapter 5 is listed below.

Backpropagation Algorithm

1. Input L , $N(L)$, \mathbf{P} , \mathbf{T} , m_c , η , Δ_η , ϕ_η , θ , k_{\max} , M , and e_{goal} .
2. Set $k = 0$, $e = 10^{16}$, $\mathbf{d}_{last} = 0$. Initialize weights.
3. Set $m = 0$, $k = k + 1$, $e = 0$.
4. Set $m = m + 1$, $l = 1$, $\mathbf{y}(l) = \mathbf{P}(m)$.
5. Calculate $l = l + 1$, $\mathbf{s}(l) = \mathbf{W}^T(l)\mathbf{y}(l - 1)$, $\mathbf{y}(l) = f(\mathbf{s}(l))$.
6. If $l \leq L$ then go to 5.
7. Calculate $\mathbf{d}(L) = (\mathbf{T}(m) - \mathbf{y}(L))f'(\mathbf{s}(L))(1 - m_c) + m_c \mathbf{d}_{last}(L)$
8. Calculate $\mathbf{d}_{last}(L) = \mathbf{d}(L)$, and $\mathbf{D} = \eta \mathbf{d}(L) \mathbf{y}(L - 1)^T$.

9. Calculate $\mathbf{W}(L) = \mathbf{W}(L) + \mathbf{D}$, and $e = e + \sum(\mathbf{T}(m) - \mathbf{y}(L))^2$.
10. Calculate $l = l + 1$, $\mathbf{d}(l) = f'(\mathbf{s}(l)) \circ (\mathbf{W}(l+1)\mathbf{d}(l+1))(1 - m_c) + m_c\mathbf{d}_{last}(l)$.
11. Calculate $\mathbf{D} = \eta\mathbf{d}(l)\mathbf{y}(l-1)^T$, $\mathbf{W}(l) + \mathbf{D}$, and $\mathbf{d}_{last}(l) = \mathbf{d}(l)$.
12. If $l \leq 2$ then go to 10.
13. If $m \leq M$ then go to 4.
14. Calculate $a = e/e_{last}$, $e_{last} = e$.
15. if $a < \theta$ then set $\eta = \eta + \Delta_\eta$.
16. if $a > \theta$ then set $\eta = \eta - \phi_\eta$.
17. If $k < k_{max}$ and $e > e_{goal}$ then go to 3, else stop.

The main drawback of BPNNs is that the design is iterative and convergence to the global minimum is not guaranteed. Even with the modifications suggested here, the backpropagation algorithm is slow. There are algorithms which form the problem as an objective function of the weights directly. This objective function is then optimized using one of the well-known quasi-Newton methods. These methods are much faster than BP, except when a large training set or network is used. This is because the dimension of the problem is such that the algorithm is memory intensive and must use the virtual memory of the computer. When this occurs, it is advantageous to use the backpropagation algorithm because its slow convergence is outweighed by its low memory requirement. Three techniques can be used to speed the convergence of backpropagation even more: multiple starting points, cross-validation, and prototype training.

The multiple starting points method uses different sets of initial weights and trains each set until convergence, or for a fixed number of epochs. The set of weights which has the lowest error after this training is chosen to continue the training until the error goal has been reached. In this way many local minima which are undesirable are rejected. The number of starting points needed is problem dependent.

Training the weights for too long can result in what is called *memorization*. Memorization is seen when the network is tested on inputs which were not in the training set. If the error of the test set is much higher than that of the training set, then the network is not *generalizing* well for data which it has not previously seen. This is illustrated in Figure 3.8. Memorization can be avoided by using a smaller network

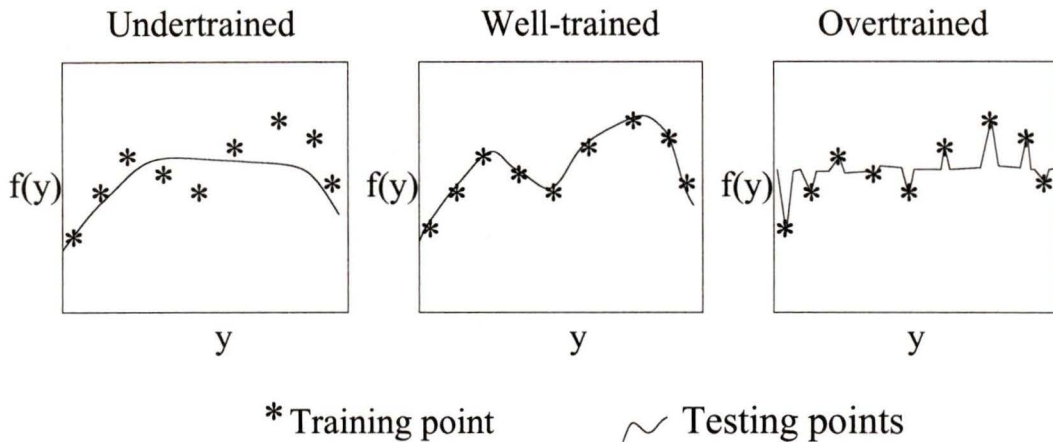


Figure 3.8. An undertrained, well-trained, and overtrained network. The overtrained network is said to have memorized the training data rather than the functional mapping. The middle network generalizes well for data in the test set.

or through the use of *cross-validation*, which is also known as early stopping. Cross-validation checks the error of the test set after each epoch of training. The error in the test set usually follows that of the training set up until the point when memorization begins. At this point the error in the test set flattens out, and then begins to increase. With cross-validation, training is stopped as soon as the error begins to increase. Figure 3.9 shows a typical training record. In this instance the network is trained for a fixed number of epochs and both the training set and test set error are calculated after each epoch. The optimal stopping point is the point at which the test set error begins to increase. If the error in the test set is not low enough at this minimum point, then a different starting point or more neurons may be required.

Prototype training is used to find a good set of initial weights before starting the backpropagation algorithm. A modified training set is used to train the weights initially. This training set can be a subset of the full training set, or, in the case of classification, it can be the mean feature vector of each class. The network is trained for a short time until the error is quite low for the small training set. After this the full training set is used, usually with the result that local minima are avoided and convergence is faster. An extension of this technique can be used when the data to be inverted is noisy. The network is trained on noise-free data initially. Noise is then added to the training set and training is continued. The local minima of the error

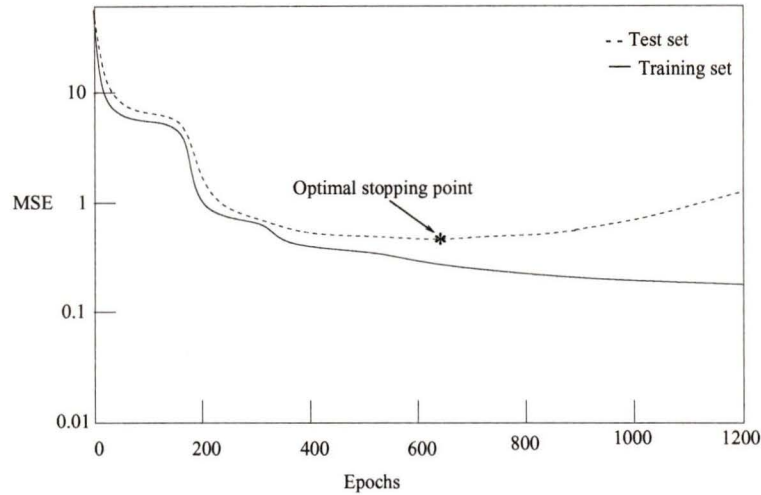


Figure 3.9. A training record for cross-validation noting the optimal stopping point.

function which are created by the noise are effectively avoided by starting with the already trained weights.

A combination of all three methods was used to obtain the results of Chapter 5. Many starting points were used and cross-validation was used as the stopping criterion. A test set which was different from both the training set and the cross-validation set was used to evaluate the network performance using the mean-squared error criterion.

3.2.2 Radial Basis Function

The design of exact radial basis function neural networks (RBFNNs) is much simpler than backpropagation networks since no iterative optimization is required. The structure of a RBFNN is illustrated in Figure 3.10. The neurons of the hidden layer are radial basis function neurons with centres \mathbf{c}_i and outputs ϕ_i . The output of the hidden layer is linearly mapped to the output of the network with the weight matrix \mathbf{W} . This is simply a layer of neurons with linear activation functions $f(s) = s$. If the centres are chosen so that each is a training input, then there will be M neurons. The weight matrix is trained using linear regression with least-squares.

The output of each neuron in the hidden layer for an input vector \mathbf{x} is

$$\phi_i = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{c}_i - \mathbf{x}\|_2^2\right) \quad (3.22)$$

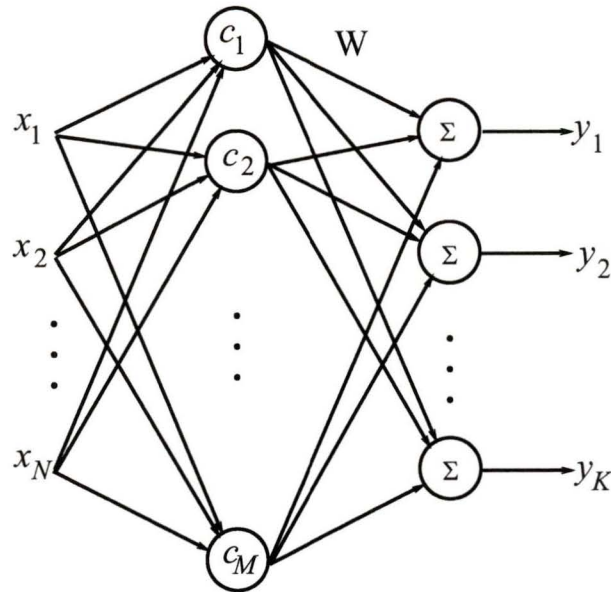


Figure 3.10. A radial basis function neural network. The hidden layer is composed of radial basis function neurons, and the output layer is linear.

The output of the entire layer in vector form is

$$\boldsymbol{\phi} = [\phi_1 \ \phi_2 \ \dots \ \phi_M \ 1] \tag{3.23}$$

where the one is included to account for the bias in the output layer. The output of the network is written as

$$\mathbf{y} = \mathbf{W}^T \boldsymbol{\phi} \tag{3.24}$$

where the weight matrix \mathbf{W} is of the form

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1K} \\ w_{21} & w_{22} & \dots & w_{2K} \\ \vdots & \vdots & & \vdots \\ w_{M1} & w_{M2} & \dots & w_{MK} \\ b_1 & b_2 & \dots & b_K \end{bmatrix} \tag{3.25}$$

and the output vector of dimension K is defined as

$$\mathbf{y} = [y_1 \ y_2 \ \dots \ y_K]^T \tag{3.26}$$

The RBFNN is most often used as a function approximator. A single output is a weighted sum of Gaussian functions. The Gaussians are used as the basis for the

function reconstruction. The centres of each radial basis function must be chosen such that the entire input and output space is sampled sufficiently.

If we choose the centres to be each of the training input examples $\mathbf{c}_i = \mathbf{p}_i$ then the output of neuron i for the j th example in the training set is

$$\phi_{ij} = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{c}_i - \mathbf{p}_j\|_2^2\right) \quad \text{for } i, j = 1, 2, \dots, M \quad (3.27)$$

The output ϕ_{ii} is equal to unity since the L_2 -norm will evaluate to exactly zero. The output matrix for the hidden layer is defined as

$$\mathbf{\Phi} = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1M} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2M} \\ \vdots & \vdots & & \vdots \\ \phi_{M1} & \phi_{M2} & \cdots & \phi_{MM} \\ 1 & 1 & \cdots & 1 \end{bmatrix} \quad (3.28)$$

where the j th column is the output of the hidden layer for the j th input. Again the row of ones is included to account for the bias terms in the output layer.

The weight matrix \mathbf{W} is found by computing the pseudoinverse of $\mathbf{\Phi}$. The output of the network is calculated using

$$\mathbf{Y} = \mathbf{W}^T \mathbf{\Phi} \quad (3.29)$$

where \mathbf{Y} is the matrix of output vectors defined similarly to the target matrix in Equation 3.5. The weight matrix is found using

$$\mathbf{W}^T = \mathbf{T} \mathbf{\Phi}^+ \quad (3.30)$$

where \mathbf{T} is the target matrix of Equation 3.5, and $(\cdot)^+$ denotes the Moore-Penrose pseudoinverse. The pseudoinverse is calculated using the singular value decomposition (SVD). The SVD of matrix \mathbf{A} is [29]

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \quad (3.31)$$

The pseudoinverse is

$$\mathbf{A}^+ = \mathbf{V} \mathbf{\Sigma}^{-1} \mathbf{U}^T \quad (3.32)$$

Since the matrix of singular values Σ is diagonal its inverse is calculated by taking the reciprocal of each diagonal element. Since each training example is included as a centre, the solution for the weights will be such that each target is met exactly. The error for the training set is zero, hence the term *exact* RBFNN.

Designing RBFs in this manner is fast compared to the iterative techniques for BPNNs but has a number of drawbacks. The first concern is the choice of the centres. Choosing centres from a randomly generated training set may result in centres which are quite close to each other with respect to the L_2 -norm. If this is the case then two of the columns and two of the rows of Φ will be nearly linearly dependent and thus Φ will be ill-conditioned. Calculating the pseudoinverse causes a problem in this case as there will be very small singular values which, when inverted, will be very large. This will cause a large dynamic range in the weight values. The solution to this problem is to choose centres which are as far apart as possible. This is sometimes done by clustering a large data set using an algorithm such as K-means [30]. The centre of each cluster is then used as a centre for the network. Of course K-means is an iterative algorithm which can be as time-consuming as backpropagation.

The choice of the spread value σ is usually unity for data normalized with respect to the L_2 -norm. A spread that is too small will still allow the RBF to estimate the training data perfectly. This is because matrix Φ will be close to the identity matrix. For inputs that are not part of the training set the output of the hidden layer will be approximately zero. Thus the network will output the bias values only. When looking at the test outputs it will appear as though the network has overtrained as seen in Figure 3.8. If, on the other hand, the spread is too large then the output of the hidden layer will be unity for any input. Matrix Φ will be very ill-conditioned since its elements are all ones. The weight matrix will also be ill-conditioned because it is dependent on Φ . The output of the network for most inputs (even the training set) will simply be the column sum of the weight matrix. This situation can be paralleled with that of an undertrained BPNN. The most suitable method for calculating the spread is to design several networks with a range of spreads and choose the network which has the lowest mse for the test set.

3.2.3 Pre-processing

In general the inputs to a neural network are heavily pre-processed in order to reduce the training time and network size. A feedforward neural network with at least one hidden layer and a nonlinear activation function is a universal function approximator [25] [26] [27]. If the inputs are noisy or correlated the network must remove the noise and extract the independent features of the data before it can estimate the function. The most common pre-processing techniques are: spectral estimation, noise reduction, normalization, and principal components analysis.

The goal of spectral estimation is to represent the data in the frequency domain. This is an effective way to reduce the dimension of time-domain data if it is known that the input signal contains only a few frequency components. The noise in the signal is also reduced because the frequency bands which are not part of the signal contain only noise. Spectral estimation is discussed more thoroughly in Chapter 4.

Noise reduction techniques are closely coupled with all stages of the pre-processing. The goal is to ensure that the data presented to the network comprise only components that are significant to the output of the network.

Normalization is used to put the experimental data on the same footing as the modeled data. The amplitude of the input signal may not be known and thus the amplitude cannot be modeled properly. Virtually any vector norm can be used to remove the input amplitude, assuming that the system is linear in the signal amplitude, or $H(A\boldsymbol{\xi}_s) = AH(\boldsymbol{\xi}_s)$, where A is the input transmitted signal amplitude, and $\boldsymbol{\xi}_s$ is the ORCA parameter controlling the signal amplitude. Another type of normalization requires that each element of the input vector have zero mean and a variance of one. This type of normalization is applied first to the training set. The mean input vector for the training set and the variance is then included in the transformation of the experimental data. The modeled mean is subtracted, and then the data is divided by the standard deviation of the training set. This type of normalization is sometimes called *standardization* and is required for principal components analysis.

Principal components analysis (PCA) [31] is used to decorrelate the input data and remove components of the data that do not contribute significantly to the total variance. The training data generated for the supervised training of a neural network are a set of measurements that correspond to a range of different environments. These

measurements also contain both correlated noise and uncorrelated noise. The uncorrelated noise is simply measurement noise. Correlated noise is problem dependent. This type of noise may arise in the ocean environment from distant noise sources whose waves propagate to the sensors through the same environment as the signal, and thus become correlated. PCA may remove some uncorrelated noise but the correlated noise would be unaffected. Principal components analysis is an active research topic in the area of statistics; what is presented here is generally termed *classical* PCA. Classical PCA is simply dimension reduction through the thresholding of a singular value decomposition of the data.

The training set of Equation 3.4 can serve as the set of measurements which we want to analyze. Each column is an observation and each row is a separate type of measurement for the same environment. Usually the measurement noise of row i is assumed to be Gaussian and the data should be normalized such that the mean of each row is zero and the standard deviation is one. Each column of the training set is \mathbf{p}_i , of dimension Q ; there are M observations.

The singular value decomposition is then applied to \mathbf{P} as

$$\mathbf{P} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (3.33)$$

where \mathbf{U} is $Q \times Q$, $\mathbf{\Sigma}$ is $Q \times M$, and \mathbf{V} is $M \times M$. Matrix $\mathbf{\Sigma}$ is diagonal with ordered diagonal elements $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ where $r \leq Q$. If $r < Q$ then matrix \mathbf{P} does not have full row rank, in other words two or more of the rows (measurements) are linearly dependent. Matrices \mathbf{U} and \mathbf{V} are the left and right singular vectors, respectively. These matrices are orthonormal, meaning that each column is orthogonal and the columns form a complete set; thus any vector in \mathfrak{R}^Q or \mathfrak{R}^M may be represented as a weighted sum of these vectors. Orthonormal matrices also have the property

$$\mathbf{A}^T = \mathbf{A}^{-1} \quad (3.34)$$

The singular values represent the standard deviation of the corresponding singular vectors. If we assume that the changes in the parameters cause most of the variance in the data, then the large singular values and their corresponding vectors are an efficient way of representing the change in the data due to the changes in the parameters. The dimension reduction comes about because we set to zero all those singular values which are below a preset threshold leaving only N non-zero values. The new reduced

singular values are $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_N \geq 0$. Rather than reconstruct the original data vector without noise, we can simply transform the data using

$$\tilde{\mathbf{P}} = \tilde{\Sigma} \mathbf{V}^T = \tilde{\mathbf{U}}^T \mathbf{P} \quad (3.35)$$

This makes use of Equation 3.34. Matrices $\tilde{\mathbf{P}}$, $\tilde{\Sigma}$ and $\tilde{\mathbf{U}}$ are the reduced matrices. This transformation reduces the dimension of the input vectors to N . An added benefit is that the rows of $\tilde{\mathbf{P}}$ are orthogonal and thus not correlated.

Pre-processing for neural networks is mostly for the input data; however, the output parameters should also be normalized. Often the parameters which are to be estimated are not expressed in the same units and there can be a large dynamic range. For example we could have t_1 in the range (1500, 2800) for sound speed in m/s and t_2 in the range (0.001, 0.003) for attenuation in dB/ λ . These two parameters make up the output vector $\mathbf{p} = [t_1 \ t_2]^T$. The large dynamic range in the output (six orders of magnitude) must also be reflected in the weight matrix for the output layer. This can predispose the weight matrix to being ill-conditioned. It is a simple matter to scale all of the output parameters to lie in the range [-1, 1]. This is done as follows

$$\tilde{t}_i = 2 \left(\frac{t_i - \min \mathbf{t}_i}{\max \mathbf{t}_i - \min \mathbf{t}_i} \right) - 1 \quad (3.36)$$

where \mathbf{t}_i is the vector containing all of the possible values of parameter t_i .

3.2.4 Identifiability Conditions

A problem common to both neural and matched-field inversion is that inverse problems are often ill-posed. Ill-posedness can be explained in terms of the identifiability conditions of a system which are:

- Uniqueness
- Continuity
- Existence

If there exist any two distinct $\boldsymbol{\xi}$ which produce an identical \mathbf{d} then G is said to be non-unique. Non-unique fields can arise because of correlated environmental parameters, an undersampled set of measurements, or noise; however, the physics of practical problems can be such that no experiment is capable of avoiding the non-uniqueness

problem. If incremental differences in the data correspond to large changes in the environment then the problem may be discontinuous. Continuity is related to the smoothness of the mapping between the input/output vectors. The smoother the mapping function, the easier it is to approximate. If there exists any \mathbf{d} for which a corresponding ξ does not exist, then the inverse function does not meet the existence criterion. In practice, existence is only verified when the solution is found.

In matched-field inversion the ambiguity surface is often replete with local optima that correspond to areas where the field is similar to the field for the true location. These local optima would not be present if the field changed slowly enough that the change in correlation was monotonic over the range of uncertainty. Occasionally there are situations where a sidelobe level equals that of the main peak. On these occasions the condition of uniqueness may be violated.

In the neural network scheme the identifiability condition violations appear in a different manner. If the problem is non-unique there will be more than one environment corresponding to the same data. This may cause the training algorithm to oscillate between different sets of weights and not converge, or the network weights will converge for the most important environmental parameter and the less important parameters will not be estimated properly. The network may also converge to the mean of the possible environments. If the function is not sufficiently smooth, the neural network will have to be prohibitively large to accurately model the inverse mapping function. If a smaller network is used it will fail to generalize for inputs it has not seen.

One solution to the uniqueness problem is to design the experiment so that there is spatial, spectral, and temporal diversity. Measuring at a large number of points will decrease the number of environments which are feasible. This introduces two problems: smoothness and dimensionality. If the dimension of the data vector increases then the number of forward model calculations must also increase. The number of inputs to the neural network increases the number of weights that must be optimized. Thus the solution time increases. The added diversity in the field also causes a problem in that the hypersurface which the neural net is attempting to reconstruct is so intricate that only a very high order network is capable of the task. This problem leads to poor generalization; thus there is a compromise that must be reached be-

tween diversity and smoothness in order to obtain the best possible estimates from the data in a reasonable time. No specific design criteria exist which fix the training set size and the network size. The most commonly used method is trial and error.

3.3 Conclusions

In this chapter we have developed a framework for geoacoustic inversion using neural networks. The problem was formulated as one of inverse system identification, and neural networks were suggested as a general structure that can be used to approximate the inverse function. Two types of neural networks were found to be suitable for this task, the backpropagation neural network, and the exact radial basis function neural network. Both networks use supervised training to determine their weights. The input/output pairs for the networks can be provided by the ORCA model. Neural classifiers were also introduced as an alternative to discrete parameter estimation. Classifiers have an advantage in that they are immune to problems with correlated parameters.

Several pre-processing steps which can be used to decrease the training time of the networks were developed. These steps include spectral estimation, normalization, standardization, and principal components analysis. Principal components analysis reduces the dimension of the input data while reducing the noise.

Finally the identifiability conditions were discussed. The manner in which violations of these conditions are manifested were described. The compromise between uniqueness and smoothness of the training data was explained.

Chapter 4

TRIAL SABLE Inversion

4.1 Introduction

The TRIAL SABLE acoustic experiment was performed off Canada's east coast in May and June of 1996. It was undertaken by the Department of National Defence, Defence Research Establishment Atlantic, along with the help of several contractors [32] [33] [14]. The ship used was the CFAV QUEST. The cruise included many different acoustic experiments. The experiment discussed here was done with the OMEGA vertical line array (VLA) [34] and a continuous-wave multi-tone source. The data from this experiment were gathered on May 27, 1996. The data consist of the recordings of eleven hydrophones in a moored vertical array obtained while the ship and source moved away on a radial track from a range of 0.4 km to 10 km.

