NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

CLASSIFICATION OF UNDERWATER SIGNALS USING WAVELET-BASED DECOMPOSITIONS

by

Ozhan Duzenli

June, 1998

Thesis Advisor:	Monique P. Fargues
Co-Advisor:	Ralph D. Hippenstiel

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE June 1998	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE:
CLASSIFICATION OF UNDERWATER SIGNALS USING WAVELET-BASED DECOMPOSITIONS

5. FUNDING NUMBERS

6. AUTHOR(S) Duzenli, Ozhan

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This thesis investigates the application of wavelet decompositions to classification applications. Two feature extraction tools are considered: Local Discriminant Bases scheme (LDB) and Power method. Several dimension reduction schemes including a newly proposed one called the Mean Separator neural network (MS NN) are discussed. Two types of classifiers are investigated and compared: Classification Trees (CT) and Back-propagation neural network (BP NN). Classification experiments conducted on synthetic and real-world underwater signals show that: 1) the Power feature extraction method is more robust to time synchronization issues than the LDB scheme is; 2) the MS NN scheme is a successful dimension reduction scheme that may be used with both LDB and Power feature extraction methods; and 3) the BP NN is a more powerful classifier than CT as it has fewer constraints than CT in partitioning the feature input space.

14. SUBJECT TERMS
Classification, Wavelet Decomposition, Local Discriminant Bases (LDB), Dimension Reduction, Classification Trees (CT), Back-propagation Neural Network (BP NN), BCM

15. NUMBER OF PAGES
179

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

Approved for public release; distribution is unlimited.

CLASSIFICATION OF UNDERWATER SIGNALS USING WAVELET-BASED DECOMPOSITIONS

Ozhan Duzenli
Lieutenant Junior Grade, Turkish Navy
Turkish Naval Academy, 1992

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1998

Author: ______________________
Ozhan Duzenli

Approved by: ______________________
Monique P. Fargues, Thesis Advisor

Ralph D. Hippenstiel
____
Ralph D. Hippenstiel, Co-Advisor

David C Jen
____
for Herschel H. Loomis, Jr., Chairman
Department of Electrical and Computer Engineering

ABSTRACT

This thesis investigates the application of wavelet decompositions to classification applications. Two feature extraction tools are considered: Local Discriminant Bases scheme (LDB) and Power method. Several dimension reduction schemes including a newly proposed one called one the Mean Separator neural network (MS NN) are discussed. Two types of classifiers are investigated and compared: Classification Trees (CT) and Back-propagation neural network (BP NN). Classification experiments conducted on synthetic and real-world underwater signals show that: 1) the Power feature extraction method is more robust to time synchronization issues than the LDB scheme is; 2) the MS NN scheme is a successful dimension reduction scheme that may be used with both LDB and Power feature extraction methods; and 3) the BP NN is a more powerful classifier than CT as it has fewer constraints than CT in partitioning the feature input space.

TABLE OF CONTENTS

ACKNOWLEDGMENT

I want to dedicate this work to my fiance Nur GUREL who makes me desire to live forever. I also want to thank Prof. Monique P. Fargues for her guidance and patience during the work in performing this investigation.

I. INTRODUCTION

Wavelet-based decompositions have been used extensively in the last decade in various areas such as engineering, finances, and statistics. In signal processing, this tool is applied to areas such as signal compression, noise removal and signal classification.

This work considers wavelet-based decompositions as applied to classification applications. A typical classification scheme consists of three parts: a feature extraction, a dimension reduction and a classification part. Chapter II briefly reviews the wavelet decomposition, and highlights the main differences relative to the Fourier transform. In Chapter III, we investigate the application of the wavelet packet decomposition to the Local Discriminant Bases (LDB) scheme originally proposed by Saito, and show that it is sensitive to time synchronization problems. Then we introduce an alternative, called the Power feature extraction method. This method is based on frequency band specific power quantities, which are more robust to time synchronization issues without worsening the classification performance. This chapter also presents four dimension reduction schemes associated with the Power feature extraction method: *Learned and Willsky's*, *most consistent*, *most discriminating* and *LDB based* dimension reduction schemes. Several examples are implemented to give some insights about the feature extraction and dimension reduction schemes introduced in this chapter. Chapter IV presents and compares two types of classifiers: back-propagation neural networks (BP NN) and classification trees (CT). Chapter V considers several feature extraction and dimension reduction methods. These steps are key in obtaining good classification performance when the amount of data available to build the classification tools is limited, or when subject to computer capability constraints. We consider the BCM neural network implementation, which can be used as a feature reduction scheme, and show that it is computationally slow. As an alternative we propose a mean separator neural network (MS NN), initially designed to distinguish between two classes, and extend it to the more-than two-classes case. We also show that the MS NN can be followed by a decision step to create a stand alone classification scheme which has a performance comparable to that obtained with more sophisticated classifiers at a fraction of the computational cost. In Chapter VI, we investigate the behavior of the various schemes and consider both a synthetic and a real-world underwater signal. This demonstrates that the proposed MS NN is a successful dimension reduction scheme that may be used with both LDB and Power feature extraction methods. Finally, conclusions are presented in Chapter VII.