The bathymetry of the track is thought to be rather flat. The water depth is approximately 78 m at the array site. The sea bottom is believed to consist of a highly reflecting sediment layer known as Sable Island sand and gravel, over the top of a tertiary bedrock basement.

During the experiment the sound speed profile in the water column was measured using expendable bathy-thermographs (XBTs). The source depth and array depth were monitored with pressure sensors. The array tilt was recorded. The position of the ship and the array were obtained using the Global Positioning System (GPS). Previous studies of the experimental location determined estimates of the bottom parameters such as the compressional speed, shear speed, density, and attenuation for each of the layers using various techniques. The data collected at the site and the information available about the ocean bottom are not precise and are generally presented as a range of possible values.

The purpose of this chapter is to summarize the data measured during the experiment and the information obtained from other studies. A discretized environment that is compatible with the ORCA normal mode model will also be derived. The validity of any assumptions will be examined and a test which uses the data for source localization will be performed as a justification. A simulation study will be performed to determine which of the poorly known parameters are important in the calculation of the field.

4.2 Environment Parameterization

4.2.1 Source locations

The ship track for the experiment is shown in Figure 4.1.¹ This track was recorded by the on-board GPS. The GPS is accurate (in a relative sense) to 2 m. The source itself was towed some distance behind the ship. This distance was not recorded but is believed to be approximately 100 m with an estimated error of ± 50 m, thus the range of uncertainty (ROU) in the source range is ± 52 m not counting the error in the receiver localization due to array tilt.

The source was localized in depth during the experiment using a pressure sensor. The sensor readings vary between 96 and 106 ± 3 ft for the entire track. The mean depth is 103 ft. Thus the nominal depth is set at 31.4 ± 2.5 m.

4.2.2 Receiver Locations

The location of each element in the array is dependent on a number of factors. The actual position of the bottom of the array is fixed with some accuracy using the GPS. Thus the distance from the array to the ship can be found. The array depth, however, is not known precisely. The array depth is recorded using a pressure sensor in the lower electronics unit (LEU) of the OMEGA VLA [34]. Unfortunately the exact position of the depth sensor within the LEU is not known. This introduces an uncertainty of about 3 m in the array depth. However, the relative positions of the hydrophones are assumed fixed and are listed in Table 4.1. The topmost hydrophone

¹This figure was generated by Mark Fallat of SEOS using IDL.

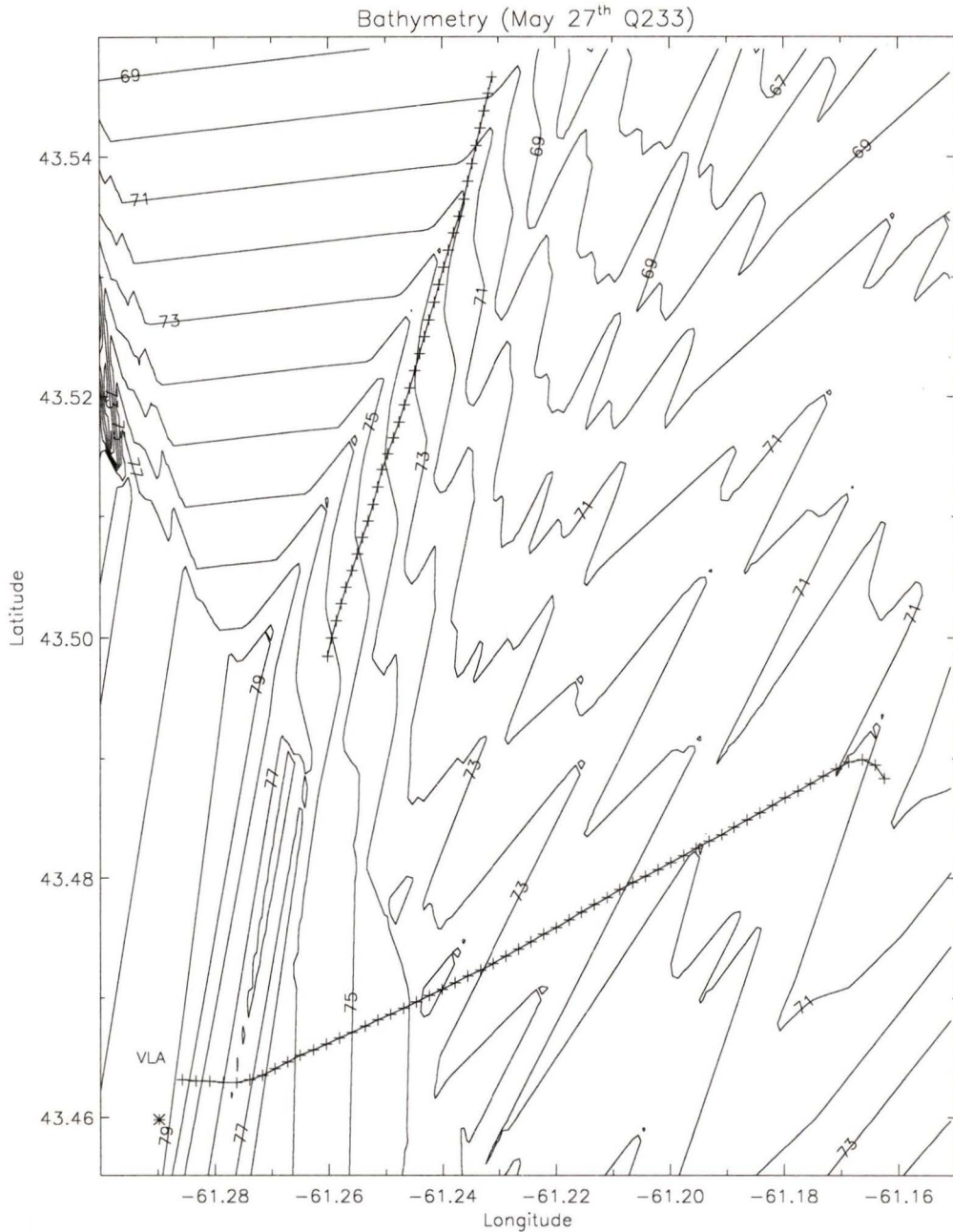


Figure 4.1. *The ship track overlaying the bathymetry of the experimental location shows a range-dependent environment. The ship track is represented by '+'s. The lower track is the outgoing leg and the upper track is the incoming leg. The first portion of the outgoing leg is used in this thesis. The array location is noted by an asterisk in the lower left-hand corner.*

Table 4.1. *Relative Hydrophone Locations*

Hydrophone #	Distance from top of LEU (m)
1	2.90
2	3.81
3	10.52
4	17.22
5	23.93
6	33.36
7	34.28
8	37.63
9	40.98
10	44.34
11	51.04

(12) was not operational. Hydrophone 1 is the bottom-most sensor, and hydrophone 11 is the topmost sensor. Other factors that may influence the exact array position are the tidal currents and, consequently, the time of day. The effect of the currents on the array is that it becomes curved, rather than remaining straight. Despite the existence of an array tilt sensor it is not known in which direction the array is tilted. The tilt readings may not be related to the effective tilt of the array because of the array curvature. From preliminary analysis at sea, the effective array tilt is thought to be less than 6 degrees at all times. Using this value each sensor can be assigned an additional range uncertainty of 8 m and an additional depth uncertainty of 0.4 m. This corresponds to the worst case for the topmost hydrophone. Another way of parameterizing the array tilt is to assume that the distance between each hydrophone is constant (this is a good assumption because the array itself does not stretch or contract), and that the array is not bent in any way. If the bottom hydrophone is assumed to be fixed in place with the previously described uncertainties in range and depth, then a single tilt angle can be assigned. A conservative maximum for the tilt angle would be ten degrees. This does not take into account the curving of the array, but it is an efficient and effective parameterization. The position of the array depth sensor is known to be 74.1 ± 3.0 m. Thus the total uncertainty in the array depth is

Table 4.2. *Array Element Parameterization*

Parameter	Nominal value	Lower bound	Upper bound	Units
array tilt	0	-10	10	degrees
array height	5	1	9	m

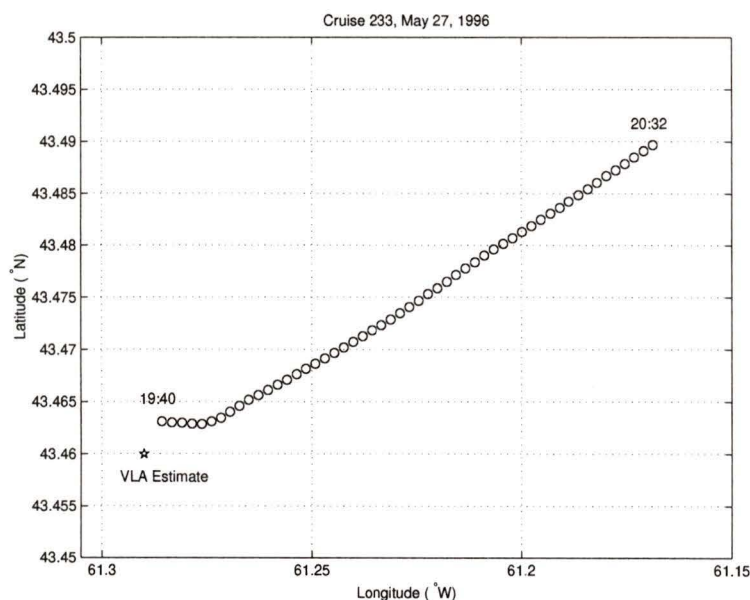


Figure 4.2. *The outgoing ship track and the location of the array are shown. Each successive circle is one minute further up the track.*

± 3.4 m. The composite uncertainty in range for each hydrophone is ± 60 m. The nominal tilt angle is set at zero with an uncertainty of 10 degrees. A positive tilt indicates that the array is tilted away from the source. Table 4.2 summarizes the hydrophone position parameters. The array height is above a fixed ocean depth of 77 m.

4.2.3 Bathymetry

The ship track and the depth below the ship track are illustrated in Figures 4.1 - 4.3. If we assume range-independence, the mean water depth can be determined using the bathymetry data of Figure 4.3. Range independence is a valid assumption because the

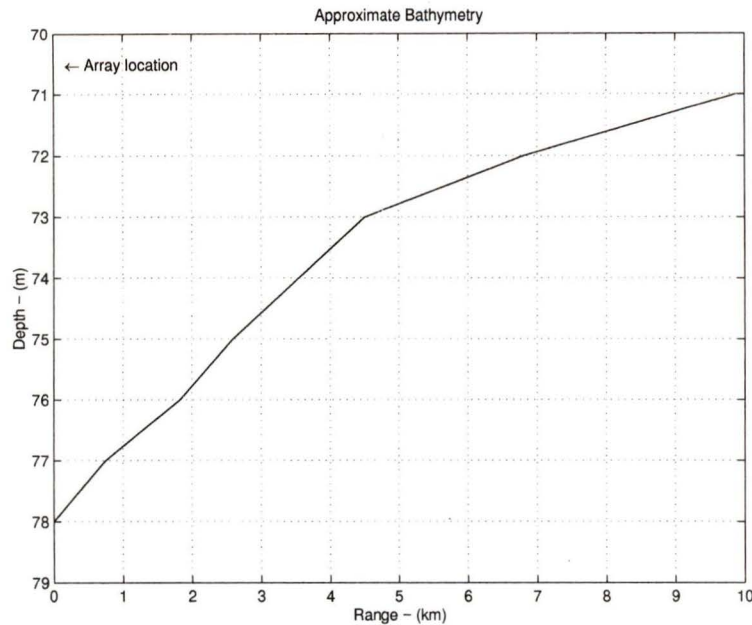


Figure 4.3. *The depth below the ship track shows that for the first kilometer the dip is less than 1.25 m, thus the slope is less than 0.125 %.*

slope is less than 0.125 % in the first kilometer. The number of modes does not change when the field is calculated at the array and at the source. The mean water depth between the source and the array is a good first approximation to the effective water depth of the range-dependent environment. The range estimates for the bathymetry points are accurate to within ± 0.080 km, whereas the depth accuracy is ± 0.5 m.

4.2.4 Sound Speed Profile

The sound speed profile (SSP) measured at the array is shown in Figure 4.4. The main fluctuations in the SSP occur in the areas where there is a significant gradient. This can be modeled by including an uncertainty in the depth at which the change in the SSP occurs. Table 4.4 is a partial listing of the data which corresponds to XSV6 from the experimental data. These data were obtained in close proximity to the array near the time of the data collection and are considered quite accurate. The profile of Table 4.4 was reduced to seven critical points, each with a depth, sound speed, and the corresponding ROU. This parameterization is listed in Table 4.5. The solid line in Figure 4.4 depicts the nominal profile. The sound speed profile in the water

Table 4.3. Bathymetry Data

Range (km)	Water Depth (m)
0	78
0.746	77
1.823	76
2.586	75
3.538	74
4.498	73
6.805	72
9.871	71

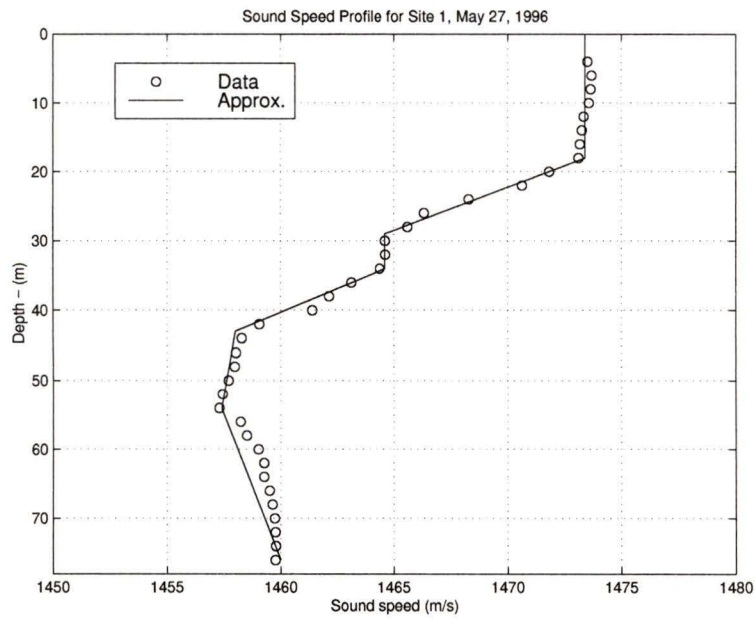


Figure 4.4. The measured sound speed profile at the array and its parameterization.

Table 4.4. *Sound Speed Profile at the Array*

Speed (m/s)	Depth (m)	Speed (m/s)	Depth (m)
1473.51	4	1458.28	44
1473.65	8	1457.97	48
1473.34	12	1457.44	52
1473.17	16	1458.23	56
1471.82	20	1459.01	60
1468.29	24	1459.26	64
1465.60	28	1459.63	68
1464.62	32	1459.76	72
1463.10	36	1459.76	76
1461.38	40		

column is very important in the calculation of the field, thus we are not satisfied with only one measurement and it should be included in the inversion.

In order to keep the number of variables to a minimum, the profile was further reduced to only two parameters: one which controls the depths, and the other which controls the sound speeds of Table 4.5. These parameters can have values between 0.0 and 4.0. If the depth controlling parameter is 0.0, then the lower depth bounds are used. If the value is 2.0, then the nominal parameters are used. If the parameter is set to 4.0, then the upper bound is used. All other parameter values are interpolated between the upper and lower bounds. A similar situation holds for the sound speed parameter. A more reliable way of parameterizing the sound speed is to use excitations of empirical orthogonal functions obtained from many measurements made in the area of the experiment [9]. Since there is only a single measurement, and its accuracy is assumed to be fairly good, the results should be satisfactory with the simpler technique.

4.2.5 Bottom Properties

In this section the bottom properties obtained from previous studies of the experimental site are summarized. The bottom is assumed to be composed of a single layer

Table 4.5. *Parameterized Sound Speed Profile and Range of Uncertainty*

Parameter	NOM	LB	UB	Units
depth1	0	0	0	m
ss1	1473.4	1472.4	1474.4	m/s
depth2	18	16	20	m
ss2	1473.4	1472.4	1474.4	m/s
depth3	30	28	32	m
ss3	1464.6	1462.6	1466.6	m/s
depth4	34	32	36	m
ss4	1464.6	1462.6	1466.6	m/s
depth5	43	39	46	m
ss5	1458.0	1454.0	1462.0	m/s
depth6	54	50	58	m
ss6	1457.4	1455.4	1459.4	m/s
depth7	water depth			
ss7	1459.8	1458.8	1460.8	m/s

Table 4.6. *Nominal Bottom Parameter Values and Range of Uncertainty*

Parameter	Nominal value	Lower bound	Upper bound	Units
Sediment				
thickness (t)	25	10	40	m
p-speed (cps)	1725	1550	1900	m/s
s-speed (css)	175	0	350	m/s
density (ps)	1.9	1.7	2.1	kg/m ³
p-atten. (aps)	0.25	0.005	0.5	dB/m-kHz
s-atten. (ass)	1.5	0	3.0	dB/m-kHz
Basement				
p-speed (cpb)	1975	1930	2020	m/s
s-speed (csb)	760	520	1000	m/s
density (pb)	2.2	2.0	2.4	kg/m ³
p-atten. (apb)	0.1	0	0.2	dB/m-kHz
s-atten. (asb)	0.25	0	0.5	dB/m-kHz

of sediment over the top of a bedrock basement.

The sediment thickness in this region is known to vary between 10 and 40 m. The nominal value is 25 m. The sediment is a mixture of sand and gravel. The relative amounts of each are not known, although their individual properties are available. The range of uncertainty and the nominal values for all the parameters that are relevant to ORCA can be found in Table 4.6. These estimates were obtained from [14]. No direct measurements of the shear attenuation have been made, but the bounds are estimated from typical shear attenuation in gravel and bedrock [1].

Some combinations of these parameters will violate the law of the conservation of energy which requires that

$$\frac{\alpha_s}{\alpha_p} < \frac{3}{4} \left(\frac{c_p}{c_s} \right)^2 \quad (4.1)$$

In this equation the shear attenuation is known to be the least important parameter [35]. Thus in order to satisfy the law of conservation of energy we can remove the

Table 4.7. *Doppler Shift Due to Source Movement at 3.2 m/s*

Frequency (Hz)	Downshift (Hz)	Shifted Frequency (Hz)
35	0.08	34.92
55	0.12	54.88
85	0.19	84.81
110	0.24	109.76
190	0.42	189.58
270	0.59	269.41
350	0.77	349.23
430	0.95	429.05
510	1.12	508.88
590	1.30	588.70

shear attenuation from the inversion and use the relation

$$\alpha_s = \frac{\alpha_p}{200} \left(\frac{c_p}{c_s} \right)^2 \quad (4.2)$$

which provides realistic shear attenuations for the nominal parameter values.

4.2.6 Source Signature

The source consisted of a moving-coil projector (MCP) and a barrel-stave projector. Continuous-wave frequencies of 35, 55, 85, 110, 190, 270, 350, 430, 510, and 590 Hz were projected in the experiment. The first three were generated by the MCP and the remaining seven were projected from the barrel-stave. The source levels remained relatively constant over the full length of the track and were monitored by fore and aft accelerometers. The variation in the projector output was less than 1%. The Doppler shift for each of the frequencies, summarized in Table 4.7, is due to a ship speed of 3.2 m/s. This is the speed relative to the array when the ship is moving on the radial. At the beginning of the tow the ship is moving at a rate of only 2.6 m/s relative to the array and thus the Doppler shift is lessened slightly.

The received signal is a time series with a sampling rate of 2048 samples/s. A typical received time-domain signal is shown in Figure 4.5. The large spikes are mainly of low frequency (<10 Hz) and have little effect on the signal.

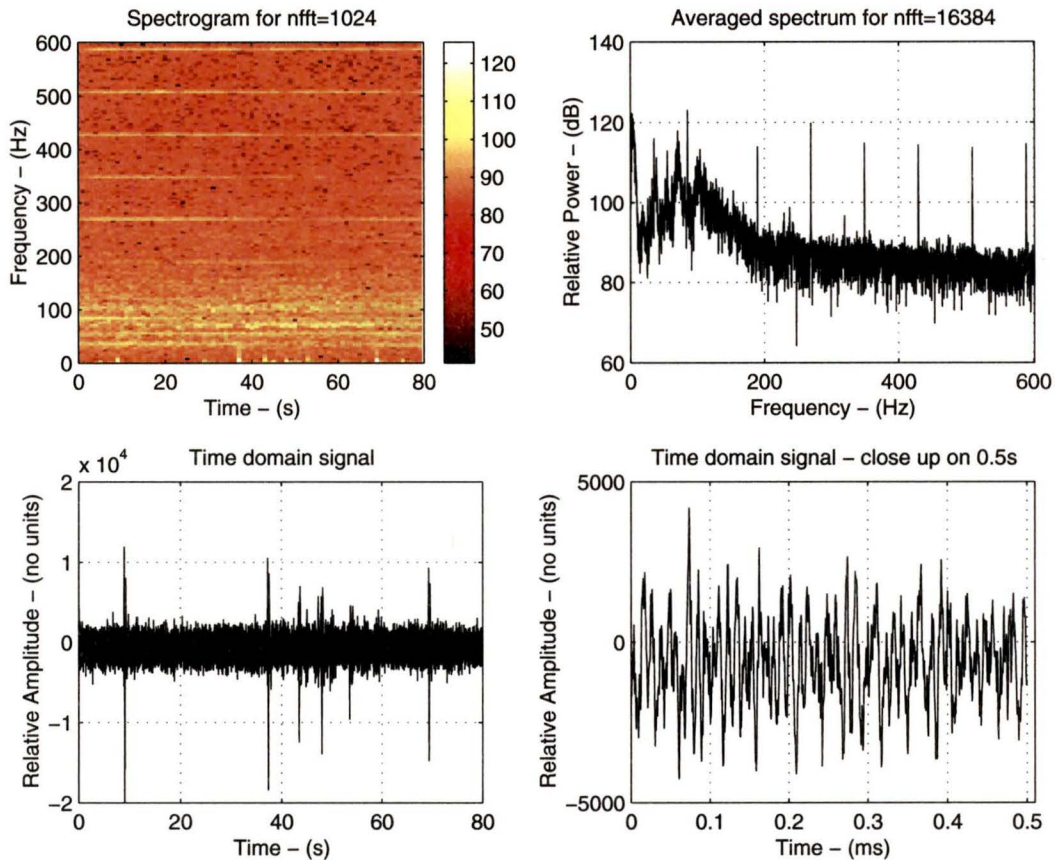


Figure 4.5. Spectrogram, averaged periodogram and time domain signal for the first hydrophone for a source range of approximately 0.5 km.

Figure 4.5 also shows the spectrogram for the received signal at the topmost sensor. The source range at this point is approximately 0.5 km. The spectrogram shows that there is considerable low-frequency noise which is centered at approximately 70 Hz. This is believed to be shipping noise. Inspection of the higher frequencies illustrates the fading effects as the source moves out in range. The averaged periodogram demonstrates that the high-frequency signals have a very good signal-to-noise ratio (SNR). The SNR for the 350 Hz signal is almost 30 dB. On the other hand, the low-frequency signals are greatly affected by the low-frequency noise. The SNR for the 110 Hz signal is 0 dB.