II. WAVELETS ANALYSIS

Wavelet analysis has been used extensively in the last decade in various fields from engineering to finances, and can be viewed as a complement to the well-known Fourier transform method [6,18]. Thus, we will first review the Fourier transform before presenting the basic concepts behind wavelet-based decompositions. Note that at this point the discussion is restricted to discrete time functions, as only discrete time domain signals are considered in this work.

A. DISCRETE-TIME FOURIER ANALYSIS

1. The Discrete-Time Fourier Series

Recall that a periodic function *x(n)* with period *N* may be defined as a linear combination of periodic complex exponentials with amplitude *A(k)* [1]:

$$x(n) = \sum_{k=0}^{N} A(k)\, e^{j\frac{2\pi nk}{N}}, \; n=0,1,\ldots\ldots,N\text{-}1. \qquad (2.1)$$

Identifying the complex amplitude terms $A(k)$ can be done by evaluating Equation 2.1 for n=0,1,2,. . . . ,N-1, which results in N linear equations with N unknowns:

$$x(0) = \sum_k A(k), \qquad (2.2)$$

$$x(1) = \sum_k A(k) e^{j\frac{2\pi k}{N}},$$

.

.

.

$$x(N-1) = \sum_k A(k) e^{j\frac{2\pi k(N-1)}{N}}.$$

It can be shown that the above set of N equations is linearly independent and can be solved to obtain the values $A(k)$ [1]. However, for practical purposes a closed form expression for calculating $A(k)$ is more desirable. Note that, if both sides of Equation 2.1 are multiplied by the term, $e^{-j\frac{2\pi rn}{N}}$, where r is an integer, and the resulting expression summed over N terms gives:

$$\sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi rn}{N}} = \sum_{n=0}^{N-1}\sum_{k=0}^{N-1} A(k)e^{j\frac{2\pi(k-r)n}{N}} . \tag{2.3}$$

Interchanging the order of the summations appearing in Equation 2.3 results in:

$$\sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi rn}{N}} = \sum_{k=0}^{N-1} A(k)\sum_{n=0}^{N-1} e^{j\frac{2\pi(k-r)n}{N}} . \tag{2.4}$$

It can be shown that the rightmost term contained in Equation 2.4 is equal to zero unless the term $(k\text{-}r)$ is zero, or is an integer multiple of N [1]. As a result, the rightmost summation expression contained in Equation 2.4 is equal to N only if $k=r$ and equal to zero otherwise. Thus, the amplitude term $A(k)$ can be derived from Equation 2.4 as:

$$A(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}} , k=0,1,....N\text{-}1. \tag{2.5}$$

The magnitudes of the coefficient terms A(k) expressed as a function of the frequency index k form the magnitude spectrum of the time domain signal $x(n)$. The frequency presentation of nonperiodic signals can be found with a similar method by assuming that the signal is periodic with period equal to the signal length N [2]. The resulting discrete frequency coefficients are then calculated using Equation 2.5.

The coefficients obtained using Equation 2.5 are one of the possible candidates for feature selection in classification tasks, as they represent the amount of power associated with the signal in a given frequency band [3].

2. Discrete-Time Fourier Series Transform And Filter Banks

The discrete Fourier transform coefficients $A(k)$ can also viewed as the outputs of a bank of FIR filters followed by decimators, as shown in Figure 2.1, where the decimation operator keeps every N$^{\underline{th}}$ term obtained in the filter outputs. Such a connection is illustrated next by deriving Equation 2.5 using the filter bank approach. Let us assume the impulse response for the k$^{\underline{th}}$ filter shown in Figure 2.1 is defined as:

$$H_k(n) = \frac{1}{N} e^{-j\frac{2\pi k(N-1-n)}{N}}, n=0,1,....,N-1,\ k=0,1,...,N-1. \qquad (2.6)$$

Then using the convolution sum, the filter output $y_k(n)$ can be expressed as:

$$y_k(n) = \frac{1}{N} \sum_{m=n-N+1}^{n} x(m) e^{-j\frac{2\pi k(N-1-n+m)}{N}}, n=0,1,......,\infty,\ k=0,1,...,N-1. \qquad (2.7)$$

At this point, note that only the N$^{\underline{th}}$ output value is kept after the decimation operation, leading to the output of the k$^{\underline{th}}$ branch as:

$$y_k(N-1) = \frac{1}{N} \sum_{m=0}^{N-1} x(m) e^{-j\frac{2\pi km}{N}}. \qquad (2.8)$$

Comparing Equations 2.5 and 2.8 shows that $A(k) = y_k(N-1)$, which validates the filter bank approach. This approach can also be viewed as using FIR matched filters. Recall that a matched filter gives a high output if the input signal looks like the impulse response of the filter. Thus the coefficients *A(k)* indicate how close the input signal is to the set of filter impulse responses defined in Equation 2.6.