The plots in Figure 4.5 were made without concern for the stationarity of the received signal. The Fourier transform and the averaged periodogram require that the

signal be stationary. Since the ship is moving away from the array, the environment through which the transmitted signal moves is changing. We have assumed that the environmental parameters such as water depth and properties do not change over a time period of a few seconds. This is generally a good assumption. The moving ship, however, causes the range to change. We can use ORCA to see the effect at each frequency. Figure 4.6 shows the transmission loss for the bottom hydrophone for the first 500 m of the experiment using the nominal parameter values. The figure shows that the signal is only stationary over a distance which is of the order of one wavelength. If the total time (and in this case range) over which the Fourier transform is taken is much greater than the wavelength, then the averaged periodogram will yield spectral components that are not correct. The following procedure can be used to ensure that the spectral components are estimated for a stationary signal. This is pictured in Figure 4.7.

- For each frequency do:

- 1 Set the $SPAN = KF_s\lambda/v_s$ where v_s is the speed of the ship relative to the array in m/s, K is an arbitrary constant less than 10, $\lambda = c_s/f$ is the wavelength in meters, and c_s is the average compressional-wave speed in the water in m/s. In order to improve FFT efficiency set $SPAN$ equal to the nearest power of 2.
- 2 Set $NFFT = F_s/\Delta F$ where F_s is the sampling frequency and ΔF is the desired frequency spacing. The frequency spacing should be set quite low in order to have a suitable resolution. If $NFFT > SPAN$ then set $NFFT = SPAN$.
- 3 Set $OVERLAP$ to the desired value. The larger the overlap, the lower the variance in the final estimate will be since the number of snapshots will increase. The overlap should not exceed 50 % of $NFFT$ to avoid correlation of the noise between snapshots.
- 4 The number of snapshots can then be calculated as $k = \text{fix} \left(\frac{SPAN-OVERLAP}{NFFT-OVERLAP} \right)$.
- 5 Calculate the discrete Fourier transform of the windowed snapshot for each hydrophone and put the complex result for frequency f into the column vector $\mathbf{p}(f)$.
- 6 Form the cross-spectral density matrix (CSM) for each snapshot as $\mathbf{R}(f) =$

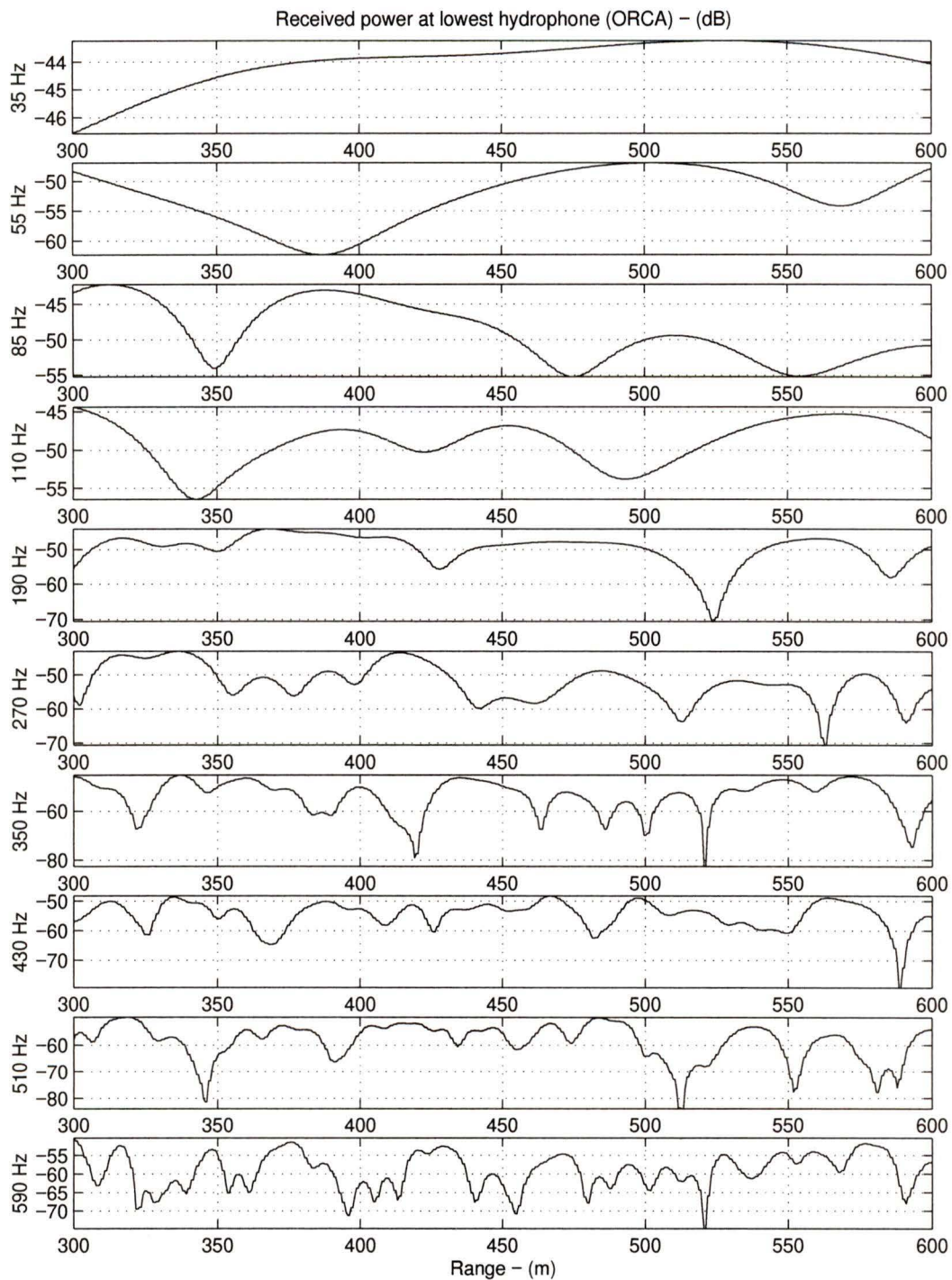


Figure 4.6. The received signal power over range generated by ORCA for ten frequencies. The total range spanned is 300 m.

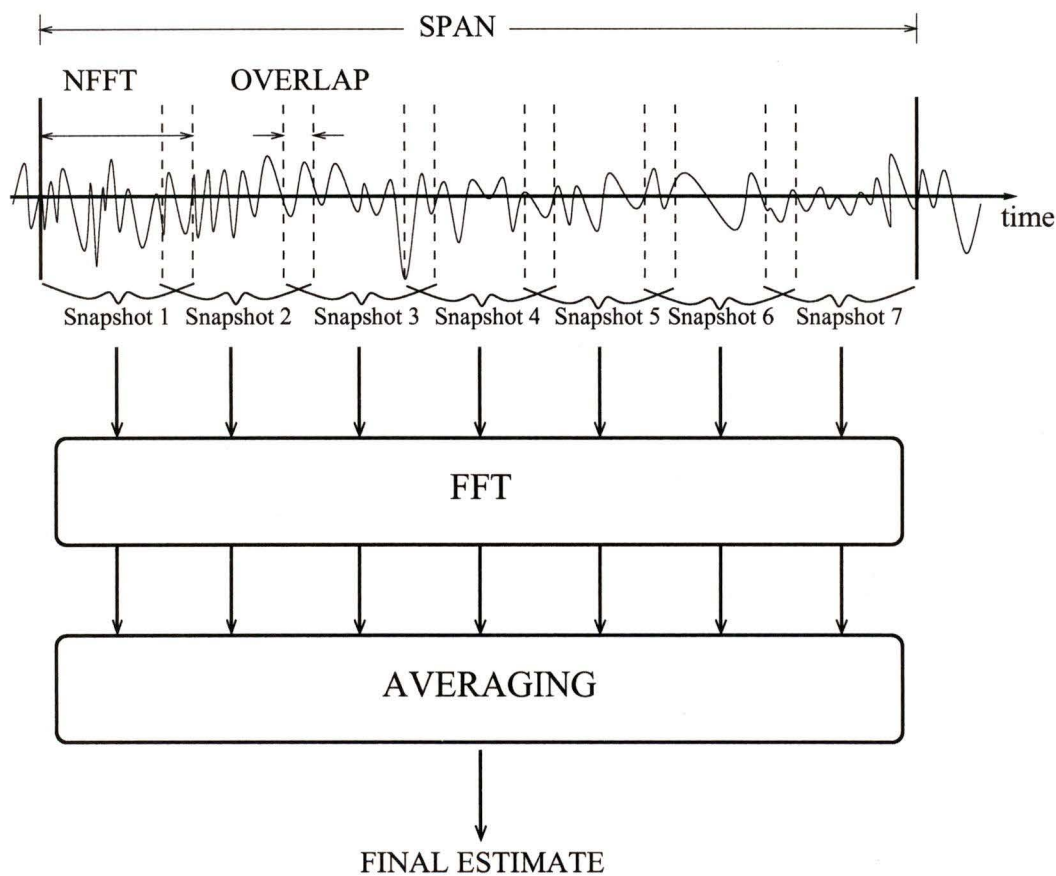


Figure 4.7. The computation of spectral estimates using the averaged periodogram.

$$\mathbf{p}(f)\mathbf{p}^\dagger(f).$$

7 Form the average CSM over all of the snapshots $\mathbf{R}(f) = \langle \mathbf{p}(f)\mathbf{p}^\dagger(f) \rangle$.

This technique reduces zero-mean noise, and thus the variance of the estimate, by averaging. The cross-spectra are averaged, rather than the pressure vector itself, because the cross-spectra are immune to absolute changes in the phase. It should be noted that this type of spectral estimator is biased. The variance can only be reduced at the cost of an increase in the bias [20]. In order to reduce the variance as much as possible k should be large. The variables *SPAN* and *NFFT* control this value. The *NFFT* must be large in order to increase the resolution of the operation, but a lower value would increase k and thus there is a tradeoff. The variable *SPAN* is constrained by the sampling rate and the ship speed. We would like to have *SPAN* to be as large as possible to increase k thus the sampling rate should also increase and the ship speed must decrease. In future experiments it would be desirable to have either a stationary source or a slow ship speed and a high sampling rate in order to obtain stable spectral estimates.

The results of using this spectral estimation process on the near range data are shown in Figure 4.8. This experimental result should be compared with the modeled data of Figure 4.6. The features seem to be similar for each frequency although they are not identical. This discrepancy is due to noise and mismatch; determining a more accurate environment will, in principle, reduce the differences.

The spectral estimation parameters used for each frequency are listed in Table 4.8. The high-frequency band uses no averaging because the *SPAN* is too short to allow averaging and a suitable resolution. The cross-spectra estimated without averaging may have a significant error associated with them but it is not known what effect this has on the calculation of the Bartlett power. Ideally the CSMs should be estimated with at least as many snapshots as sensors to ensure that the matrix is of full rank. In this experiment it is impossible to do this without sacrificing resolution. An estimate of the signal-to-noise ratio in dB is also given in the last column. This is calculated using the total noise power in a 20-Hz band on either side of the frequency in question. The estimates are conservative.

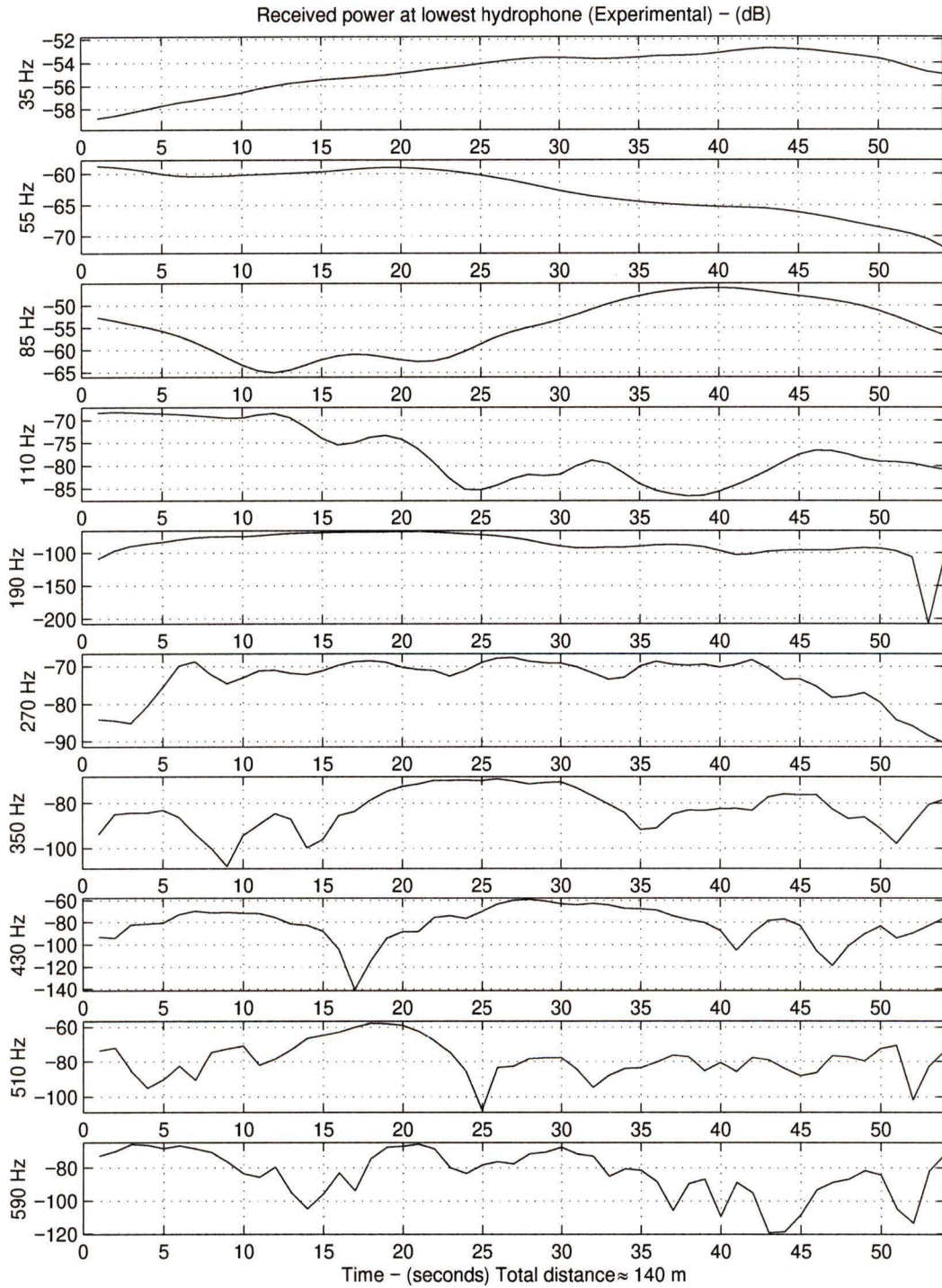


Figure 4.8. Experimental spectral components estimated with a resolution of ≈ 3 m. The total range spanned is one-half that of Figure 4.6.

Table 4.8. *Spectral Estimation Parameters*

Frequency (Hz)	λ (m)	<i>SPAN</i>	<i>NFFT</i>	<i>OVERLAP</i>	<i>k</i>	SNR(dB)
35	42	65536	16384	8192	7	15
55	27	40960	16384	8192	4	11
85	17	26624	16384	8192	2	18
110	13	20480	16384	8192	1	0
190	8	12288	12288	0	1	22
270	5	8192	8192	0	1	32
350	4	6144	6144	0	1	27
430	3.4	6144	6144	0	1	27
510	3.0	4096	4096	0	1	27
590	2.5	4096	4096	0	1	27

4.2.7 The ORCA Environment

ORCA is a normal mode code developed for range-independent viscoelastic environments [19]. This model was described in detail in Chapter 2. The assumptions for this analysis are: range independence, 7 water column layers, a tilted array, a single homogeneous sediment layer, and a homogeneous halfspace basement. The range of uncertainty has been widened for some parameters which had relatively few previous estimates, such as compressional speed in the basement. Table 4.9 summarizes the *a priori* parameterized environment for this experiment. Figure 4.9 depicts this environment.

4.3 Simulation Study

The incoherent average of the Bartlett mismatch function (Equation 2.43) can be used to assess the sensitivity of the field to each of the parameters. Traditionally the sensitivity is defined as

$$S_{\xi}^p = \frac{\partial p(\xi)}{\partial \xi} \quad (4.3)$$

This definition states that the sensitivity is high if a small change in the environment ξ results in a large change in the pressure field p .

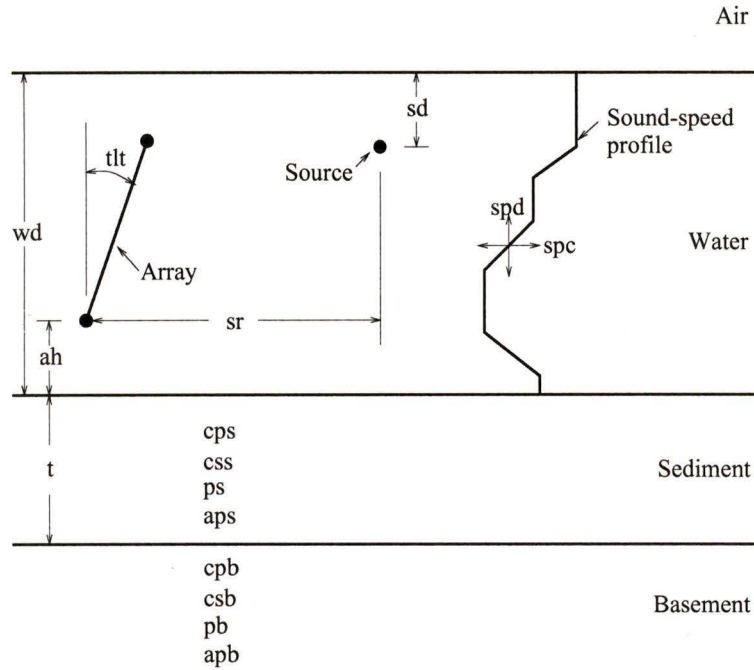


Figure 4.9. The parameterized ORCA environment.

Analytical expressions to calculate the pressure field gradients are not readily available for forward models such as ORCA. Thus Equation 4.3 is not a useful measure of the sensitivity. Rather than discuss the sensitivity of the field to small changes in the parameters, we would like to propose a simpler measure of the sensitivity for each parameter based on the *a priori* information and the Bartlett processor.

The sensitivity of a parameter seeks to measure the significance of the parameter to the calculation of the acoustic field. It is difficult to do this for the general case since each experiment samples the field in a different way which affects the sensitivity calculation. Since the nominal parameters and the range of uncertainty over which we wish to search have already been defined, a practical and experiment-dependent measure of the *importance* I of the l th parameter for the calculation of the field would be

$$I_{\xi_l} = \max [M_B(\xi_l)] \tag{4.4}$$

where $M_B(\xi_l)$ is the Bartlett mismatch calculated over the entire range of uncertainty for the parameter ξ_l with all of the other parameters fixed at their nominal values. Figure 4.10 shows the Bartlett mismatch for changes in each parameter. The Bartlett

Table 4.9. *Nominal a priori Parameter Values and Range of Uncertainty*

Parameter	NOM	LB	UB	Units
(s) - sediment				
(b) - basement				
array height (ah)	6	1	11	m
array tilt (tlt)	0	-10	10	degrees
source depth (sd)	32	28	36	m
source range (1 km data) (sr)	1.0	0.9	1.1	km
water depth (wd)	78	76	80	m
sspd (spd)	2	0	4	no units
sspcp (spc)	2	0	4	no units
thickness of sediment (t)	25	10	40	m
p-wave speed (s) (cps)	1725	1600	1850	m/s
s-wave speed (s) (css)	175	0	350	m/s
density (s) (ps)	1.9	1.7	2.1	kg/m ³
p-wave attenuation (s) (aps)	0.25	0.005	0.5	dB/m-kHz
p-wave speed (b) (cpb)	1975	1930	2020	m/s
s-wave speed (b) (csb)	760	520	1000	m/s
density (b) (pb)	2.2	2.0	2.4	kg/m ³
p-wave attenuation (b) (apb)	0.1	0	0.2	dB/m-kHz

mismatch is averaged incoherently over each frequency as in Equations 2.50 and 2.43. In the plot for the shear speed in the sediment there is a discontinuity when the shear speed is set to zero. This is an artifact of ORCA and was ignored in the calculation of the importance measure. In all subplots the power is zero when there is a perfect match. When the power is close to unity, the mismatch is maximized. The same plots were generated using a nominal range of 500 m rather than 1 km. The uncertainty in the range was maintained at ± 100 m. The results were quite similar for both ranges and are tabulated in Table 4.10.

Not only is the importance measure dependent on the ranges of uncertainty but it also depends on the nominal values. The parameters have been grouped according to their relative importance. The most important parameters are in Group I; while

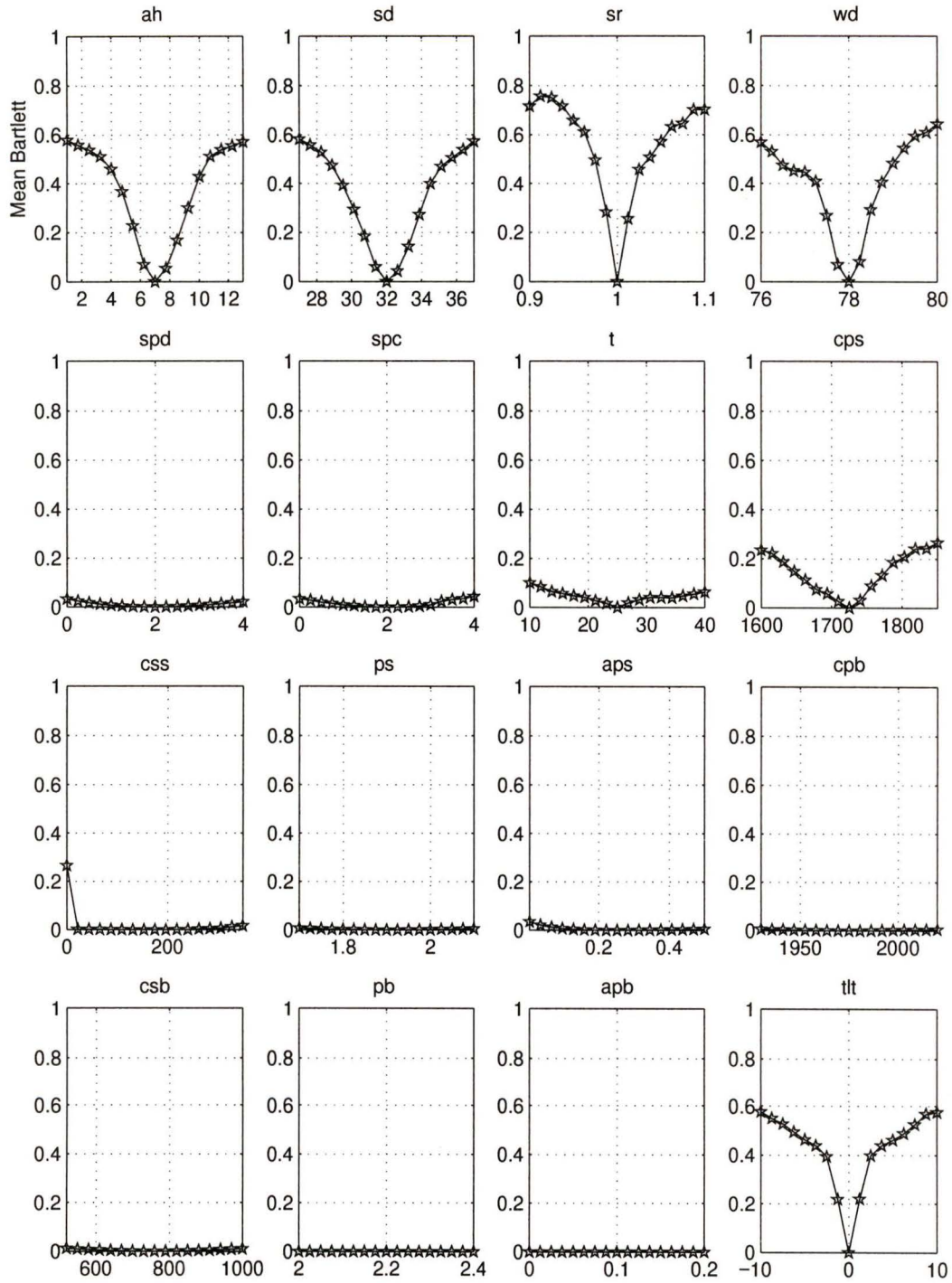


Figure 4.10. Simulation study for the nominal parameter environment. Each parameter is varied while the other parameters are held constant at their nominal value and the Bartlett mismatch of Equation 2.43 is calculated.