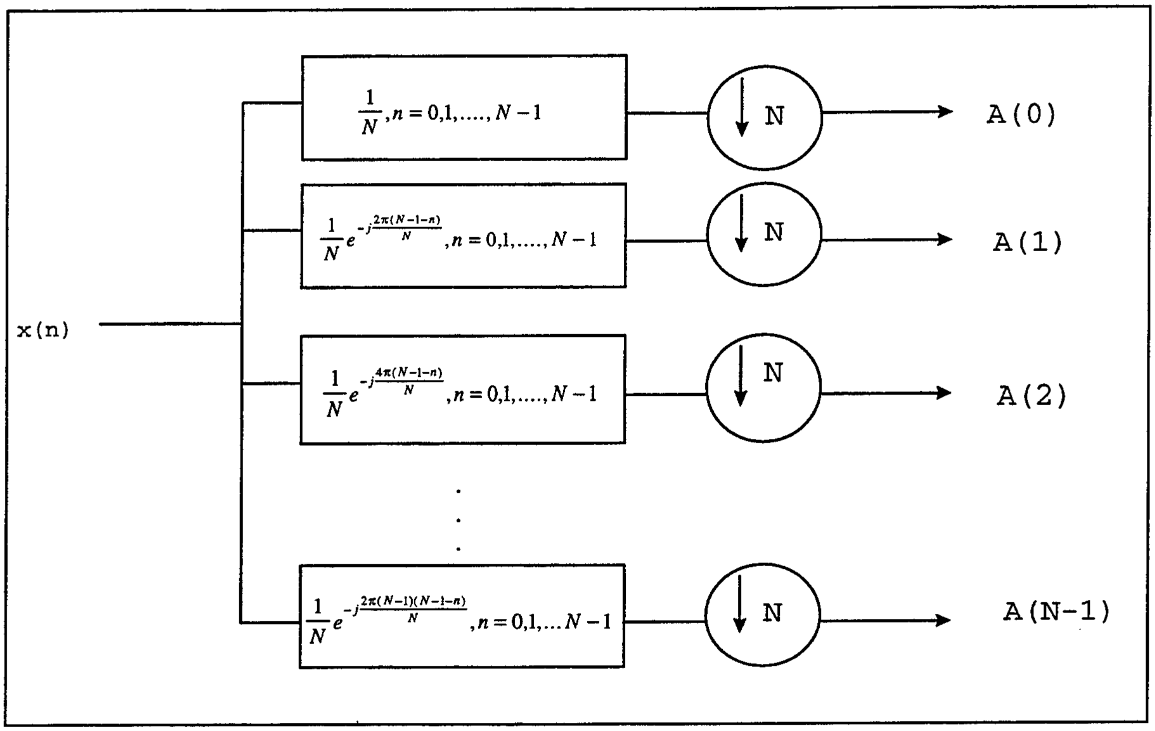
x(n)
$\frac{1}{N}, n=0,1,\ldots..,N-1$
$\frac{1}{N}e^{-j\frac{2\pi(N-1-n)}{N}}, n=0,1,\ldots..,N-1$
$\frac{1}{N}e^{-j\frac{4\pi(N-1-n)}{N}}, n=0,1,\ldots..,N-1$
$\frac{1}{N}e^{-j\frac{2\pi(N-1)(N-1-n)}{N}}, n=0,1,\ldots N-1$
N
N
N
N
A(0)
A(1)
A(2)
A(N-1)

Figure 2.1: Discrete Fourier series transform interpretation as a filter bank.

3. DFT Coefficients as Feature Parameters

Classification tasks are usually two-step processes, as one must first extract relevant feature parameters which accurately characterize each signal class, prior to classifying the data. The feature selection or extraction process has been extensively studied [12, 13, 17, 21] and we will address it in later chapters. Signal energy quantities have been used as a simple choice of feature parameters, as they are easy to compute and often lead to good results. Recall that the magnitude squared of the k<u>th</u> discrete Fourier series coefficient, $|A(k)|^2$, represents the amount of signal energy in the frequency band centered at $\frac{2k\pi}{N}$ with bandwidth $\frac{2\pi}{N}$. Such "frequency band"-specific energy quantities have also been selected as feature parameters, and used as inputs to a back-propagation neural network in numerous implementation [23, 24]. For example, simulations using underwater biological signals showed that the resulting classification rates exceed 90%, when used on properly segmented data [23, 24]. Here, the problem becomes the selection of the frequency bands that best discriminate between the signal classes to reduce the number of feature parameters. Note that such feature selection schemes are very different from those applied in compression applications, where the selection criterion is designed to minimize the difference between original and compressed signals. As a result, frequency bands with the high energy are kept in compression applications. Such a selection may not be valid for classification tasks, where the class discriminant information may be contained in frequency bands of relatively low energy. Discriminant selection schemes are addressed further in Chapter III.

4. Short-Time Fourier Transform

As mentioned earlier, the Fourier transform allows the user to obtain the frequency content of the time domain signal. However, this method is not very useful if the signal frequency representation changes with time [5], as is the case for non-stationary signals. In such a case, the frequency information obtained with the Fourier transform represents the average frequency behavior observed in the time interval over which the Fourier transform is computed. A more accurate representation of the time-varying nature of the frequency information is obtained with the short-time Fourier transform (STFT) as the STFT mapping is from the time domain space to a two-dimensional time-frequency representation.

The main idea behind the STFT is the introduction of a finite-time moving window *w(n)* of length *N* in which the signal frequency content is computed via the Fourier transform. The window length is selected so that the signal is considered to be stationary over the window length. Thus, the short-time Fourier transform of a given time domain signal *x(n)*, using a window *w(n)*, is defined as:

$$A(n,f) \equiv \sum_{k=-\infty}^{\infty} x(k)w(n-k)e^{-j2\pi fk}\,. \qquad (2.9)$$

The resulting two-dimensional coefficient *A(n,f)* has two indexes; *n* represents the time index, while *f* represents the frequency. Thus *A(n,f)* represents the time-varying frequency information of the time domain signal *x(n)*. The square of the magnitude, $|A(n,f)|^2$, is called the spectrogram. For example, Figure 2.2 shows the spectrogram obtained from the signal *x(n)* which is the sum of a constant tone at frequency 0.5 Hz and a linear chirp with sweep rate $\frac{1}{4096}$ Hz/sec, which are sampled at 2 Hz:

$$x(n) = \sin(\frac{0.5\pi n^2}{8192}) + \sin(0.5\pi n),\ n=0,1.....8191. \qquad (2.10)$$

Note that different types of window functions can be used to compute the STFT, resulting in different time-frequency resolutions. However, recall that the product of the time duration window size Δt and the frequency bandwidth Δf of any signal has a lower bound, given by $\frac{1}{4\pi}$, due to the Heisenberg's uncertainty principle [5]. The specific time-frequency partitioning is fixed by the specific choice of time window and one cannot obtain good time and good frequency resolution simultaneously. Thus, the STFT is well-suited to analyze signals which are either narrowband (a good frequency resolution can be obtained by selecting a long time-window), or wideband (a good time-resolution is obtained by selecting a short time-window). However, the STFT is ill-suited to analyze signals which exhibit both narrowband and wideband components, as a fixed window will not be able to analyze both types of components well. The window length restriction is one of the main problems associated with the STFT. Wavelet analysis addresses this shortcoming by defining a two-dimensional time-frequency transform with a variable time window length.

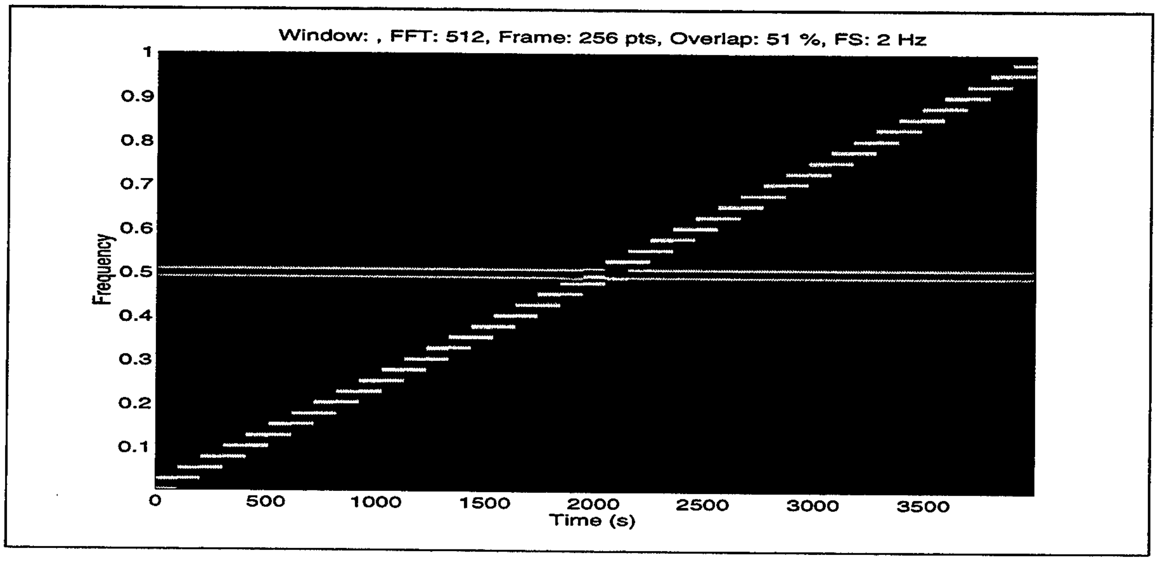

Figure 2.2: Spectogram plot of a linear chirp and a single tone.