Table 4.10. *Relative Importance of Parameters*

	Parameter	I_{ξ} 1 km	I_{ξ} 500 m
GROUP I	source range, (sr)	1.0	1.0
	water depth, (wd)	0.85	0.65
	source depth, (sd)	0.77	0.75
	array tilt, (tlt)	0.76	0.76
	array height, (ah)	0.76	0.88
GROUP II	p-wave speed (s), (cps)	0.35	0.34
	thickness of sediment, (t)	0.13	0.14
GROUP III	sspcp, (spc)	0.059	0.040
	p-wave attenuation (s), (aps)	0.047	0.025
	sspd, (spd)	0.043	0.055
	s-wave speed (s), (css)	0.022	0.0067
	s-wave speed (b), (csb)	0.016	0.013
	density (s), (ps)	0.011	0.0045
GROUP IV	p-wave speed (b), (cpb)	0.0077	0.011
	density (b), (pb)	0.00036	0.00034
	p-wave attenuation (b), (apb)	0.00026	0.00056

the least important parameters are in Group IV. This grouping is done to differentiate between parameters that are essential for modeling the field and those for which precision is not necessary. It is significant that despite the fact that the geometry of the experiment was the only directly measurable quantity, Group I consists entirely of these parameters. The order of magnitude and thus the grouping of the parameters remains the same for slightly different nominal environments. The difference in ranking between the 1.0-km and 0.5-km ranges is not significant. No parameter strayed far from its original placing.

From these groupings we can safely say that the Group IV parameters are not likely to be estimated accurately from this experiment. The Group III parameters may be estimated but their importance is so low that estimation will be difficult. A more positive way of looking at this problem is that we have enough knowledge of the Group III and IV parameters to accurately model the field, so we should turn our attention to the Group I and II parameters. Consequently, at this point the Group III and IV parameters are held at fixed values, leaving only seven parameters to be estimated.

4.4 Source Localization with Experimental Data

The nominal environmental model was applied to an inversion to determine the source location using matched-field processing. Figure 4.11 shows the Bartlett surfaces for each frequency as well as the incoherent mean of all of the surfaces except for the 110 Hz surface which was omitted due to its low SNR. The source should be localized within the box shown at a range of approximately 0.5 km and a depth of 35 m in each plot. As can be seen, the source is not correctly localized at any single frequency. The incoherent average also shows only a broad local maximum near the true source location. There are two main peaks at 0.30 km and 0.55 km in range whose depths are 60 m and 40 m respectively. The global maximum is at the 0.30 km point and the Bartlett power is 0.42. The localization is poor for a number of reasons. The uncertainty in water depth, array tilt, and array height is significant enough to cause the Bartlett peak power to be reduced. The correlation between any of the unknown parameters and the range or depth can cause the peak to shift. The higher frequencies

Table 4.11. *Result of Grid Search of Group I*

Parameter	NOM	Step Size	Units
source range (0.5 km data)	0.470	0.010	km
water depth	77.67	0.33	m
source depth	34.1	0.7	m
array tilt	0.0	1.7	degrees
array height	8.8	0.8	m

have high variance in their estimates and this source of error is worsened by the high resolution of the higher frequencies. This is evident from the low peak power in the high frequencies.

In order to improve the localization a coarse grid search was done over the most important parameters (Group I) using all eight of the frequencies in the cost function. Five points within the range of uncertainty were chosen for each parameter for a total of 3125 points. The incoherent mean of the Bartlett power over all eight frequencies was calculated. The best point found was then used in another search with the same number of points but a smaller step size. The best point obtained is shown in Table 4.11. In order to do an exhaustive grid search with 25 points per parameters approximately 10^{17} forward models would be required. With the grid search done here we only wish to determine the coarse features of the ambiguity hypersurface and the five most significant parameters.

The Bartlett surfaces for the new environment are shown in Figure 4.12. These surfaces show that the source is localized with much less ambiguity than with the nominal environment. The 55-Hz surface in particular has a main peak in the correct location with a Bartlett power of 0.94. This is a 15 % improvement relative to the nominal parameter values. The 35 Hz surface also has a peak in the correct region; however, there are numerous sidelobes with the same power. The peak of the averaged surface has a power of 0.44 and is located just outside of the feasible region.

A one-dimensional search for the optimum value of each parameter is shown in Figure 4.13. In each of these plots a single parameter is varied and the remaining parameters are fixed at values obtained from the grid search. The plots show that each parameter is at or near a maximum whether we consider the low-frequency band

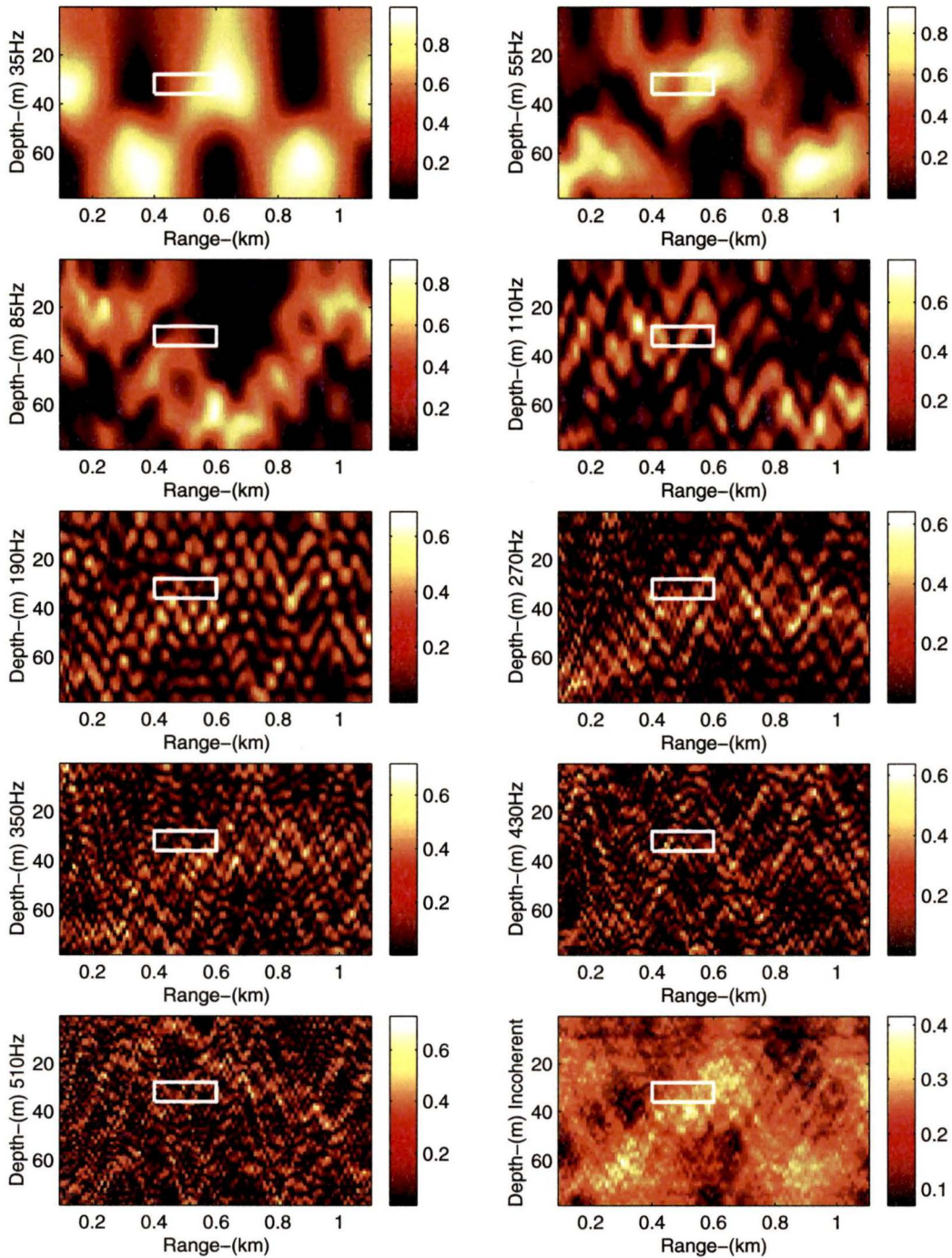


Figure 4.11. *The Bartlett surfaces for nine separate frequencies and the incoherent average of the surfaces (with the exception of the 110 Hz surface) show that for the nominal parameter values source localization is unsuccessful.*

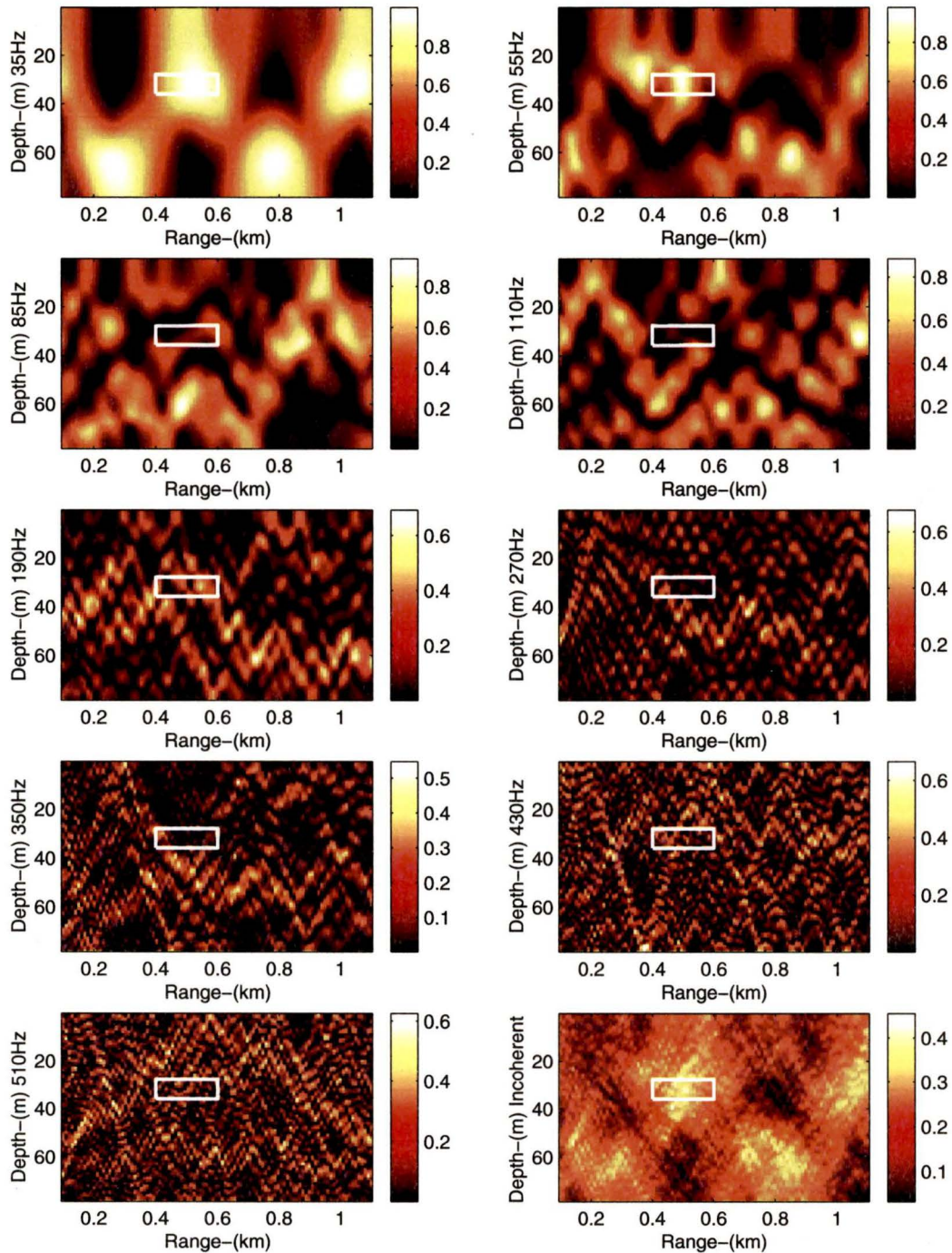


Figure 4.12. The Bartlett power for all frequencies and the incoherent mean of these surfaces show that the source localization is improved with the new parameters. The 55 Hz surface is the most influential with a peak power of 0.94.

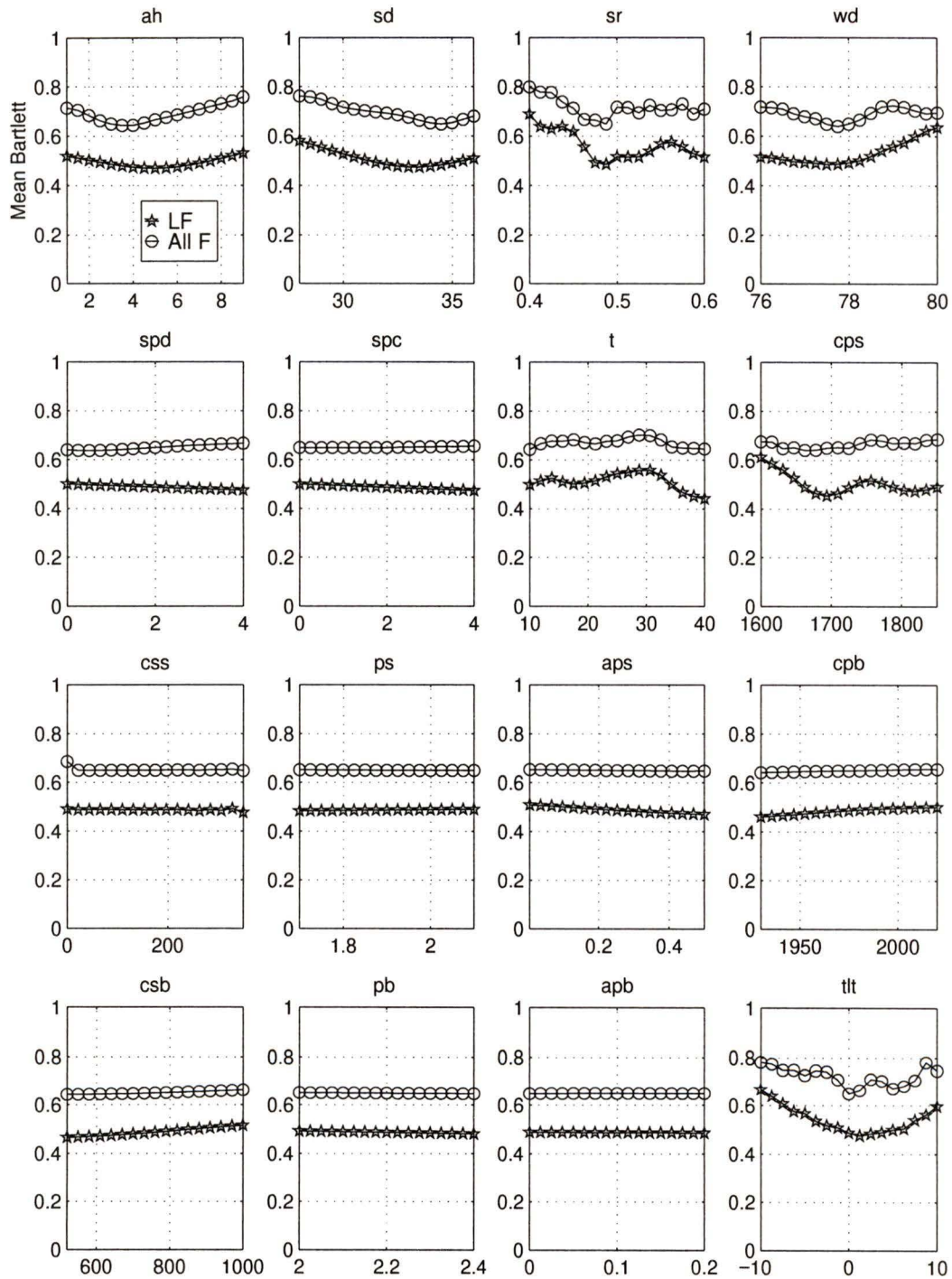


Figure 4.13. The Bartlett power averaged incoherently for all frequencies and for the low frequency band (35, 55, 85 Hz) shows that the most important parameters are at or near their peak values for the new parameters.

of 35-85 Hz or all of the frequencies. The least important parameters have no effect on the Bartlett power and are flat. The Group II parameters show a multimodal pattern and will be the main focus of the neural inversion efforts.

4.5 Conclusions

In this chapter we summarized the results of previous studies of the experimental area and have developed a set of environmental parameters suitable for use with the ORCA model. The sensitivity study and the grid search have isolated the parameters which we wish to determine using the neural inversion technique. The compressional speed in the sediment and the sediment layer thickness are the most important of these parameters. From this point on we will consider the values of Group I to be fixed and concentrate on the Group II-IV parameters. This is done because it is believed that the Group I parameters are known with sufficient accuracy and that there is no significant correlation between the Group I parameters and the less important ones.

Chapter 5

Results and Discussion

In this chapter we describe the results of training the backpropagation and radial basis function neural networks. The purpose of the training is to approximate the inverse of the ORCA forward model and to use the network to estimate ocean model parameters from the experimental data. The output of the neural network is not useful unless an estimate of the network's estimation accuracy is also obtained. The network error is evaluated by inverting a test set of data which is as similar to the experimental data as possible. The test set must contain all of the noise and mismatch that is present in the experimental data otherwise the error estimate will be optimistic.

A large number of decisions are made respecting training and test set sizes, training parameters, and pre-processing; for this reason a simple case of range-finding without mismatch or noise is examined in detail. The extended results which include mismatch, additive noise, and multiple environment parameters are then presented. Finally the performance of a robust neural bottom classifier is evaluated.

5.1 Backpropagation Results

In this section we use the backpropagation algorithm of Chapter 3 as the core for a larger algorithm which uses multiple starting points and early stopping (cross-validation). The training parameters which we must determine are:

1. Training and test set size.
2. Number of network inputs (principal components).
3. Number of neurons in the hidden layer.
4. Number of initial starting points.

5. Number of epochs, error goal, momentum, and learning rate for initial training.
6. Number of epochs, error goal, momentum, and learning rate for final training.

Several other options can improve the convergence and final result including:

1. Pre-processing of inputs and outputs.
2. Randomizing the order of presentation within the training set.
3. The error measure.
4. The effect of cross-validation.

The influence of each of these factors on the network estimation error is evaluated for the simple case of range-finding in the absence of noise or mismatch. The changes which must be made to the network when noise and mismatch are added are examined. These results are then extrapolated to help design a network which can be used with the experimental data to estimate the less important bottom parameters.

5.1.1 Range Finding

In this section we present the results of the design of a network which seeks to localize a source in range. In some cases the source location may not be known but the environmental parameters are. An example of this would be a submarine detection and localization system. In the TRIAL SABLE experiment the location of the source is known *a priori* and we are using this exercise as a validation for the neural inversion technique.

The training and test data are generated using ORCA with the environmental parameters fixed and the range varying. The output of ORCA is the complex pressure at the array for each frequency. After these pressures are generated, zero-mean white Gaussian noise is added to each pressure so that the signal-to-noise ratio (SNR) is the same at each sensor in the modeled data as it is in the actual data. In order to maintain simplicity the noise added is the same for each hydrophone but differs for each frequency using the SNRs of Table 4.8.

In Chapter 4 we mentioned that the relative pressure must be used to account for the lack of source information. To remove the source level each frequency is normalized to the first hydrophone, although any hydrophone could be used. This removes the dependency on the source level. The dimension of the data is reduced

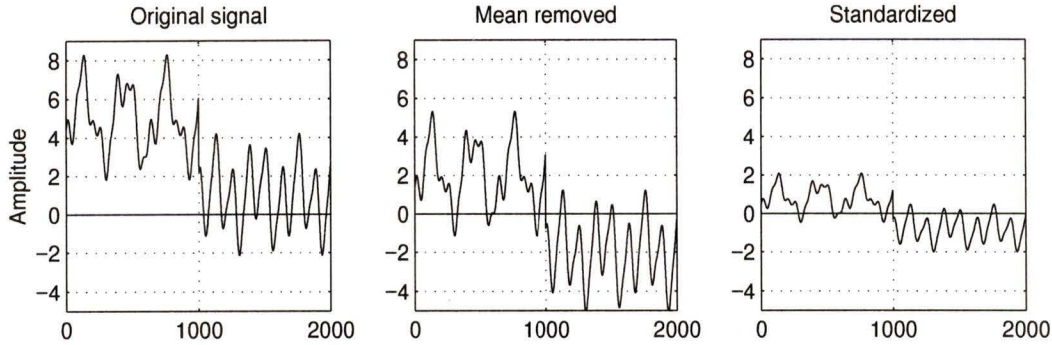


Figure 5.1. *The effect of standardization. (a) the original signal, (b) the signal with the mean removed, (c) the signal with the standard deviation set to unity. The shape of the signal is preserved, whereas the effect of possibly unknown gains and offsets has been removed. The x-axis indicates the sample number.*

to 80 (10 relative pressures and 8 frequencies, the 110 Hz frequency is omitted due to low SNR). It should be emphasized that the TL is used as the input to the neural network rather than the full-field information.

These 80 elements are compiled into the super vector presented in Chapter 2. This is the base data set from which the pre-processing proceeds. After this point the replicas are as close to the experimental data as is possible and each succeeding step is applied to both the modeled and experimental data identically. The correlated noise which may be present in the signal is very difficult to model, and thus remains a source of error for this technique.

It is known that representing the inputs to the network as excitations of empirical orthogonal functions can improve the convergence of the backpropagation algorithm [27]. The principal components analysis suggested in Chapter 3 provides an efficient method of implementing this representation. Before applying PCA the data must be standardized by removing the mean of each measurement in the training set and dividing by the standard deviation. The effect of this transformation is shown in Figure 5.1. This transformation is applied to each row of the training set (refer to Equation 3.4). The mean and standard deviation of each row of the training set is retained and then removed in the same way from the test set and the experimental data. The assumption is that the test set and the experimental data have the same distributions with similar means and standard deviations, thus the training set should

Table 5.1. *Initial Training Parameters and Conventions*

training set size	200
test set size	200
mismatch	no
additive noise	no
normalization	yes
$\log_{10}(p)$	yes
standardization	yes
# of principal components	40
output scaling	yes
# neurons in hidden layer	10
# starting points	1
stopping criterion	number of epochs=300
randomization	yes
learning rate	0.01
momentum	0.9
output neuron type	linear
hidden neuron type	tanh

be representative of the experimental data and be sufficiently large that the estimates are stable.

Principal components analysis as described in Chapter 3 is performed on the training inputs. The transformation matrix is retained and applied to the test and experimental data. Each row of the training data is uncorrelated and orthogonal. After transformation the test set does not share this property; however, since the transformation is linear no information is actually lost until the thresholding is performed. If the test set is sufficiently different than the training set then significant errors can occur when thresholding. The test set error would be reduced by including all of the principal components.

In order to evaluate the effects of pre-processing, the training parameters, and the network architecture, we adopted the standard training set-up of Table 5.1. The training set and test set were generated using ORCA. The ranges were between 0.1

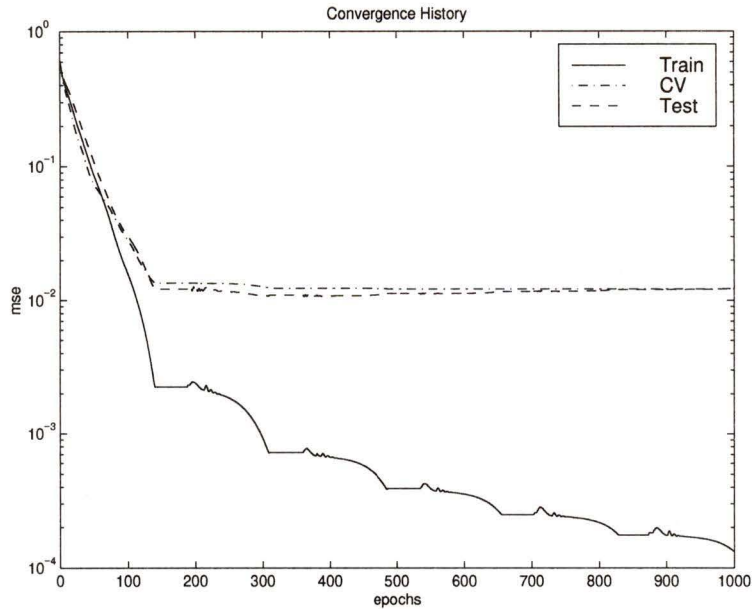


Figure 5.2. *The convergence history for the training, test and cross-validation data sets shows that the error for the cross-validation and test sets after 1000 epochs is the same. The training set error is much lower. The asymptotic error for the CV and test sets is determined by the network size, the starting point, and the sampling density.*

and 1.1 km with 600 evenly spaced points. The other environmental parameters were held fixed at their nominal values. The training set used every third range point, the test set used the remaining 400 points; of these, 200 were used for cross-validation and 200 for testing.

The convergence history for the mean-squared error as the backpropagation algorithm progresses is shown in Figure 5.2. This figure shows that the training set error decreases steadily whereas the CV and test sets decrease to a constant value after which no improvement is seen. The small bumps in the error history are due to the momentum term. The larger humps where it seems to flatten out are due to the adaptive learning rate. In this run, cross-validation was not used and the network was stopped after 1000 epochs had been reached. The characteristic increase in the test set error due to memorization was not evident here. This is perhaps due to a fortuitous choice of network size; the network was too small to memorize the training data and thus generalized well even for very low errors in the training set. In

this instance the only advantage of cross-validation is that training would have been stopped at approximately 150 epochs rather than 1000.

The network output for the training and test sets is shown in Figure 5.3. The error is very low ($mse=0.00013$) in the training set, and considerably higher, but still acceptable, in the test set ($mse=0.0122$). The cross-validation set is almost identical to the test set. The stopping criterion (e.g., number of epochs, error goal, or cross-validation) should be chosen judiciously to ensure that the network is not stopped in an area where the test set error is high. From Figure 5.2 a good stopping time would be after 150-200 epochs. Training for too many epochs is not risky in this case because the network is too small to memorize the training set. In this situation we desire the lowest possible error. An error goal slightly less than that seen in Figure 5.2 would be appropriate since a different starting point may result in a lower error, but there is always the risk that the goal will never be reached. Cross-validation removes the guesswork by stopping only after the lowest possible test set error has been reached. When momentum and an adaptive learning rate are used the training set error may not decrease monotonically and cross-validation will stop the network training at a sub-optimal point. Multiple starting points with cross-validation provide a way of sampling the local minima of the test set error function. The network found with the lowest error is then chosen as the final network. The number of different starting points is determined by trial and error. If cross-validation is used as the stopping criterion with multiple starting points then further training is not required. This technique is used for the remainder of this section. The number of starting points is set to the somewhat large value of 100 to increase the likelihood of finding a satisfactory end point (low test set error).

The most important variable in achieving an acceptable test set error is the sampling density. In this instance 200 samples were used for the range of 0.1 to 1.1 km. If fewer are used the result is as shown in Figure 5.4. This network uses only 100 samples in the same range and the error has increased by an order of magnitude to 0.185. The error in the training set is effectively zero. The lower mse for the sparse training set is because there are fewer examples to learn and the approximated function is smoothed relative to the more densely sampled example. The smoothness of the approximated function based on the sparse training data is greater than the

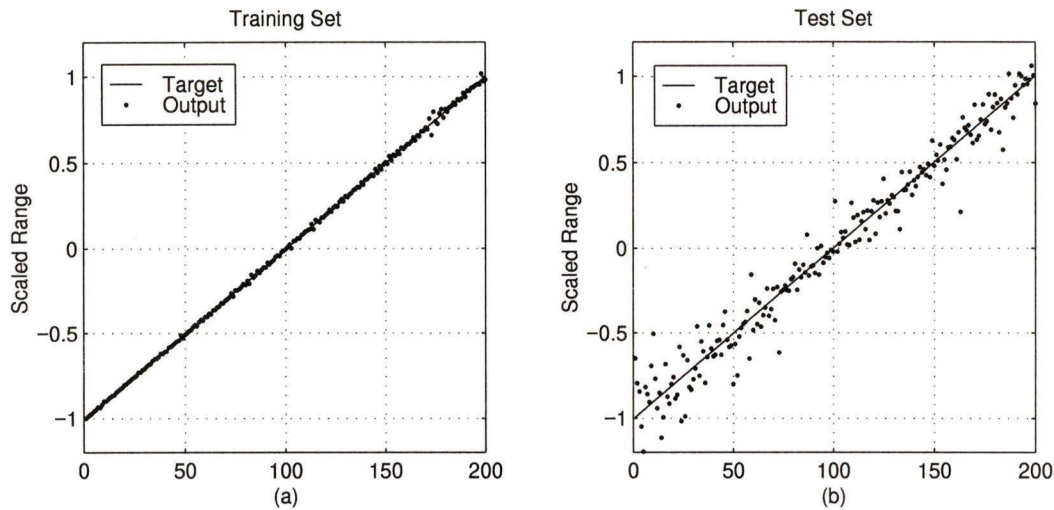


Figure 5.3. *The output of a network for both the training and test sets. The test set shows an increase in error relative to the training set. This error can most easily be reduced by increasing the size of the training set to more than 200 samples.*

actual smoothness of the inverse mapping, thus there is an increase in the test set error.

The next step is to evaluate the effect of the pre-processing. To do this accurately we should design several networks and take the average final mse for the test set. This must be done since the initial weights are always chosen randomly and the expected value is more indicative of the true performance of a particular configuration. A single design may result in a network that converges to a poor local minima of the error function. The average of 100 networks designed with different starting weights is taken. The expected value of the mean-squared error for the default training parameters is 0.0140.

Figure 5.5 shows the effect of varying the number of principal components. The principal components used are always those with the highest variance. The test set error decreases along with the training set until 30 inputs are reached. At this point the error in the test set increases while the training set continues to decrease. As the number of inputs increases the number of weights is also increased. Once the number of weights is large enough the test set error will begin to increase because the network has become capable of memorizing the test set. The mse for 30 inputs is 0.0115. The

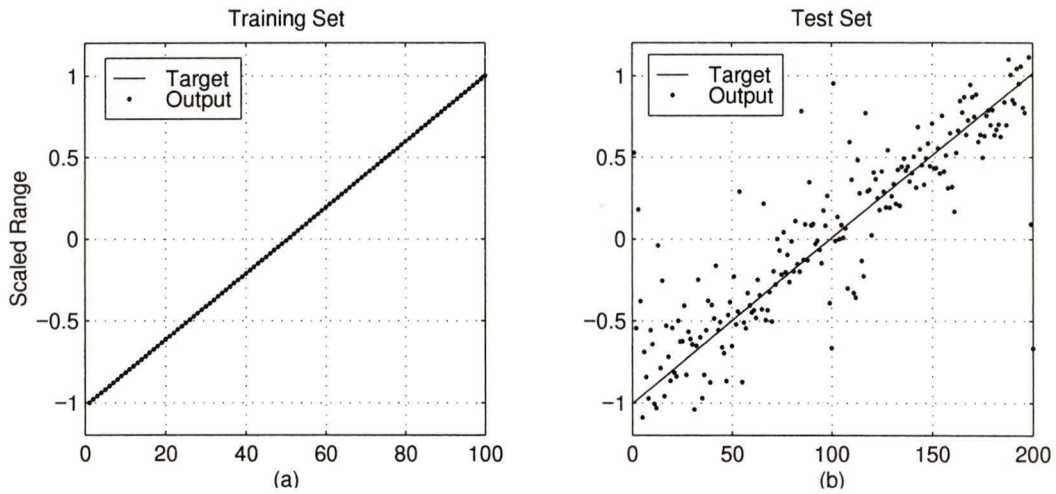


Figure 5.4. Training with an undersampled function results in poor generalization and a high error for the test set. This figure was generated with only 100 range samples between 0.1 and 1.1 km.

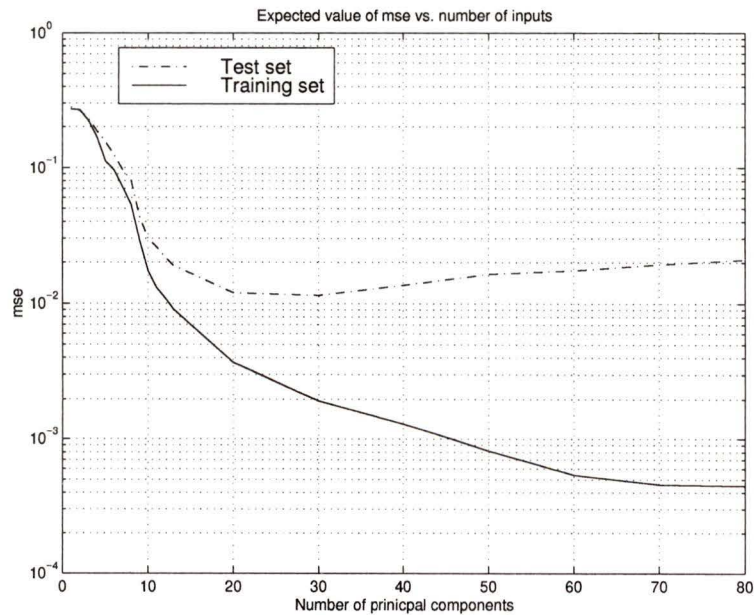


Figure 5.5. The mean-squared error vs. the number of inputs shows that the optimal number of inputs is approximately 30. This corresponds to a mse of 0.0115 for the test set.

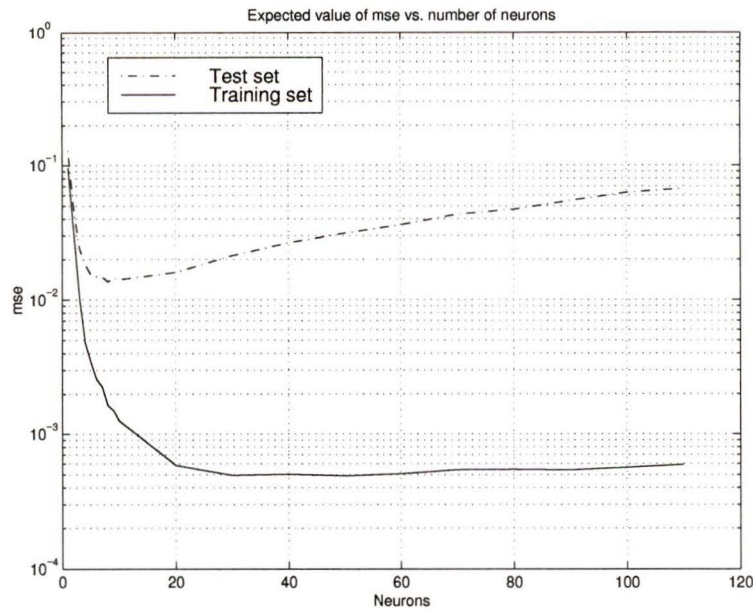


Figure 5.6. The mean-squared error vs. the number of hidden neurons shows that the optimal number of neurons is 8. This is a mse for the test set of 0.0136.

total number of weights at this point is 310.

Figure 5.6 shows the effect of varying the number of neurons in the hidden layer. The mse shows a marked increase for a network size greater than 10. This is again due to the increase in the number of weights and the corresponding poor generalization performance. The optimal error attained was 0.0136. The number of weights in this instance is 328. This is very close to the optimal number of weights used when the number of inputs was varied. We can conclude that any network with slightly more than 300 weights would learn the data and generalize properly provided that enough information about the output was provided by the inputs (e.g., a network with only one input would not likely have enough input diversity to predict the output behavior, even if it did have 300 weights).

The effect of the pre-processing steps were observed as well. If the inputs are not transformed using a logarithm, then the expected value of the test mse increases to 0.0384. The logarithmic transform smoothes the mapping from the data to range and thus a simpler network is required for good generalization. When the training inputs are presented in order of increasing range then the mse increases slightly to 0.0143. Since randomizing the presentation order does not increase the training time

and may have some advantages in terms of avoiding local minima [27] this procedure was used throughout.

The most significant decrease in test set error occurs when the output layer is changed from linear to the hyperbolic tangent activation function. This function ‘squeezes’ the output to a number between -1 and 1. Since the output is already scaled to lie in this range it is beneficial to use this function on the output layer. The mse of the test data decreases to 0.0099 when this is done.

The error was also evaluated when no principal components analysis was performed. All 80 inputs were used and the output mse attained was 0.0135. This is a slight improvement over principal components analysis but the number of weights used (810) with a similar performance makes it unattractive.

When all of the improvements in the pre-processing and training parameters were used, the expected value of the mean-squared error decreased to 0.0083. This corresponds to an accuracy of 46 m in range. When noise was added to the test set the expected value for the error increased to 0.197.

The test set error in the presence of noise was improved when the training data was also perturbed with similar noise; the test mse decreased, but was still unacceptable at 0.131. The number of inputs and the number of neurons may need to be increased to obtain reliable results in the presence of noise. Figure 5.7 shows that the number of neurons should be increased to be somewhere in the range of 10-30. Figure 5.7 was generated in an identical fashion to that of Figure 5.6 except that the training and test sets were perturbed with noise. The fact that the network requires more neurons to learn the mapping from the data to the range indicates that the mapping is less smooth than in the noise-free case. Figure 5.8 shows that the number of inputs is less important in the presence of noise. Any number between 10 and 40 approaches the best achievable error.

The effect of the noise is severe enough that the expected mean-squared error has increased from 0.0083 to 0.129 for 40 inputs and 30 hidden layer neurons. This corresponds to an increase in the estimation error from 46 to 180 m. The initial uncertainty in range for this example is 500 m so the network is improving the estimate, if only marginally. The advantage of using a larger network and more inputs to compensate for the noise is offset by the large increase in training time. The error can be reduced

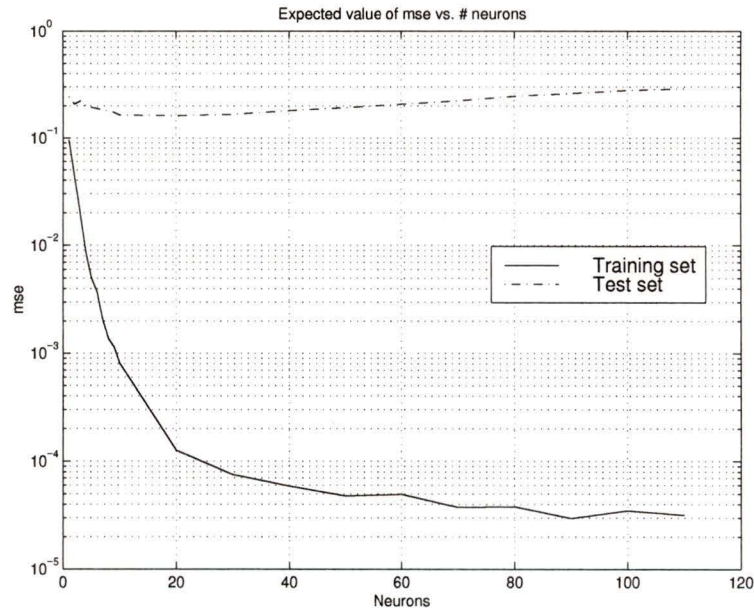


Figure 5.7. The mean-squared error vs. the number of hidden layer neurons in the presence of noise. Each point is the expected value of the final mse for 100 different starting weights. The noise has severely degraded the performance.

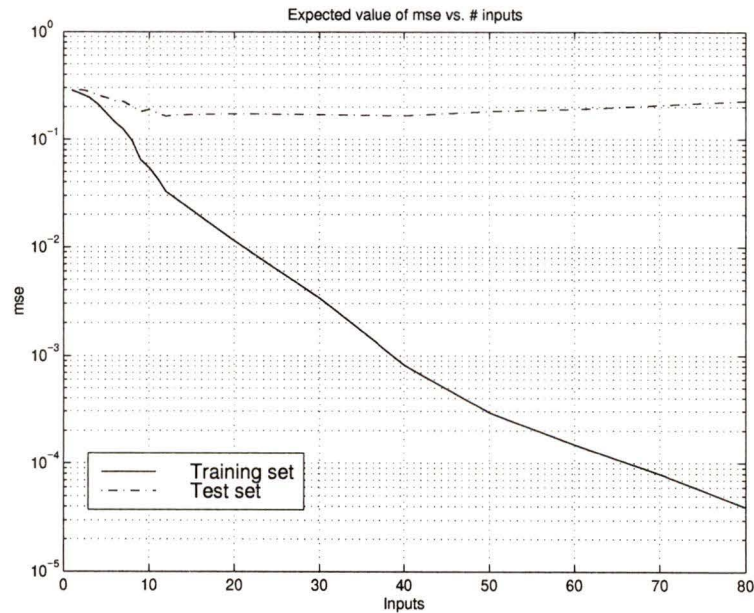


Figure 5.8. The mean-squared error vs. the number of inputs (principal components) in the presence of noise. Each point is the expected value of the final mse for 100 different starting weights. The noise has severely degraded the performance.

by increasing the size of the training set. The exact same training set was used, only a different noise perturbation was added repeatedly. The error using this technique and nine extra training sets was reduced to 0.054 which corresponds to an estimation error of 116 m. Further noise reduction could be achieved by adding the noise more times; but again it would be at the expense of a large increase in training time.

5.1.2 Range Finding with Mismatch

In the previous section only the range was varied, all of the other parameters were held constant. In the case of mismatch in the Group II–IV parameters we would like to determine how many more weights may be required to achieve a low error in range. The training set was chosen as before to be 200 points evenly spaced between 0.1 and 1.1 km. Since the target parameter space has increased in dimension to 12 the number of points required to account for the mismatch must increase to account for the added complexity. For each point in range 5 randomly chosen environments were used. The total number of distinct points was 1000. The number of training points was 500, the number of test points was 300 and the cross-validation set remained at 200 (one for each range but a separate environment for each). Noise was not added to either the training or the test sets. The number of inputs was set to 40 and the number of neurons in the hidden layer was 10. The network was trained for 100 separate starting points until the cross-validation error in range began to increase. The expected value of the error in the mismatch, noise-free case was 0.066 which was an absolute error of 128 m. This was a substantial increase in the error relative to the no mismatch case. The influence of mismatch was less than that of the noise in this instance. The error could be further reduced by increasing the sampling density of the training set; however, the training set size was not increased in order to keep the training times reasonable.

The expected value of the test set error with both mismatch and additive noise was 0.157 or 0.198 km. The error can be reduced by increasing the training set size to include multiple identical environments with different noise. Increasing the training set size to 1500 different examples, 500 distinct environments perturbed by separate sets of noise three times, reduced the error to 0.090 or 0.150 km. Using this network the range estimate for the experimental data was 0.568 ± 0.150 km which agrees

with the value of 0.483 ± 0.017 km obtained with the grid search and also agrees with the GPS estimate of 0.500 km. Recall that the initial bounds used to estimate the distance from the array to the ship using the ANN are from 0.1 to 1.1 km. A narrower range of uncertainty with the same training set size would have a more densely sampled parameter space and thus a much lower error in the test set.

The results of this section show that the number of training points sampling the parameter space should be as large as possible for the best performance. The effect of noise on performance was reduced by applying different noise perturbations from the same distribution to the training data repeatedly. When this was done the error in the test set data was reduced considerably. The network essentially learns the data with mismatch and noise. The next step is to determine if the network can accurately estimate the mismatched parameters as well as the range.

5.1.3 Inversion for Group II–IV Parameters

The perturbed environments of the previous section were included in the target outputs for the network along with the range. Thirty networks were trained using cross-validation as the stopping criterion. Thirty hidden layer neurons were used and the noise was applied to the data set six times for a total of 3000 training examples. The best network based on the mse of the test set for all of the parameters was chosen. The results are shown in Table 5.2. The mean-squared error is the error of each scaled parameter, equivalent to the relative error measure used by Stephan et al. [12]. The second column contains the expected error in the parameter value. The third column contains the final parameter estimate using the experimental data. These results are somewhat deceiving since the network may not have trained for the parameters despite having a low mse. A common local minimum which is often seen is one in which the mean of the training targets is output regardless of the input. This can occur for one or more parameters. Some or all of the weights mapping the input to the output layer are near zero and the biases of the output layer have converged to the mean of the training targets. This result is shown in Figure 5.9. The topmost figure shows the target and actual output of a network whose training has stopped in a local minimum. The output of the network is the mean of the target outputs for all of the inputs. The bottommost graph shows what happens when training is continued for

Table 5.2. Estimation Results (BPNN): Range, Group II-IV

Parameter	mse (scaled)	Expected error	Estimate
source range, (sr)	0.171	0.206	0.576
p-wave speed (s), (cps)	0.409	79.9	1707
thickness of sediment, (t)	0.353	8.92	24.7
sspcp, (spc)	0.530	1.46	1.61
p-wave attenuation (s), (aps)	0.377	0.154	0.271
sspd, (spd)	0.466	1.37	2.69
s-wave speed (s), (css)	0.484	121.8	177.7
s-wave speed (b), (csb)	0.448	160.6	823.9
density (s), (ps)	0.516	0.143	1.88
p-wave speed (b), (cpb)	0.594	34.7	1974
density (b), (pb)	0.633	0.159	2.17
p-wave attenuation (b), (apb)	0.609	0.0780	0.105

that parameter; the output of the network begins to approach the targets at selected points where there is a large difference. This phenomenon occurs whether or not the targets are scaled, and for either sigmoid or linear output activation functions. The results can be evaluated based on whether or not the network estimate is superior to simply choosing the midpoint between the upper and lower bounds. The mean-squared error of a parameter scaled to lie between -1 and 1 whose estimate is always zero and whose targets are drawn from a uniform distribution is 0.333 ¹. If training is continued the weights and biases are slowly changed to try to reduce the error in the output for each of the parameters, without increasing the error for any of the others. The most efficient way for the backpropagation algorithm to reduce the output error is to change the weights so that the most crucial parameters are estimated first, since they cause the largest changes in the input data. From Table 5.2 we can see that the mse for the range is 0.171 which is less than $1/3$. The remaining parameters are greater than $1/3$ which signifies that more training is required for these parameters.