B. WAVELET ANALYSIS

1. The Continuous Wavelet Transform

The easiest way to understand the basic concept behind wavelet analysis is to compare it to the STFT method mentioned earlier. Recall that the STFT is computed by moving a windowed function $w(n)e^{-j2\pi fn}$ along the time axis, and computing the inner product between the signal *x(n)* and the windowed function [5]. Now assume we use a function $\Psi_{a,b}(t)$ in place of the windowed function in the STFT definition, where $\Psi_{a,b}(t)$ is defined in terms of a function $\Psi(t)$, defined with specific properties as:

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\Psi(\frac{t-b}{a}), \quad a,b \in R, a \neq 0. \tag{2.11}$$

Note that $\Psi_{a,b}(t)$ has two variable parameters: *a* and *b*. The index *b,* called the time shift, allows for time shifting of $\Psi_{a,b}(t)$, while the index *a*, called the scale, allows the function $\Psi_{a,b}(t)$ to expand or contract. These two indexes allow for the definition of a two-dimensional transformation which uses a time window of varying length, depending on the value chosen for *a*. Such a definition leads to a varying time-frequency partitioning. The function $\Psi(t)$ is called the mother wavelet. The continuous wavelet transform is defined as:

$$W_f(a,b) = \left\langle f(t), \Psi_{a,b}(t) \right\rangle, \tag{2.12}$$

where the notation "< >" denotes the inner product.

Several types of mother wavelet functions $\Psi(t)$ can be defined, which offers more flexibility than the STFT where the basis function type is restricted to that of a windowed complex exponential. However, the wavelet function must satisfy two important conditions: 1) The wavelet function $\Psi(t)$ should be of finite time duration; 2) the area under $\Psi(t)$ should be equal to zero [6]. There are numerous functions that satisfy these conditions. Examples, such as Daubechies, Haar, Coiflet, and Symmlet wavelets are plotted in Figure 2.3.

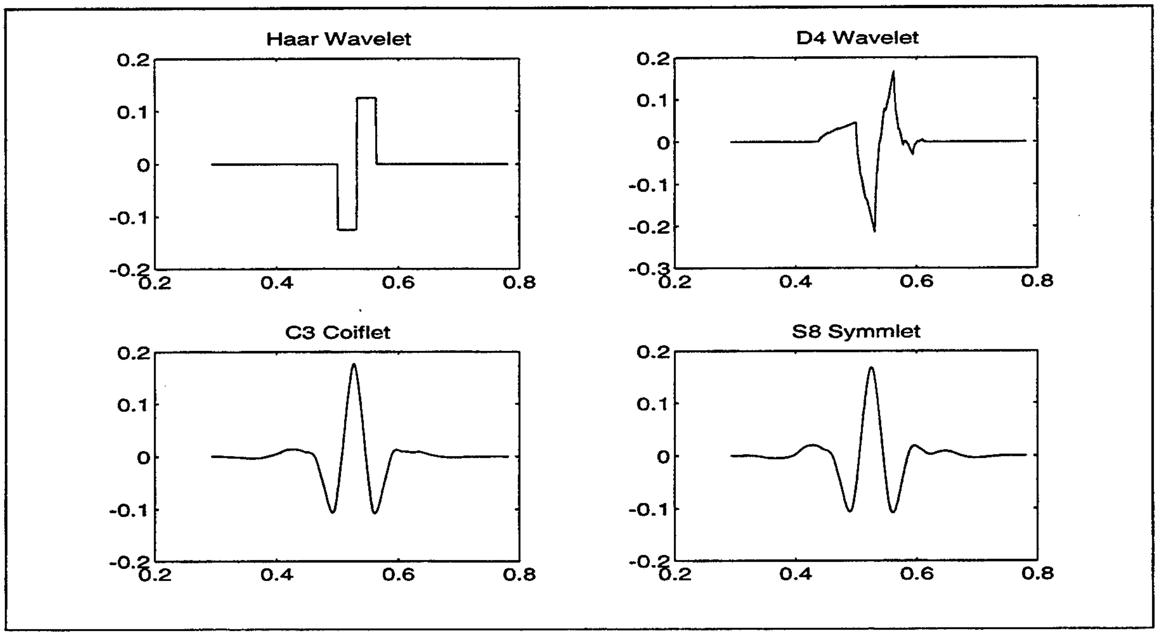

Figure 2.3: Four wavelets in the time domain, from Ref. [10].

Let us further expand on the meaning of the indexes *a* and *b*, which are key to understanding the power of the wavelet decomposition. By convention, a low scale (i.e., small value of the index *a*) leads to a high frequency wavelet function $\Psi_{a,b}(t)$ which provides good time resolution with poor frequency resolution. Conversely, a large value for the index *a* refers to a low frequency wavelet function $\Psi_{a,b}(t)$, which provides poor time resolution with good frequency resolution [6,8]. This behavior is further illustrated in Figure 2.4 which plots the function $\Psi_{a,b}(t)$ obtained for various sets of indexes (a,b) = { (7,95), (6,43), (6,32), ... }, for a Symmlet-8 wavelet. Figure 2.4 clearly shows that as the scale decreases, the wavelet function becomes more localized in time but its frequency resolution becomes poorer.