The estimate of the experimental range is slightly higher than that estimated

¹The variance of a variable (mean-squared) with a uniform distribution on $[-1,1]$ is $1/3$.

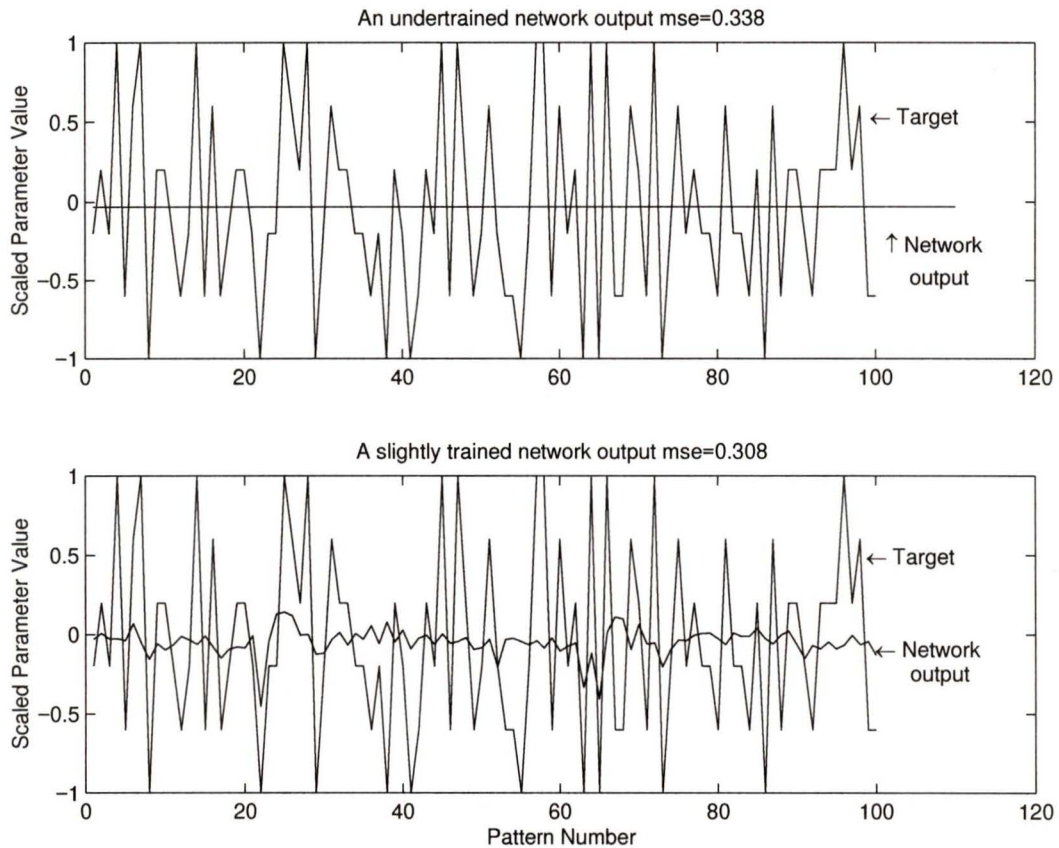


Figure 5.9. The output of an undertrained and slightly trained network overplotted with the target. The undertrained network has an output equal to the mean of the targets and a mse of 0.338. The slightly trained network has begun to approximate the targets and has a reduced error of 0.308.

Table 5.3. *Estimation Results (BPNN): Group II-IV*

Parameter	mse (scaled)	Expected error	Estimate
p-wave speed (s), (cps)	0.076	34.4	1844.3
thickness of sediment, (t)	0.102	4.80	29.7
sspcp, (spc)	0.097	0.621	3.89
p-wave attenuation (s), (aps)	0.279	0.131	0.060
sspd, (spd)	0.061	0.492	3.62
s-wave speed (s), (css)	0.278	92.2	225.9
s-wave speed (b), (csb)	0.285	128.2	568.7
density (s), (ps)	0.264	0.102	1.95
p-wave speed (b), (cpb)	0.345	26.4	2017
density (b), (pb)	0.434	0.131	2.14
p-wave attenuation (b), (apb)	0.436	0.0660	0.152

using the grid search, but the uncertainty of the estimate is high enough that the grid search result of 0.483 km remains feasible. The remaining parameters were not estimated any better than simply choosing the midpoint of the range of uncertainty and thus little confidence should be placed in those estimates.

If we fix the range at 0.483 km and generate a new data set, we expect that the network will train for the more important parameters such as compressional speed in the sediment and the sediment thickness. The range is fixed so that the largest variations in the training data are due to changes in the parameters which are less important than the range. The range of 0.483 km was chosen rather than the network estimate of Table 5.2 because it was found using a grid search which had an error of 0.017 km.

From Table 5.3 we see that the network trained with a known value for the range is capable of estimating all of the parameters except for the three least important ones (p-wave speed, density, and attenuation in the basement). The error for both the compressional speed and thickness of the sediment has been cut in half. Interestingly both of the water column sound speed profile parameters (spc and spd) have very low mse. This result indicates that they are more important to the calculation of the field than the simulation study indicated. The simulation study was performed with

all of the other parameters fixed at their nominal values, these nominal values may correspond to a point where the sound speed profile in the water is less important. The error in the other parameters is large but is significantly reduced from the errors of Table 5.2.

In both cases the network estimated the most important parameters and returned only the average values of the least important parameters. There may be several reasons for this. The most likely is that the noise completely dominated the relatively small effects caused by changes in the unimportant parameters (Group IV). The only way to accurately estimate the Group IV parameters would be to use a very clean data set along with a densely sampled training set.

Another way of obtaining rough estimates of the unimportant parameters is to exploit known correlations between parameters. For instance it is well known [1] that high sound speeds are often associated with high densities in realistic bottoms. Since it appears that a good estimate of the basement parameters is not achievable using neural inversion with this data, an alternative is to classify the sediment as belonging to a certain type rather than attempting to extract specific parameter values.

5.1.4 Bottom Classification

The goal of bottom classification is to determine the composition of the ocean bottom using acoustic data. A suitable classification breakdown is shown in Table 5.4, taken directly from the book by Jensen et al. [1]. Each bottom type is a separate class with its own distinct properties. The table shows that the density, compressional speed, and shear speed are all strongly correlated. The compressional and shear wave attenuations are somewhat negatively correlated with these parameters.

In the previous section we have shown that it is possible to estimate both the compressional speed and density of the sediment. The uncertainty in these parameters is high due to measurement noise and sparse sampling of the parameter space in the training data. The attenuations and shear wave speeds are difficult to estimate due to their minor effect on the acoustic field. Using the density and compressional speed estimates we can conclude that the sediment is likely a mixture of sand and gravel. This classification is consistent with *a priori* investigations [14]. In some experimental situations there may be no *a priori* information about the site, for those cases it would

Table 5.4. *Bottom Types and Nominal Parameter Values*

Class	Material	Dens.	P-wave Spd	S-wave Spd	Atten. (p)	Atten. (s)
1	Clay	1.5	1500	100	0.2	1.0
2	Silt	1.7	1575	160	1.0	1.5
3	Sand	1.9	1650	220	0.8	2.5
4	Gravel	2.0	1800	360	0.6	1.5
5	Moraine	2.1	1950	600	0.4	1.0
6	Chalk	2.2	2400	1000	0.2	0.5
7	Limestone	2.4	3000	1500	0.1	0.2
8	Basalt	2.7	5250	2500	0.1	0.2
UNITS		kg/m ³	m/s	m/s	dB/λ	dB/λ

be useful to quickly assess the bottom type using a classifier.

The estimator of the previous section could be used as a classifier by simply observing the compressional speed and comparing it with that in Table 5.4. However, the network was only trained to estimate compressional speeds in the range 1550 to 1850 and would not be useful if the bottom type was basalt, for example. A network could be designed which encompassed all of the possible environments of Table 5.4, but the number of training examples that would be required to sample the function space sufficiently would be prohibitively large. A classifier using a small training set and a small network can be quickly trained to accurately classify bottom type.

The classifier training set is generated by choosing sample environments from within each class. In this case the class boundaries will be arbitrarily defined as the midpoint between the nominal environment parameter values for each class. The acoustic field is simulated using ORCA for each of these training cases. A similar, but distinct, data set is generated to use for testing and cross-validation. The target outputs may be defined in a number of ways. One of the simplest and most robust methods is to use the ‘one-hot’ approach where there is a single output for each class. The largest output determines the class to which the data belongs. This is called the ‘winner-takes-all’ rule. The output of the neuron for class i is set to unity if the data belongs to class i , otherwise the output is zero. In order to train the network the target outputs are simply ones and zeros and the outputs of the network are allowed

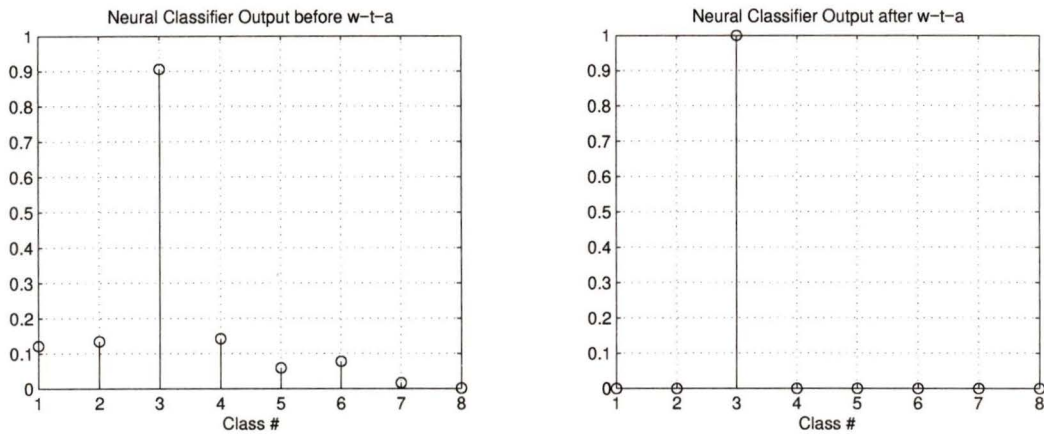


Figure 5.10. The output of the neural classifier before and after the application of the winner-takes-all function.

to take on *any* value between zero and one. The network is trained as before to minimize the mean-squared error between the target and the output. The winner-takes-all function is only applied after training to determine the misclassification rate of the test set. Figure 5.10 shows the output of the classifier before the application of the winner-takes-all rule and afterwards. The maximum of the function determines the classification.

The pre-processing is identical to that for the continuous estimation case. Several heuristic methods can be used to speed convergence. The neural network classifies by constructing *decision boundaries* [30], if the input vector falls inside the decision boundary for class i then it is said to belong to that class. In a well-conditioned data set the data for each class is *clustered* within its boundaries. In a poorly constructed problem there may be many outliers and data points which are near to the decision boundaries. This is avoided by pre-processing the data to extract *features* representative of each class. Principal components analysis extracts the features of the data which have the greatest variance within the training set; features which are common to all of the classes are excluded.

The advantage of the neural network method is that very complex nonlinear decision boundaries can be constructed [27]. One good way of initializing the network is to first train on a subset of the data. Once the training set is constructed a good representation of each class is its mean feature vector. This feature vector can be

used to generate the initial decision boundaries for the network. Once the network has trained to have zero error for these *prototype* feature vectors the entire training set is used to continue the training. This method effectively avoids local minima and greatly reduces the total training time. In order for the network to classify noisy data as well, a noisy training set should be used. This set can be introduced after the network has trained for the noise-free data. If the network fails to train for the noisy data more features should be included in the feature vector to promote greater class separation. Increasing the size of the network increases the complexity of the decision boundaries but cannot overcome non-uniqueness or significant overlap of the feature vectors. The decision boundary construction process is illustrated in Figure 5.11. The figure shows that the initial training using the mean feature vector for each class arrives at a decent solution but that solution may misclassify some of the vectors in the complete training set. When the complete training set is included, a classifier with zero error can be designed. The introduction of noise requires a further modification of the decision boundaries. This step-wise approach to training is fast and results in efficient and accurate networks.

A network was trained, using the step-wise approach, to classify sediment types. Five hidden neurons and 15 principal components were used. The network was trained in each case until the mean-squared error was less than 0.001; this value invariably corresponded to zero error in training set classification. The training set consisted of 100 samples per class, and within each class the environmental parameters were sampled uniformly. The feature vectors were formed by normalizing, standardizing (subtracting the mean and dividing out the standard deviation) and using principal components analysis on the data vectors. The test set was formed in an identical fashion. The probability of error in the noisy test set was 36.5% after training with the class prototypes. The error was reduced to 13.3% after training with the noise-free feature vectors. The final error after training with 3 sets of noisy data was 4.3%. Increasing the number of hidden neurons to 15 and the number of features (inputs) to 25 lowered the probability of misclassification to 3.9%. The output of the network for the experimental data is shown in Figure 5.12. The network has high outputs for both sand and gravel. The gravel output is slightly higher, thus it is the final choice for the class. A small perturbation in the input could easily result in a sand classification

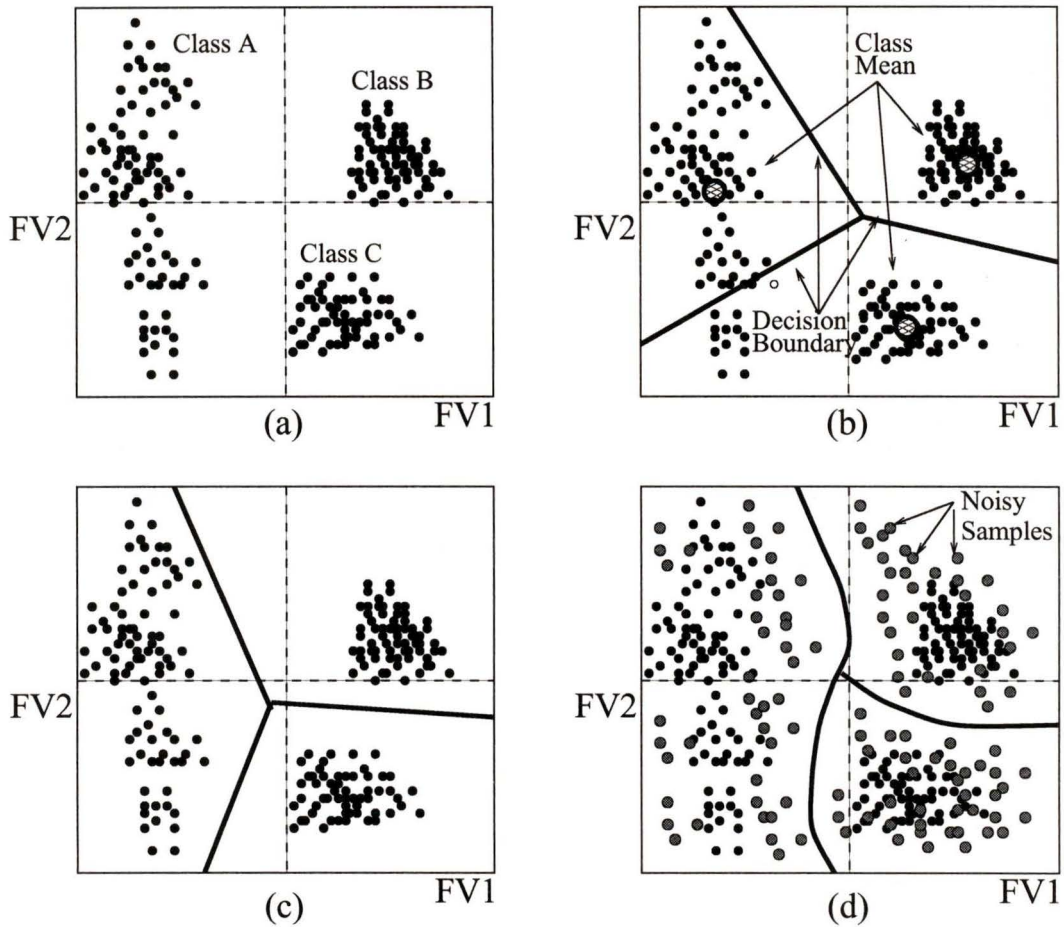


Figure 5.11. *The classification process with a two-element feature vector and three classes. (a) Initial feature vectors and classification. (b) Decision boundaries after training on the prototype feature vectors. (c) Decision boundaries after training on the complete data set. (d) Decision boundaries after training on the noisy data.*

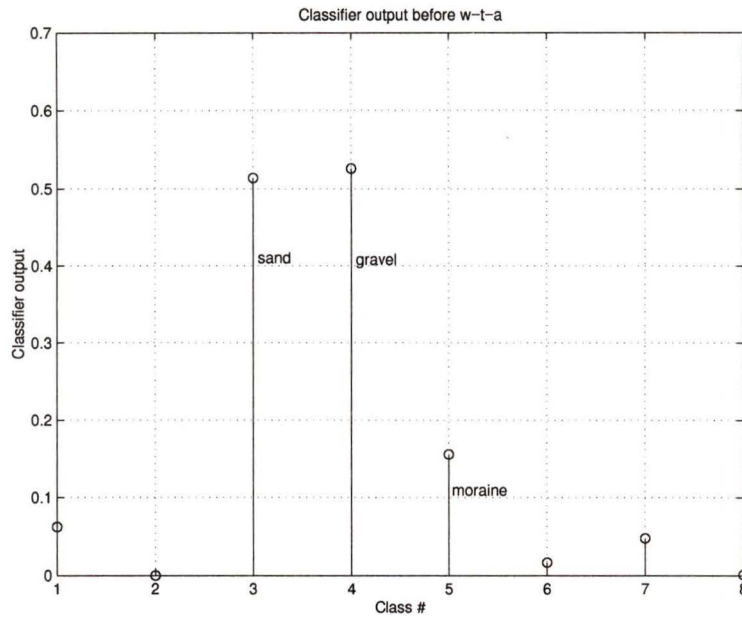


Figure 5.12. The output of the backpropagation neural classifier for the experimental data show that the bottom is likely a mixture of sand and gravel. The final classification is gravel.

since the data point is near to the boundary between the sand and gravel classes.

The feature vector size was increased to include all 80 principal components but the probability of error was not reduced. A network was also trained using the raw data without PCA and the error in the test set was increased to 86%. The final error can only be reduced by a denser sampling of the classes or possibly through the use of a different set of features to eliminate the possibility of non-uniqueness and further increase class separation (and thus the resistance to noise and mismatch).

5.2 Radial Basis Function Results

In the previous section ORCA was used to generate the training and test sets for the BPNN. In this section identical training sets and pre-processing were used to design the radial basis function networks of this section. The main difference is that each network is designed deterministically using the training set. In an effort to keep the networks small, only the noise-free data sets were used as the centres. The spreads

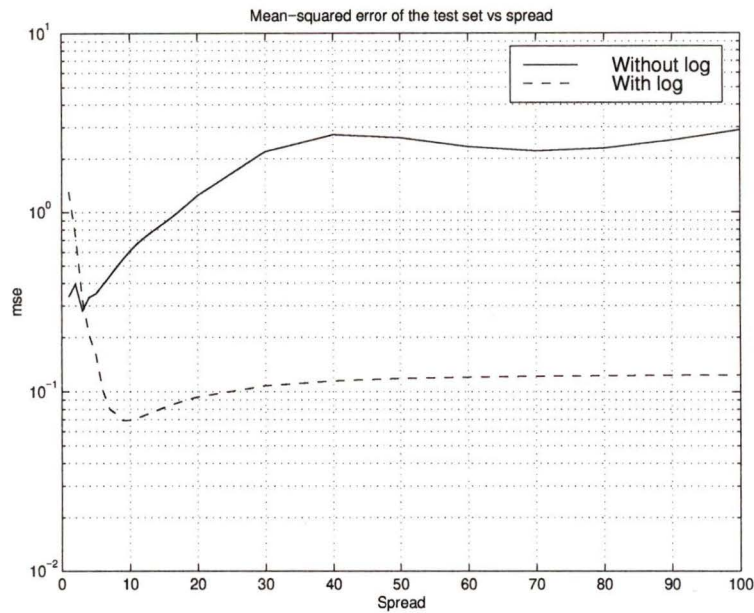


Figure 5.13. *The mean-squared error of the test set plotted vs. the spread with and without smoothing of the input with a logarithm. The compressed data achieves a lower error in the test set.*

were chosen based on a grid search. Several networks were designed, each with a different spread. The network with the lowest training set error was chosen. When the noise is included in the training set the position of the centres is perturbed and the final estimation suffers. Adding noise to the network several times, as was done with the BPNN, results in an ill-conditioned network, because there are several centres that are close together and correspond to the same output.

5.2.1 Range Finding

The range finding network was designed with 200 noise-free training examples, each at a different range. The mse of the test is plotted vs. the spread in Figure 5.13. Two separate cases are shown. In one, the data are not pre-processed using the logarithm, in the other the logarithm is used. The logarithm serves to compress the dynamic range of the input data and reduce the effect of outliers. The error here can be reduced further only through denser sampling. The number of principal components used was the same as with the backpropagation network (40). The optimal spread is

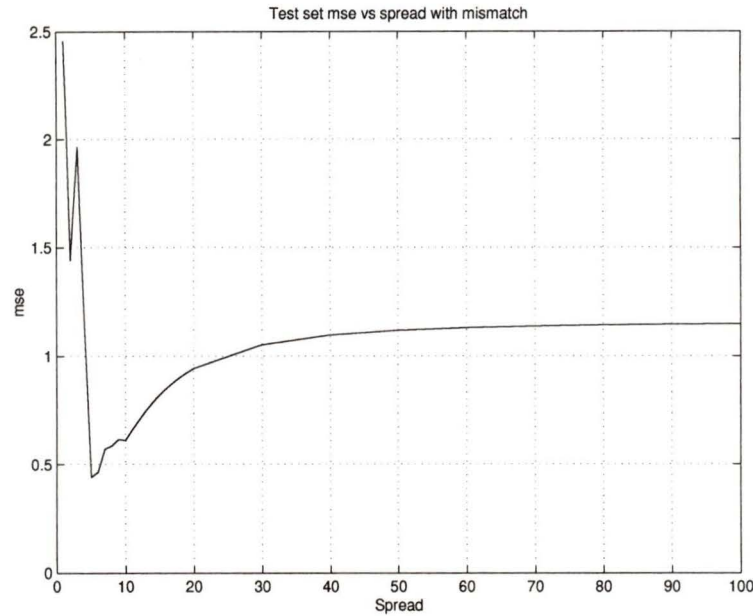


Figure 5.14. *The test set mse is minimized for a spread of 5. The nonlinear changes in the spread indicate that the parameter space is poorly sampled with only 250 centres.*

nine and the mse for this spread is 0.069. The estimation error was 131 m. The best accuracy achieved for the backpropagation network with the same noise level in the test set was 116 m.