The magnitude squared of the wavelet coefficients $W_f(a,b)$ plotted as a function of the indexes a and b shows the energy distribution of the signal in the time-scale plane, and is called the scalogram [9].

2. STFT and WT comparisions

In this section, we compare the STFT and the wavelet transform (WT) using two simple examples.

a) Wideband signal

Consider a delta function located at $t{=}to$ in the time domain. The resulting spectrogram is shown in the top left plot contained in Figure 2.5. Note that the time domain uncertainty in localizing the impulse location is constant for all frequencies when using the STFT, as expected, as the time window length is fixed once selected. Thus, it may be difficult to estimate the exact occurrence of the impulse when a long time window is selected. The top right plot in Figure 2.5 shows the scalogram obtained for the same impulse located at time $t{=}to$. Note that the scalogram leads to a better localization of the impulse, since a good time resolution is obtained at low scales (i.e., high frequencies).

b) *Narrowband signal*

Now assume we have two sinusoidal signals with low and high frequencies. The spectrogram plot in the bottom left plot of Figure 2.5 shows that both sines have the same frequency resolution, due to the constant time window for the STFT. However, the frequency resolution is not constant in the WT, which leads to the bottom right plot of Figure 2.5. In this case, the sinusoidal component with higher frequency has a poorer frequency resolution. The frequency resolution is a direct result of allowing for time windows of varying length in the time domain. These two simple examples point out an important feature in the wavelet transform; it is well matched to real-world signals that are transient having high frequencies or are of relative long duration at low frequencies. In general, the WT can handle signals which contain both low frequency narrowband and wideband components, while the resolution of the STFT is fixed by the specific choice of the time-window.

The multiresolution WT time-frequency mapping and that of the STFT are plotted in Figure 2.6.

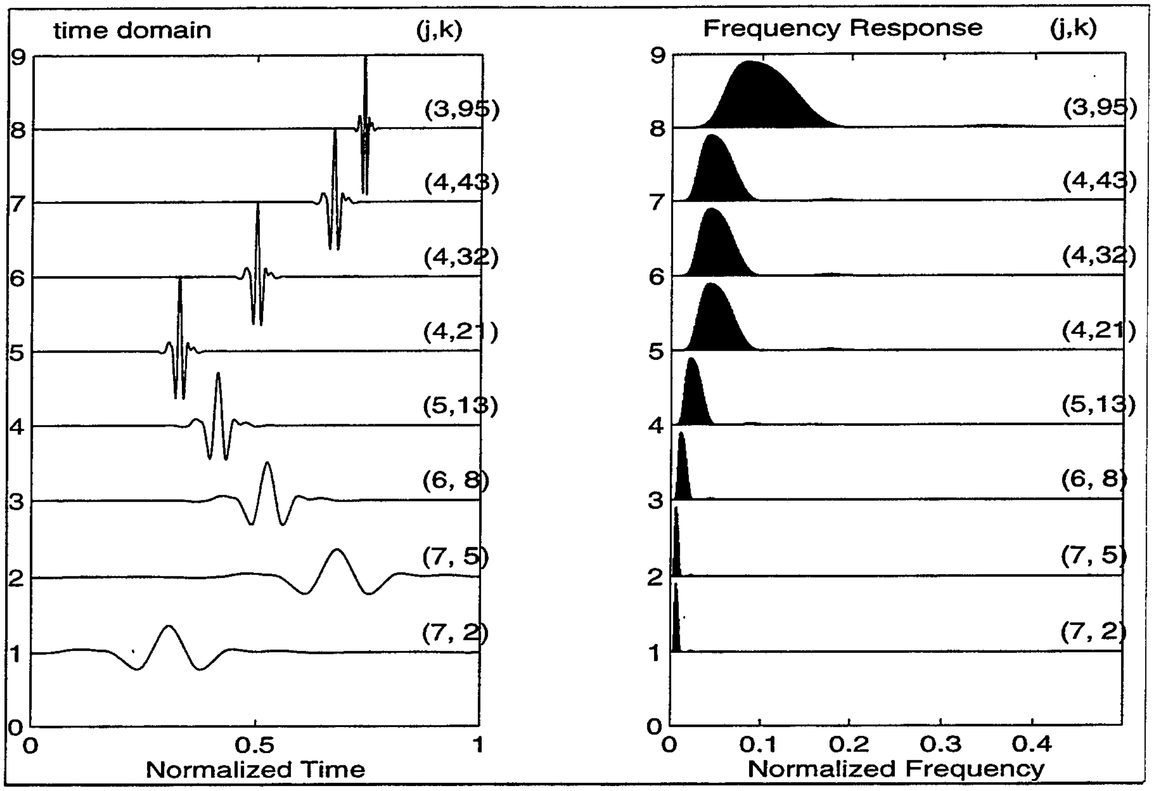

Figure 2.4 :Symmlet-8 wavelet in the time and frequency domains, from Ref. [7].

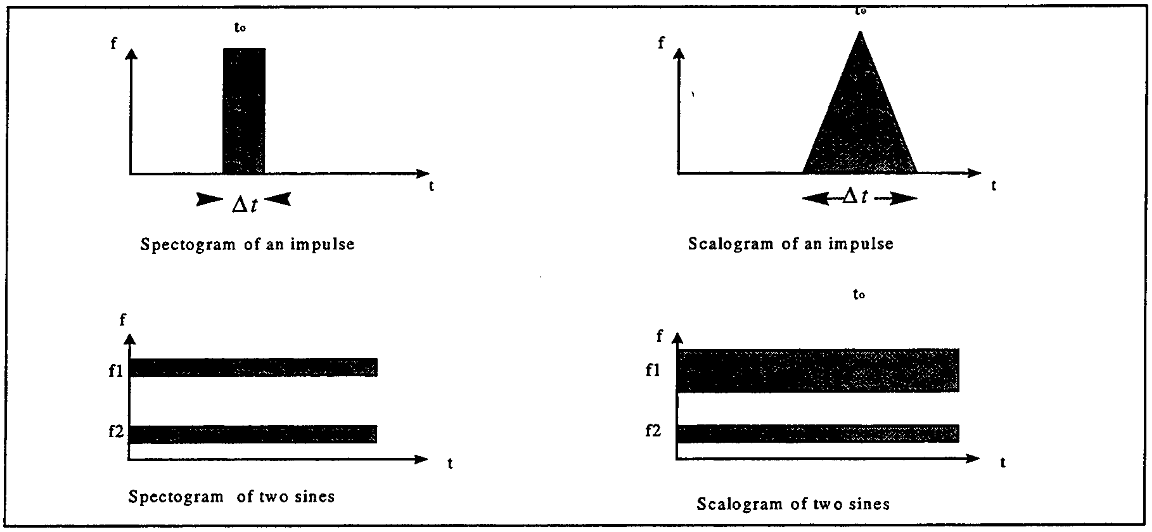

Figure 2.5: Spectogram and scalogram plots for two signals. Top plots show transforms for an impulse function and bottom plots show transforms for two sinusoidal signals. After Ref. [10].

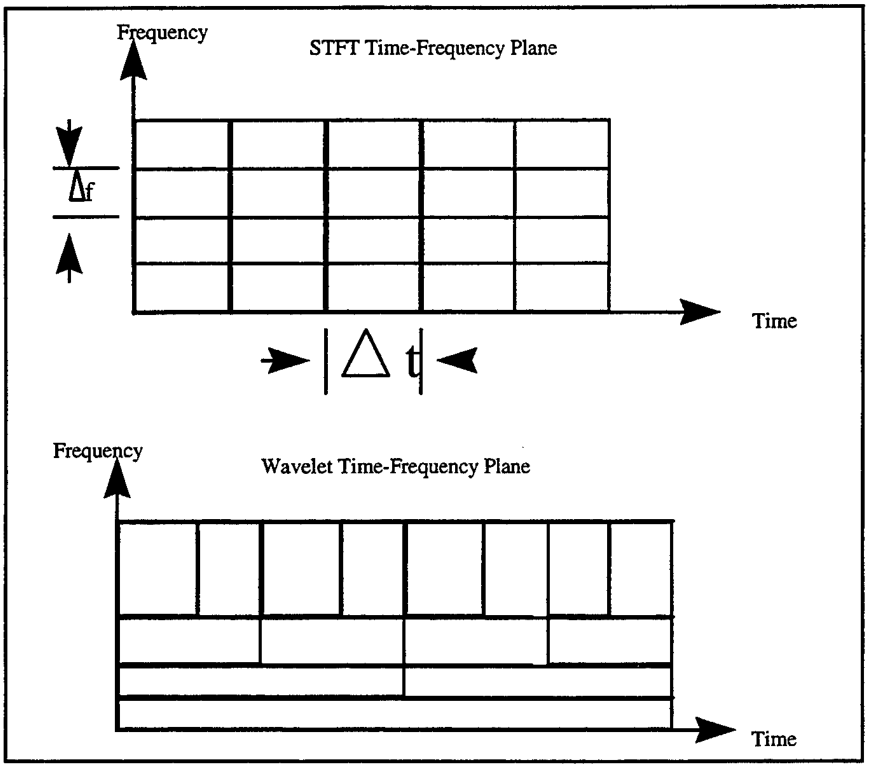

Figure 2.6: Time-Frequency plane for STFT and Wavelet Transform, after Ref. [6].

3. The Discrete Wavelet Transform

The discrete wavelet transform (DWT) is the sampled version of the continuous WT. The DWT of a time domain signal *x(n)* is defined as :

$$W_x(a,b) = \sum_n \frac{1}{\sqrt{a}} x(n) \Psi^*(\frac{n-b}{a}). \tag{2.13}$$

Note that indexes *a* and *b* take only discrete values in the DWT. The index *a,* commonly chosen as 2^j, where $j= 0,1,2....,log_2(N)$, is called the octave of the transformation. As the scale index *j* increases by one, the discrete mother wavelet function is stretched in the time domain and compressed in the frequency domain by a factor of two. Thus, the frequency resolution doubles with every scale increase. Next, if the time shifting parameter b is restricted to $k2^j$, where k is an integer, this version of the DWT is known as the decimated DWT and can be rewritten as:

$$w_{j,k} = \sum_{n} \sqrt{\frac{1}{2^j}} x(n) \Psi^*(2^{-j}n - k) , \tag{2.14}$$

where $j=0,1,2,\ldots, log2(N), k=1,2,\ldots., N2^{-j}$, and N is the length of the signal $x(n)$.