5.2.2 Range Finding with Mismatch

In the presence of mismatch in the training and test sets the number of centres (training points) should be increased. The number of training examples used for the backpropagation network was 500, but that method requires significantly more computation time. Therefore a network with only 250 centres was tried. Figure 5.14 shows that the optimal spread in this case is approximately 5. The error has significantly increased to 0.44 (mse) or 0.332 km. The high error and the sharp changes in the mse with spread indicate that the parameter space is not sampled properly. Ideally the input vectors should be evenly placed throughout the function space. The training set was generated by uniformly sampling from the parameter space, not the feature vector space. One improvement would be to use an iterative algorithm which varies both the centres and the spreads. This type of algorithm is available [27] but

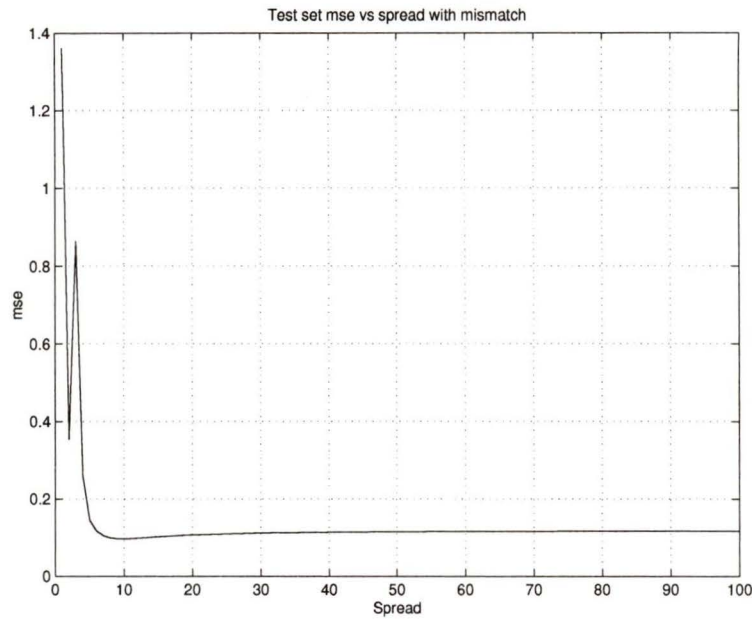


Figure 5.15. *The test set mse is minimized for a spread of 10. The smoothing of the mse curve indicates that the parameter space is well sampled with 500 centres.*

the advantage of quick design would be lost.

With 500 centres the spread value stabilizes somewhat and the error is reduced. The test set mse plotted against the spread is shown in Figure 5.15. The optimal spread value is 10 and the mse of the test set is significantly improved to 0.097 or 155 m. This error is roughly the same accuracy as that seen with the BPNN, however, the design time is still reduced by an order of magnitude, even with the inversion of a 500×500 matrix and the required use of virtual memory. The range estimate for the experimental data is 0.522 ± 0.155 km which is in agreement with the grid search estimate of 0.483 ± 0.017 km and the GPS estimate of 0.500 km.

5.2.3 Inversion for Group II–IV Parameters

The estimates for the mismatched parameters including range are shown in Table 5.5. The spread was determined to be 6.0 using a grid search. The errors are similar to those of the BPNN, although the error in the range is less than that of the BPNN. As with the BPNN the other parameters were not estimated well. In this case however the parameters which are not estimated well do not generally show the tendency to

Table 5.5. *Estimation Results (RBFNN): Range, Group II-IV*

Parameter	mse (scaled)	Expected error	Estimate
source range, (sr)	0.108	0.164	0.507
p-wave speed (s), (cps)	0.365	75.5	1732
thickness of sediment, (t)	0.478	10.3	22.3
sspcp, (spc)	0.425	1.30	2.50
p-wave attenuation (s), (aps)	0.537	0.183	0.141
sspd, (spd)	0.606	1.55	2.69
s-wave speed (s), (css)	0.613	137.0	196.3
s-wave speed (b), (csb)	0.787	212.9	818.7
density (s), (ps)	0.486	0.139	1.89
p-wave speed (b), (cpb)	0.620	35.4	1983
density (b), (pb)	0.935	0.193	2.25
p-wave attenuation (b), (apb)	0.672	0.0820	0.130

estimate closer to the mean. Some parameters may appear overtrained and have a very large mse if the spread value is too small relative to the optimal spread for that parameter.

Table 5.6 contains the results of the estimation of the Group II-IV parameters when the range is fixed at 483 m. The results are similar to those obtained using the BPNN. The most important parameters such as compressional speed in the sediment are estimated well. The least important parameters such as basement density are estimated poorly.

5.2.4 Bottom Classification

Classification with the RBFNN is slightly different than with the BPNN because of the way that the networks are trained. In this instance we will choose our centres to be the prototype feature vectors. The decision boundaries for RBFNNs are based on Gaussian functions. Each class has a multi-dimensional Gaussian centred at its mean. The input vector is compared (using the Euclidean distance) to the mean of each class. The lowest distance determines which class the input belongs to. When

Table 5.6. Estimation Results (RBFNN): Group II–IV

Parameter	mse (scaled)	Expected error	Estimate
p-wave speed (s), (cps)	0.065	31.9	1847.7
thickness of sediment, (t)	0.199	6.68	31.3
sspcp, (spc)	0.120	0.694	2.68
p-wave attenuation (s), (aps)	0.285	0.133	0.32
sspd, (spd)	0.230	0.960	3.86
s-wave speed (s), (css)	0.332	100.9	143.9
s-wave speed (b), (csb)	0.352	142.5	861.9
density (s), (ps)	0.486	0.139	1.65
p-wave speed (b), (cpb)	0.459	30.5	2025
density (b), (pb)	0.704	0.167	1.98
p-wave attenuation (b), (apb)	0.439	0.0663	0.0613

the distance from the data to all of the clusters is large then the classification is ambiguous. Since the data set has outliers, the radial basis function approach is not the best way to create the decision boundaries. Using these 8 centres the error rate was 18.8 % for an optimized spread of 2.75. This is not an unacceptable result. Because RBFs are *local* function approximators [27], the method has difficulty dealing with outliers in the data. The BPNN can deal with outliers because it is a *global* function approximator. If we look at a scatter plot of the first two feature vectors for two of the classes we can see why choosing a single centre and a Gaussian function is not appropriate. Figure 5.16 shows the first two feature vectors for the first two classes. The small points correspond to the clay class, and the larger points to the silt class. There is little overlap between the two clusters but they are not well separated. The problem that the RBFNN has in classifying them is that the clusters are not normally distributed; thus choosing the decision boundaries based on the mean of the cluster and a Gaussian is not efficient in this case. This approach, however, would be appropriate if we knew that the distribution within each cluster was Gaussian. Unfortunately, the distribution within each cluster is determined by the forward model ORCA operating on a uniformly sampled environmental parameter space.

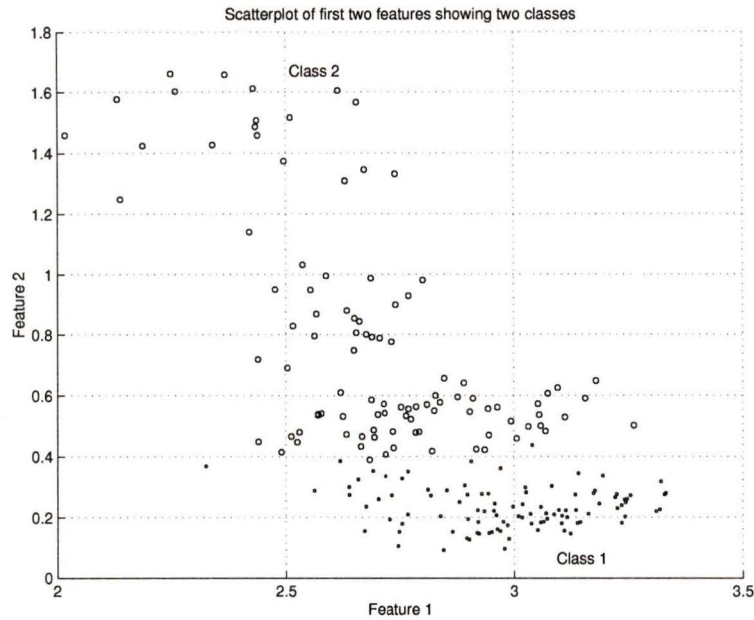


Figure 5.16. Scatterplot of the first two feature vectors for the first two classes. The small points correspond to the clay class and the larger points are for the silt class. The distribution of points within each class are not Gaussian.

The output of the RBFNN classifier for the experimental data is shown in Figure 5.17. The classification in this case is much more ambiguous than with the BPNN. The final classification is for sand. Despite the large probability of misclassification the estimate agrees with the *a priori* information which asserts that the sediment is a mixture of sand and gravel.

5.3 Discussion of Results

In the range-finding case both the RBFNN and the BPNN performed well, estimating the range to within 0.100 km for a range of uncertainty of 1.00 km. This is expected since the range is a very important parameter relative to the bottom parameters in the calculation of the field. The networks which were designed to estimate the range as well as the Group II-IV parameters have a degraded range estimation, and the Group II-IV parameters were not estimated at all (the errors for the parameters were all greater than 1/3).

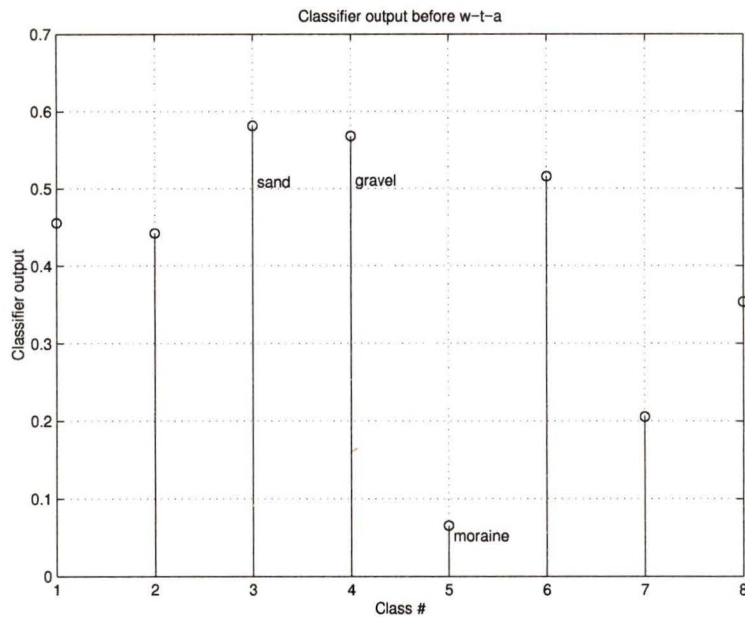


Figure 5.17. *The output of the neural RBF classifier for the experimental data is more ambiguous than with the BPNN. The probability of misclassification is 18.8%. The final classification is sand.*

The BPNN essentially trains to reduce the error in the simplest possible way. Since changes in the range cause such large changes in the field it was relatively easy for the network to learn this behavior and thus reduce the mse in the range. The other parameters initially converge to values close to their mean. When training was continued the network began to memorize the range and thus the mse increased. Since training was stopped at this point the remaining parameters were not estimated well because of their low importance. Generally these parameters have estimates which are nearer to their mean value.

The RBFNN performance was very sensitive to the spread value, which was selected based on a grid search. The range caused significant changes in the data vector which were not as easily masked by the noise and thus the optimal spread resulted in a network capable of estimating the single most important parameter and the network demonstrated poor generalization for the remaining outputs.

When the range was fixed at its known value, the networks showed an improved performance for the less important parameters, in particular the compressional speed

in the water column and the sediment were estimated with a low expected error. Again in this case the most important parameters were estimated more accurately than those parameters which had little effect on the calculation of the field and the basement parameters remained inestimable.

In the BPNN case the depth of the sound speed profile in the water column was estimated with the least relative error. The next best approximations were for the compressional speed in the sediment, the parameter controlling the sound speed profile in the water column, and the thickness of the sediment. The remaining Group III parameter estimates were improved slightly over the case where range was included and offered a slight improvement over the initial range of uncertainty for those parameters. The Group IV were not identified as indicated by their high mse for the test set. Since the Group IV parameters had almost no effect on the field calculation this was not a serious setback.

The RBFNN estimates showed the main drawback of the spread selection method. Again we saw that only the most important parameter was estimated (in this case the compressional speed of the sediment). The other parameters which showed an acceptable mse were the sound speed profile in the water column, the thickness of the sediment, and the attenuation of the sediment. The remaining six parameters have mean-squared errors which are too large to place any confidence in the network estimates.

The classification of the data with respect to sediment type was satisfactory. Both networks accurately classified the bottom as consisting of either gravel or sand. The probability of error based on the test set for the the BPNN was less than 4%. The error probability for the RBFNN was significantly higher at 20%. Using the continuous estimates as a classifier, it is apparent that compressional speed in the sediment is the main indicator of the class, and that both the BPNN and RBFNN would have classified the data correctly. The comparison between the continuous estimator and the classifier is not fair since the continuous compressional speed was constrained to lie within the bounds of 1500 and 1850 m/s. The classifier had to deal with much wider bounds of up to 5000 m/s.

5.3.1 Application of New Environments to Source Localization

Figure 5.18 shows the ambiguity surface for all nine frequencies including the incoherent average for the environment predicted by the BPNN. Figure 5.19 shows the same surfaces for the environment predicted by the RBFNN.

When the neural estimates of the parameters are used to localize the source using MFP several problems with this technique became apparent. The peak values for both estimated environments improved. The Bartlett incoherent average of all eight frequencies showed a peak value of 0.435 for the BPNN output versus 0.44 for the parameters before the inversion. The peak location, however, has shifted into the feasible region. The sidelobes for the 55 Hz peak have increased to the point where the global maximum is no longer near the correct range and depth. There appears to have been no change in the 35 Hz surface. The results from the RBFNN show that the source is correctly localized with the 35 and 55 Hz surfaces as well as with the incoherent average surface. The Bartlett power of the global maximum is improved to 0.45 from the nominal value of 0.44.

Both of the neural networks found solutions which were different but allowed localization of the source. The increase in the Bartlett power, however, is minimal; there are two main reasons for this. The parameters which we seek to find have little influence on the field and the error bounds after the inversion are still quite wide. It appears that the averaged surface is most heavily influenced by the low frequencies since they have the largest Bartlett powers. The low frequencies also have the lowest resolution (in terms of peak width) and the highest sidelobes. The low frequency fields are less sensitive to mismatch in geometric parameters, thus if the grid search results were invalid then the effect of mismatch would be more pronounced with the higher frequencies. The other reason why the low frequencies are seen to dominate the inversion is that the mapping between the data and the parameters may be smoother and easier to approximate. In fact, with this technique, it may be that the lower the resolution the better the performance will be.

There may also be a problem with spatial aliasing. In plane-wave beamforming the maximum spacing between the hydrophones is restricted to $1/2$ the wavelength. If the spacing is wider, then the Nyquist criterion in space is not met and the beamforming

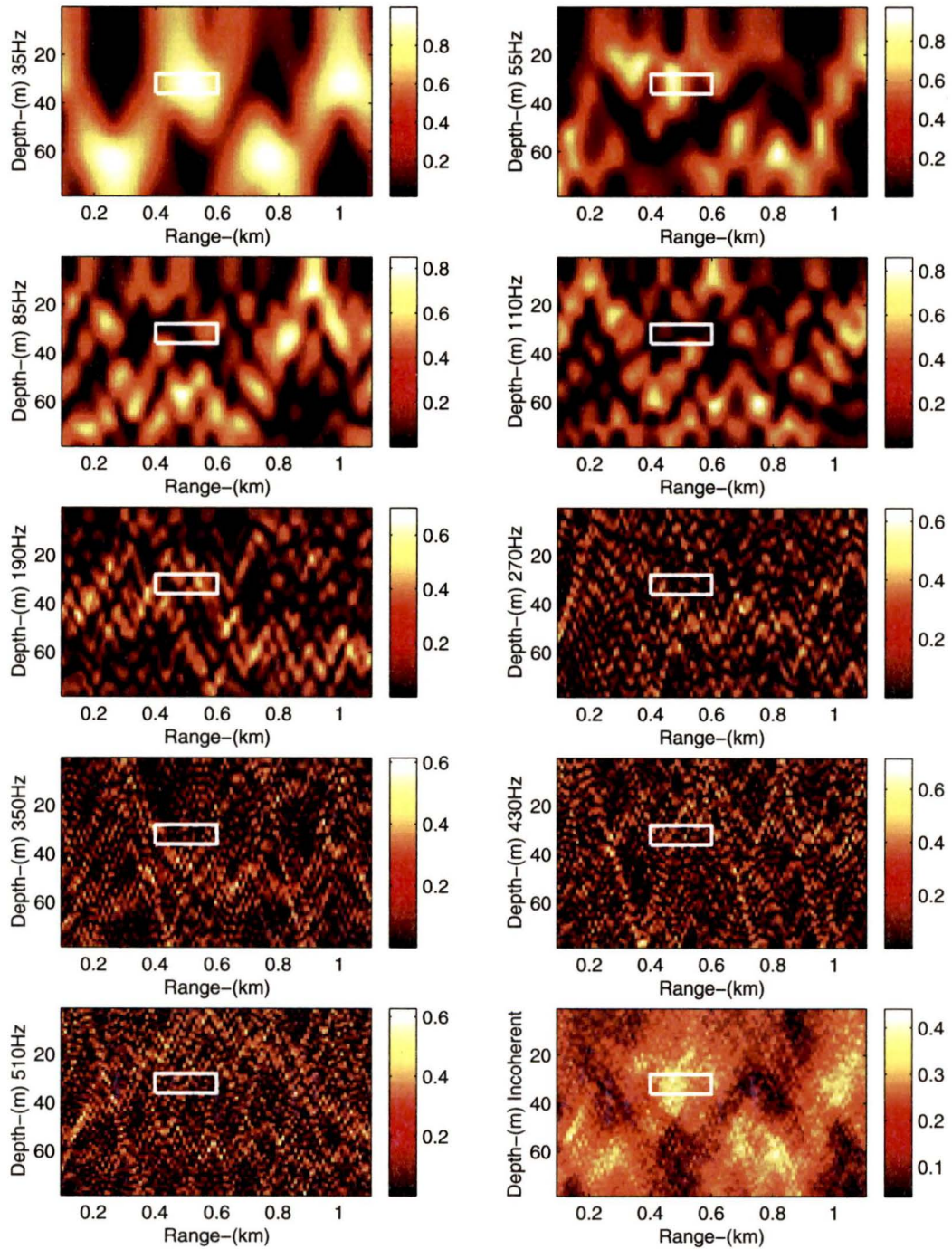


Figure 5.18. The ambiguity surfaces for the environment predicted by the BPNN.

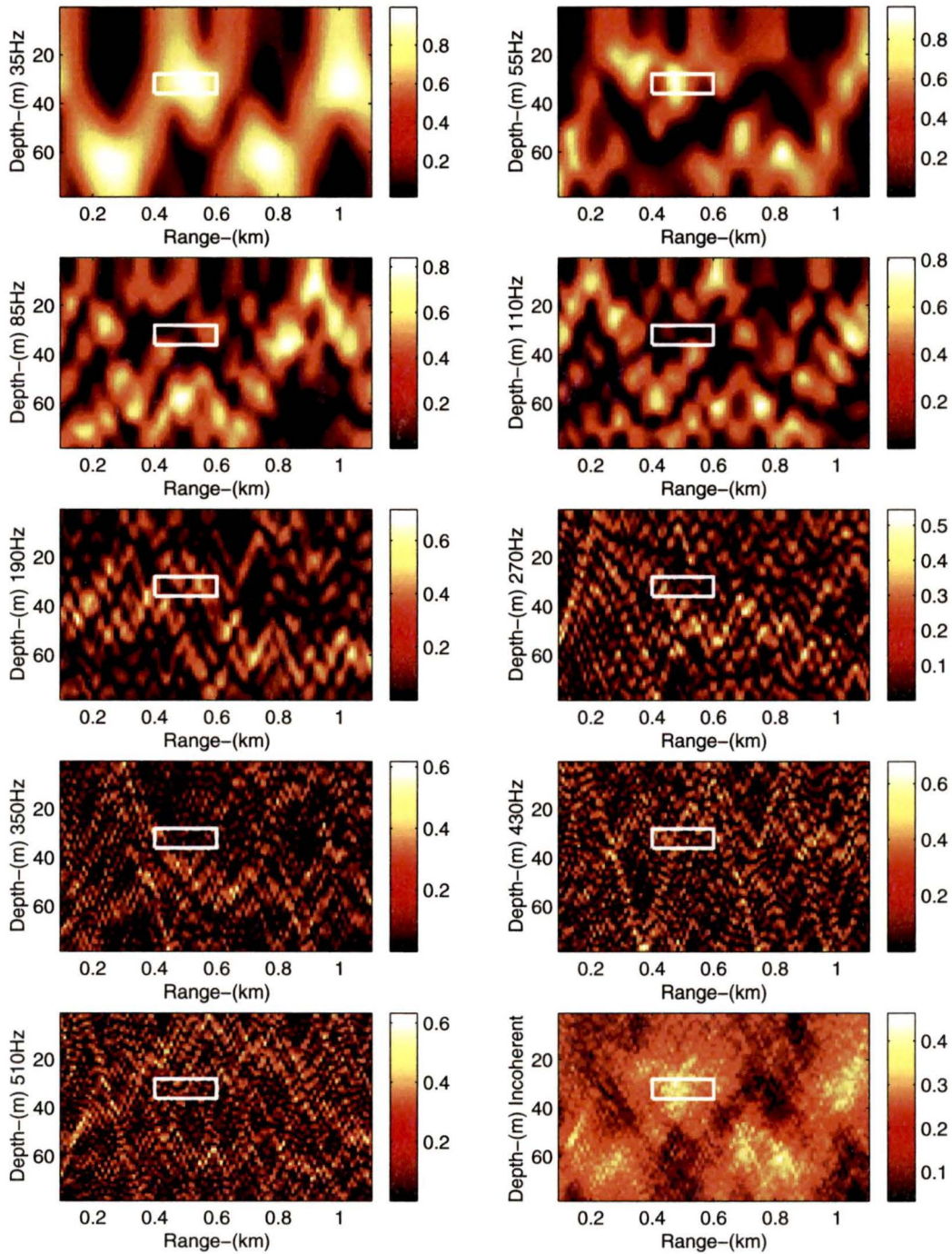


Figure 5.19. The ambiguity surfaces for the environment predicted by the RBFNN.

operation generally fails. It is not clear whether a similar condition exists for matched-field processing [4].

5.3.2 Sources of Error

There is one main source of error in the neural inversion method. The data set used to test the neural network accounts for both mismatch in the parameters and the additive noise, but it does not include the possibility that the assumed bottom structure is incorrect. For example the range-independence assumption could be incorrect and including range-dependence in the test data could result in a more accurate mse which is higher than the range-independent data would produce. Thus the estimate of the error is always slightly optimistic. The actual network error could also be higher if the bottom structure is more detailed (for example several overlying sediment layers rather than just one). The networks are only capable of estimating the best *effective* parameters in this case.

5.4 Conclusions

In this chapter the results of applying the neural inversion approach to experimental data were presented and discussed. Both BPNNs and RBFNNs were designed to estimate the parameters of the TRIAL SABLE experiment and to classify the bottom type based on the data. The networks were then tested using an independent set of simulated data in order to estimate the network accuracy for each parameter. The parameter estimates from the inversion of the experimental data were then used for source localization with matched-field processing.

Chapter 6

Conclusions and Recommendations

In this chapter a thesis summary is given along with some conclusions about the application of neural inversion to experimental data. Possibilities for future work in this area are then outlined.

6.1 Summary and Conclusions

The objective of this thesis was to evaluate the use of neural networks in estimating parameters from experimental data. Three main issues were investigated: the suitability of neural networks for the model inversion problem, the features of an experiment required for neural inversion to succeed, and the proper construction of a test data set to evaluate the network performance.

In Chapter 1 the research work has been outlined along with the basics of ocean acoustic parameter estimation. Chapter 2 served as an introduction to the topic of ocean acoustic modeling and model-based signal processing. The acoustic wave equation was derived and the ORCA normal mode solution was detailed. Matched-field processing and matched-field inversion were introduced along with practical processors and suitable normalization. The effect of mismatch in the model parameters on source localization was examined with a simulated example, which showed that even a small relative change in the model parameters can cause the localization to fail.

In Chapter 3 neural networks for inverse system identification were introduced. The backpropagation and radial basis function networks were described. The training algorithms used to design these networks were presented along with several modifications for large problems which have many local minima in the error function.

Chapter 4 summarizes the TRIAL SABLE experiment in detail and should be a

useful reference for future users of this data set. A complete description of the *a priori* information available at the site was given, followed by a simplified environment parameterization which is suitable for use with ORCA. A new measure of parameter importance in inversion methods was developed and calculated for the parameterized environment. The most important parameters turned out to be the geometric parameters of the experiment including the positions of the source and receivers, and the water depth. The next most important parameters were those of the compressional sound speed profile in the water and the sediment. The parameters of the basement layer were found to be relatively unimportant for the calculation of the field. It is believed that this is because the sediment layer is highly reflective and thus very little of the signal penetrates to the basement. Source localization based on the nominal environment parameters was unsuccessful. A coarse grid search for the geometric parameters was carried out and superior estimates were obtained. Source localization with the new environment was successful, but the ambiguity remained high. Neural inversion was proposed as a method to obtain better estimates of the less important parameters and thus reduce the ambiguity.

In Chapter 5 the inversion of experimental data was done with both BPNNs and RBFNNs. The BPNN is capable of estimating a number of the more important parameters fairly well whereas the RBFNN estimates only the single most important parameter with any accuracy. The reasons for this lie in the methods in which the networks are designed and are described in Chapter 5. The parameters of low importance are not estimated well by either network because they simply do not have enough of an effect on the acoustic field to be detected by the network. In order to obtain an indication about the actual physical properties of the sediment layer a neural classifier was designed. The BPNN classifier effectively determined the sediment type to be gravel, with only a 4% probability of misclassification. This result is consistent with *a priori* investigations which determined that the bottom is a mixture of sand and gravel. The RBFNN classifier had a misclassification rate of almost 20%, and classified the data to be sand. The low error rate for the BPNN classifier indicates that it is a superior choice for future classification problems of this type.

When the parameters obtained from the neural networks were used for source localization the peak values remained roughly the same, but their positions have

shifted. This may indicate that there are some correlations between the estimated parameters and the source location. The fact that both networks have found parameters which enable the successful localization of the source is promising, but it is not possible to accurately quantify the success of the inversion without knowledge of the true environmental parameters.

The neural inversion of experimental geoacoustic data can deliver accurate estimates of some environmental parameters. The main advantage of the technique is that the expected estimation accuracy is easily determined through the inversion of a large test data set. Similar estimation accuracies are not available with methods such as simulated annealing or genetic algorithms since these methods are usually evaluated based on their efficacy at estimating only one set of simulated parameters, thus the accuracy estimates are for the specific experiment and not the general case. Previous studies of neural inversion have shown that fairly good accuracy can be attained for simulated data. This thesis shows that under less than ideal experimental conditions, the estimation of important parameters is still possible, though it may be less effective than the commonly used global search methods, since the parameters obtained with the neural inversion have not greatly improved the Bartlett power. The other advantage of the ANN-based inversion is that the number of forward models which need to be calculated is an order of magnitude less than that required by global search routines; however, the global search routines may have superior accuracy.

Backpropagation neural networks serve as very effective data classifiers and are capable of determining the sediment type unambiguously under noisy conditions. They are particularly simple to implement when the data classes are known *a priori* as in this case. More work needs to be done on the use of radial basis function networks for classification.

6.2 Suggestions for Further Research

Future work in this area should involve three separate considerations: experiment design, forward modeling, and inversion technique improvements. In order to use experimental data more accurate information needs to be gathered about the geometry of the experiment. The position of the source relative to the position of each of

the hydrophones is very important in the calculation of the field, an effective way to determine this during the experiment needs to be developed. The water depth and the sound speed profile in the water column should be measured repeatedly as the experiment progresses so that changes may be taken into account. More hydrophones could be used to increase the spatial sampling rate and thus reduce the instances of spatial aliasing at high frequencies. The time sampling rate should also be increased so that the high frequency components of the signal can be ascertained with lower variance. This could also be achieved by reducing the ship speed or using a stationary source. Background noise levels should be recorded periodically during the run in order to accurately estimate the SNR; ideally the experiment should be conducted in an area where the shipping noise is lower or absent. A normalization scheme which is not dependent on the first sensor needs to be found in order to avoid possible ill-conditioning problems that arise when the first hydrophone is located in a mode null.

Despite the small changes in water depth at the site, the inversion accuracy could be improved by incorporating a range-dependent model. The adiabatic normal mode models are well suited for this experiment since the slope of the bottom is much less than 3% which is the upper limit for this approach. A greater number of layers, or allowing gradients of properties within each layer, could be added. These additions would increase the computation time but could conceivably result in more accurate field calculations.

The neural inversion techniques could be improved in many ways. The backpropagation algorithm could be replaced by a more efficient search technique such as a quasi-Newton or conjugate direction method with a line search. This can be done if a computer with large amounts of memory (more than 256 MB of RAM) is used. The cross-validation stopping method should be maintained since it is quite efficient. The radial basis function network could be improved through the use of an algorithm that uses variable spreads and centres, even though these algorithms do not have the advantage of fast training time, which is seen with the exact RBFNN. The pre-processing could be improved by selecting only those frequencies that are important for the parameters of interest. The time required to find the optimal network size, the number of principal components, and other network training parameters could be

reduced by employing a genetic algorithm, as opposed to the exhaustive search used here. For instance low frequencies tend to penetrate quite deeply into the ocean bottom and are thus useful for determining basement parameters. The lower frequency fields are also less sensitive to small uncertainties in the experimental geometry. The higher frequencies allow for higher resolution and can be used to determine parameters such as source location, water depth, and sediment thickness after the bulk parameters have been found.

The observation that neural networks tend to estimate the most important parameters and have trouble with the less important parameters suggests the following successive approximation technique. A network is designed using a training set and a test set with variations in all of the unknown parameters, but only the most important parameter is used as an output. Once the error in the test set is sufficiently low, the most important parameter is estimated using the experimental data and the network. *New* training and test sets are then generated with the estimated parameter fixed, and a second network is trained to estimate only the second most important parameter. This parameter is then estimated from the experimental data using the second network. A third training set is generated with the first two estimated parameters fixed at their new values and the remaining parameters floating. A network is trained and the third parameter is estimated. This continues until all of the parameters are found. An obvious disadvantage of this method is the increase in the inversion time.

An alternative technique would be to weight the cost function used to optimize the network weights so that parameters which are less important are more heavily weighted than those which are not. Using the importance measure described in Chapter 4 would be a suitable basis on which to determine the weights.

This data set should be inverted with a global search scheme such as genetic algorithms optimizing a mismatch function. The results could then be compared with the neural inversion technique to assess whether the final estimation errors are due to the inversion method or to the limits of the experiment itself.

Bibliography

- [1] F.B. Jensen, W.A. Kuperman, M.B. Porter, and H. Schmidt, *Computational Ocean Acoustics*, AIP Press, Woodbury, New York, 1994.
- [2] R. J. Urick, *Principles of Underwater Sound*, McGraw-Hill, New York, 1983.
- [3] N. R. Chapman and K. S. Ozard, "Matched field inversion for geoacoustic properties of young oceanic crust," in *Full Field Inversion Methods in Ocean and Seisom-Acoustics*, O. Diachok, A. Caiti, P. Gerstoft, and H. Schmidt, Ed. Kluwer Academic Publishers, New York, 1995.
- [4] A. Tolstoy, *Matched Field Processing for Underwater Acoustics*, World Scientific, Singapore, 1993.
- [5] M. Sen and P. L. Stoffa, *Global Optimization Methods in Geophysical Inversion*, Elsevier Science B. V., New York, 1995.
- [6] M. J. Hinich, "Maximum likelihood estimation of the position of a radiating source in a waveguide," *Journal of the Acoustical Society of America*, vol. 66, pp. 480–483, 1979.
- [7] H. P. Bucker, "Use of calculated sound fields and matched-field detection to locate sound sources in shallow water," *Journal of the Acoustical Society of America*, vol. 59, pp. 368–374, 1976.
- [8] A. B. Baggeroer, W.A. Kuperman, and H. Schmidt, "Matched field processing: Source localization in correlated noise as an optimum parameter estimation problem," *Journal of the Acoustical Society of America*, vol. 83, pp. 571–587, 1988.
- [9] M. D. Collins and W.A. Kuperman, "Focalization: Environmental focusing and source localization," *Journal of the Acoustical Society of America*, vol. 90, pp. 1410–1421, 1991.
- [10] A. Tolstoy, N. R. Chapman, and G. Brooke, "Workshop '97: Benchmarking for geoacoustic inversion in shallow water," *Journal of Computational Acoustics*, 1998, in press.
- [11] D. F. Gingras and P. Gerstoft, "Global inversion of acoustic field data in shallow water using genetic algorithms," in *Full Field Inversion Methods in Ocean and Seisom-Acoustics*,

- O. Diachok, A. Caiti, P. Gerstoft, and H. Schmidt, Ed. Kluwer Academic Publishers, New York, 1995.
- [12] Y. Stephan, Y. Demoulin, and O. Sarzeaud, "Neural direct approaches to geoacoustic parameter estimation," *Journal of Computational Acoustics*, 1998, in press.
- [13] S. M. Jesus and A. Caiti, "Estimating geoacoustic properties from towed array data," *Journal of Computational Acoustics*, vol. 4, pp. 273–290, 1996.
- [14] J. Osler, "A geo-acoustic and oceanographic description of several shallow water experimental sites on the Scotian shelf," technical memorandum 94/216, DREA, 1994.
- [15] E. K. Westwood and P. J. Vidmar, "Eigenray finding and time-series simulation in a layered-bottom ocean," *Journal of the Acoustical Society of America*, vol. 81, pp. 912–924, 1987.
- [16] M. Porter, "Ocean acoustics library," <http://oalib.njit.edu>, 1998.
- [17] E. K. Westwood, *ORCA User's Manual*, 1997, Available via FTP from <ftp.arlut.utexas.edu>.
- [18] D. G. Zill and M. R. Cullen, *Advanced Engineering Mathematics*, PWS-Kent, Boston, 1992.
- [19] E. K. Westwood, C. T. Tindle, and N. R. Chapman, "A normal mode model for acousto-elastic ocean environments," *Journal of the Acoustical Society of America*, vol. 100, pp. 3631–3645, 1996.
- [20] S. M. Kay, *Modern Spectral Estimation*, Prentice Hall, New Jersey, 1988.
- [21] E. K. Westwood, "Broadband matched-field source localization," *Journal of the Acoustical Society of America*, vol. 91, pp. 2777–2789, 1992.
- [22] Z. Michalopoulou and M. B. Porter, "Matched-field processing for broad-band source-localization," *IEEE Journal of Oceanic Engineering*, vol. 21, pp. 384–392, 1996.
- [23] Z. Michalopoulou, "Robust multi-tonal matched-field inversion: a coherent approach," *Journal of the Acoustical Society of America*, vol. 104, pp. 163–170, 1998.
- [24] W. Menke, *Geophysical Data Analysis: Discrete Inverse Theory*, Academic Press, Inc., San Diego, California, 1989.
- [25] J. A. K. Suykens, J. P. L. Vandewalle, and B. L. R. De Moor, *Artificial Neural Networks for Modelling and Control of Non-Linear Systems*, Kluwer Academic Publishers, Boston, 1996.
- [26] A. Chichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, John Wiley and Sons, Inc., Toronto, 1994.

- [27] S. Haykin, *Neural Networks: A Comprehensive Foundation*, MacMillan College Publishing Company, Englewood Cliffs, New Jersey, 1994.
- [28] R. Fletcher, *Practical Methods of Optimization*, John Wiley, New York, 1980.
- [29] D. S. Watkins, *Fundamentals of Matrix Computations*, John Wiley and Sons, Inc., Toronto, 1991.
- [30] J. T. Tou and R. C. Gonzales, *Pattern Recognition Principles*, Addison-Wesley, New York, 1974.
- [31] J. L. A. van Rieckevorsel and J. de Leeuw, *Component and Correspondence Analysis*, John Wiley and Sons, Inc., Toronto, 1988.
- [32] T. Cottreau, "Transcription scientific log: Sea trial Q233 May 21 - June 3, 1996," contractor report CR/96/459, DREA, 1996.
- [33] T. Cottreau and S. MacDonald, "Recording, down-sampling and transcription of non-acoustic data from trial Sable (Q233)," contractor report CR/96/458, DREA, 1996.
- [34] W. C. Risley and A. A. Bruce, "The omega acoustic research system," in *IEEE Proceedings of Oceans 93*, 1993, vol. II, pp. 59-64.
- [35] J. Benson, N. R. Chapman, and A. Antoniou, "Geoacoustic model inversion with artificial neural networks," in *IEEE Symposium on Advances in Digital Filters and Digital Signal Processing*, 1998.

Appendix A

ORCA Input Files

```

      ### SVP FILE FOR ORCA NORMAL MODE MODEL ###
      ### ENTER DATA AFTER LINES WITH * IN FIRST COLUMN ###
      ### USE OTHER LINES FOR COMMENTS ###
Note: ap,as = p-wave, s-wave attenuations (pos~db/m/kHz, neg~dB/wavelength)
*(1) ver_no  'SVP TITLE':
      1.1  Calibration environment
-----
*(2) cp      cs      rho      ap      as      For Upper Halfspace:
      343.0    0.0    .00121    0.0    0.0
-----
nsvp = number of SVP points in ocean to be read on Line (4)
ctol = tolerance used in fitting SVP to eliminate layers (0=keep all layers)
*(3) nsvp  ctol
      2    1.0
-----
Ocean SVP (add constant rho, ap in ocean on first line):
*(4)  z      cp
      0  1480  1  0
      90  1460
-----
*(5) nbot (# bottom layers to be read in Line 6):
1
-----
Layer Profiles: type: 1=linear, 2=blug, 3=blug, 4=blug;  examples:

```

```

1 h cp1 cp2 cs1 cs2 rho1 rho2 ap1 ap2 as1 as2
2 h cp1 g cs1 cs2 rho1 rho2 ap1 ap2 as1 as2 beta ctol
3 h cp1 cp2 cs1 cs2 rho1 rho2 ap1 ap2 as1 as2 beta ctol
4 h cp1 cp2 cs1 cs2 rho1 rho2 ap1 ap2 as1 as2 g ctol
*(6) ii h cp1 cp2/g cs1 cs2 rho1 rho2 ap1 ap2 as1 as2 beta/g ctol
1 50 1550 1700 0 0 1.5 1.5 -.23 -.23 0 0

```

```

*(7) cp cs rho ap as For Lower Halfspace:
1800.0 0 2.0 -.23 0

```

```

*(8) ntop (# Top Layers to Be Read on Line 9):
0

```

```

*(9) ii thick cp1 cp2/g cs1 cs2 rho1 rho2 ap1 ap2 as1 as2 beta ctol

```

OPTION FILE FOR ORCA NORMAL MODE MODEL

ENTER DATA AFTER LINES WITH * IN FIRST COLUMN

USE OTHER LINES FOR COMMENTS

==> TYPE(S) OF COMPUTATIONS DESIRED:

ver_no = Version Number of Option File (DO NOT CHANGE THIS);

iicw = Mode Computations (0=no; 1=CW; 2=Broadband);

iikpl = Complex k-Plane Images [0=no; 1=ln(R1*R2); 2=ln(1-R1*R2)];

iirc = PW Reflection Coeff (0=no; 1=R vs angle,f; 2=FFT file vs. angle);

iiparm = Parameter Study Using iicw,iikpl,iirc above (0=no; 1=yes);

n_env = Geoacoustic Profile (_prof) Output File (0=no; pos=# depth pts);

iifmt = Output Format (1=HDF;2=MAT;3=ASCII;0=all);

```

*(1) ver_no iicw iikpl iirc iiparm n_env iifmt
1.6 1 0 0 0 0 2

```

==> MODE COMPUTATION GENERAL PARAMETERS (iicw>0 on Line 1):

SET ALL TO 0 TO FIND MODES AUTOMATICALLY

```

iirx  = Real axis version (0=no, 1=yes);
cphmin = Min Phase Speed (0=p-wave modes only; -1=seismic modes also);
cphmax = Max Phase Speed [0=use rmin; pos=speed; neg=max angle(deg)];
rmin   = Min Range of Interest in km [>999=use cphmax; 0=use S/R geom];
rmax   = Max Range of Interest in km (0=use S/R geom);
phfac  = Phase Step Parm: Step by 2*pi/phfac (set to 4-8, 0=default==>4);
db_cut = Modes Weaker By db_cut Ignored (set to 30-60, 0=default==>50);
iifb   = False Bottom (0=no, 1=def, >1=thickness in lambda);
iigbs  = Gaussian Beam Source (0=no,1=yes) [iirx=0 only; add beam angle
      and beamwidth for each source depth in lines (5) or (8)];
iidiag = Print Diagnostic Msgs;
*(2) iirx  cphmin  cphmax   rmin  rmax  phfac  db_cut  iifb  iigbs  iidiag
      0      0      0      0    0    0      0      0    0      0      0

```

CW MODE PARAMETERS ON NEXT 5 LINES (iicw=1 on Line 1)

Frequencies (nf>0 ==> List f's; nf<0 ==> List first,last f):

```

*(3) nf f1 f2 ...
      1 35

```

CW Mode Output Options:

```

iitl  = TL (0=no;1=zs,zr,r on Line 5; 2=src track/rec array on Line 10)
iimf  = Mode Functions (0=no,1=p-wave,2=s-wave,3=both). Enter depths on Line 6.
iisig = Disp uz,ux and Stress sigzz,sigzx (0=no,1=yes). Enter depths on Line 6.
iimt  = Mode Trajectory in k-plane (ASCII file).
iidc  = Disp Curves(0=no,1=vg,2=vph,3=both);
iikn  = Mode Eigenvalues Re(kn),Im(kn);
iieig = List of eigenvalue characteristics;
iikrak,iioas,iifepe,iimlab = Output Kraken, OASES, FEPE, MODELAB input files;
*(4) iitl iimf iisig iimt iidc iikn iieig iikrak iioas iifepe iimlab
      -1  0  0  0  0  0  0  0  0  0  0  0

```

For iitl=1, Enter Source Depths, Receiver Depths, Ranges:

```

*(5) nzs zs1 zs2 (m) ... nzs zr1 zr2 (m) ... nr r1 r2 (km) ...
      1 20
      -10 10 100
      1 1

```

For iimf>0 or iisig=1, enter Re/Im(0-3), Mag/Phase(0-3) Options and Depths:

```

*(6) iir/i iim/p nzm zm1 zm2 ...

```

BB MODE COMPUTATION PARAMETERS (iicw=2 on Line 1):

```

fs      = Sample f;
nfft/Tw = pos=NFFT (power of 2); neg=Time Window in s;
iifft   = Output FFT File (0=no;1=zs,zr,r on Line 8; 2=read file on Line 10);
iiout   = Output BB Eigenvalues and Functions (same options as iifft above);
iift    = Freq Traj(ASCII); iimt = Mode Traj(ASCII);
iidc    = Disp Curves (0=no,1=vlg,2=vph,3=both);
iimf    = Mode Functions at Rec Depth on Line 8 (0=no,1=2D HDF,2=3D HDF);
*(7) fs nfft/Tw fmin fmax iifft iiout iift iimt iidc iimf

```

For iifft=1, iiout=1, or iimf>0, enter src depths, rec depths, ranges:

```

*(8) nzs zs1 zs2 (m) ... nzs zr1 zr2 (m) ... nr r1 r2 (km) ...

```

PARAMETER STUDY (iiparm=1 on Line 1):

```

nrun   = # of CW mode runs;
nparm  = # of parameters to vary;
rseed  = Random # seed (integer) [0=none (vary linearly from val1 to val2)];
obt    = Medium (0=ocean, 1=bottom layers, 2=top layers);
nlay   = Layer #;
ktb    = Bottom/Top of Layer (1=top, 2=bottom, neg=maintain gradient);

```

pc = Parm Code (0=h, 1=cp, 2=cs, 3=rho, 4=ap, 5=as);
 val1,val2 = First, Last Values of Parameter;
 *(9) nrun nparm rseed [(obt nlay ktb pc val1 val2),j=1,nparm]

 S/R GEOM FOR SRC TRACK AND REC ARRAY (iitl=2 on Line 4 or iiff=2 on Line 8)
 *(10) zs n_src_seg 'array_geom_file_name'

 Source Track: v m/s; t , dt min; cpa , x , y km; ϕ deg E of N (see manual)
 1 iic v t1 t2 +dt/-nt cpa phi (POLAR FORM)
 *(11) 2 iic v t1 t2 +dt/-nt x1 y1 x2 y2 (X-Y FORM)

 ==> k-PLANE IMAGES (iikpl>0 on Line 1)
 iivar = Variable to Image [1=R1*R2, 2=d(R1*R2)/dk, 3=d(R1*R2)/dw];
 iikf = Form for k_r, k_i (1=k/kmin, 2=k, 3= $k_r \sim k_r/k_{min}, k_i \sim dB/km$);
 kdud = Duct Number for Reference Depth (0 for min cp in ocean);
 iiph = Phase Wrap (1=-180,180; 2=0,360);
 iishp,iishs = Riemann Sheet for p/s Branch Cuts (+1 or -1);
 *(12) freq iivar iikf kr1 kr2 nkr ki1 ki2 nki nduct iiph iishp iishs

 ==> REFLECTION COEFFICIENT VS f (Hz) AND GRAZING ANGLE (deg)
 R vs Frequency and Angle (iirc=1 on Line 1):
 *(13) freq1 freq2 nfreq iilog theta1 theta2 ntheta

 FFT File of R vs. Frequency and Angle (iirc=2 on Line 1):
 *(14) freq1 freq2 fs nfft theta1 theta2 ntheta

VITA

Surname: Benson

Given Names: Jeremy

Place of Birth: Canada

Date of Birth: Aug. 13, 1972

Educational Institutions Attended

University of Victoria	1996 to 1998
University of Saskatchewan	1991 to 1996

Degrees Awarded

B.E.	University of Saskatchewan	1996
------	----------------------------	------

Conference Publications

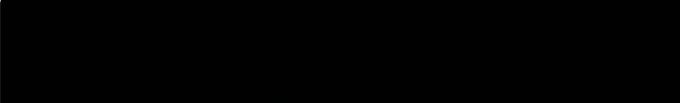
1. J. Benson, N.R. Chapman, A. Antoniou, "Geoacoustic Model Inversion Using Artificial Neural Networks", *Proceedings DFSP '98, Symposium on Advances in Digital Filtering and Signal Processing*, Victoria, BC, CANADA pp. 121-125, Jun. 1998.

PARTIAL COPYRIGHT LICENSE

I hereby grant the right to lend my thesis to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Title of Thesis: APPLICATION OF ARTIFICIAL NEURAL NETWORKS
TO GEOACOUSTIC INVERSION.

Author:


JEREMY PAUL JAMES BENSON
14 December 1998