Note the number of wavelet coefficients drops to half of those contained in the adjacent lower scale. Figure 2.7 displays various scaled and time shifted versions, i.e., $\Psi\ (2^{-j}n - k)$, of the Symmlet-8 wavelet. Note that as the scale j decreases, the wavelet becomes more localized in time.

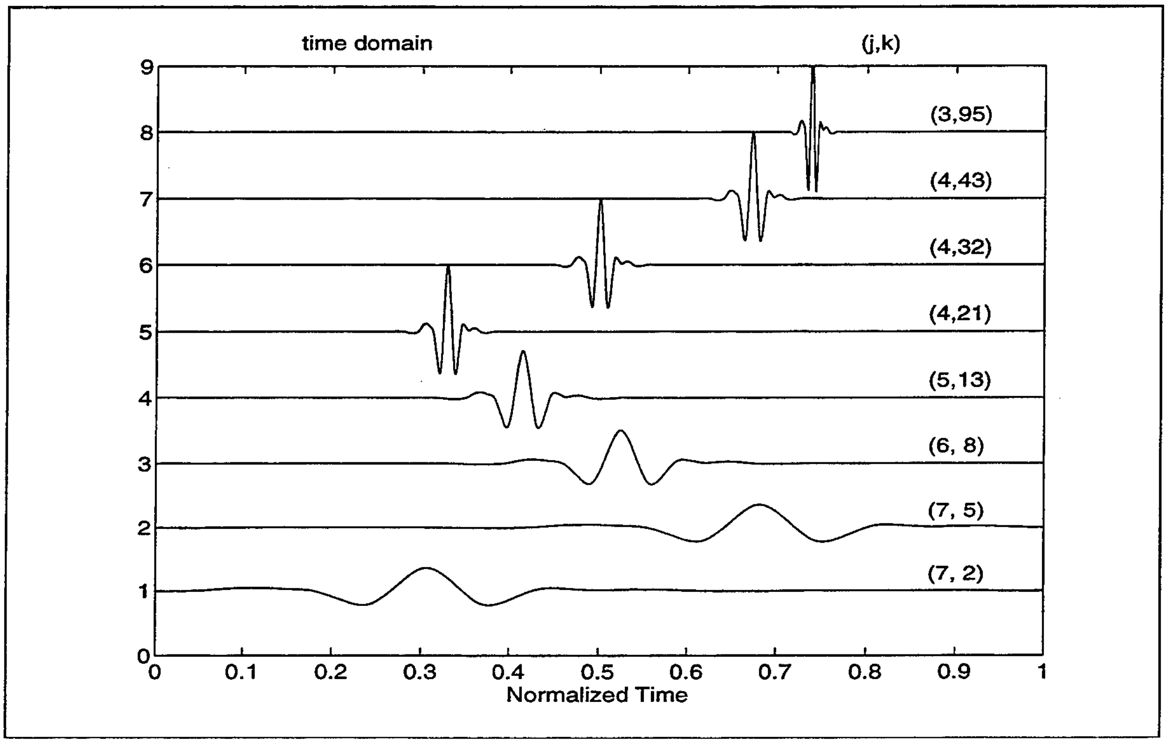

Figure 2.7: Symmlet-8 wavelet at various scales j and shifts k, from Ref. [7].

4. Multiresolution Analysis and Filterbanks

An efficient procedure to implement the DWT using filterbanks was proposed by Mallat [9]. Mallat's multiresolution algorithm is based on a pair of lowpass and highpass filters which equally partition the frequency axis. These filters, called quadrature mirror filters (QMF), must satisfy very specific properties. Further details may be found in [6, 9]. The output of the highpass (HP) filter *H(z)* contains the high frequency detail components of the signal, while the output of the lowpass filter *G(z)* contains the low frequency components, as shown in Figure 2.8. The output of each filter can then be decimated by a factor of 2, as each filter output covers only half the frequency bandwidth. The resulting decimated coefficients obtained as the HP filter output constitute the wavelet coefficients at the first scale. The decimated lowpass filter output is then passed through a highpass and a lowpass and decimated again. The decimated coefficients obtained after the highpass filter operation are the wavelet coefficients of the second scale. Filtering and decimating operations can then be repeated again, until the decimated signal is one point, if desired. Thus, the wavelet transform operation can be represented in a tree structure, as given in Figure 2.9. Note that the WT decomposition can also be represented as in Figure 2.10 by combining the successive decimation and filtering operations.

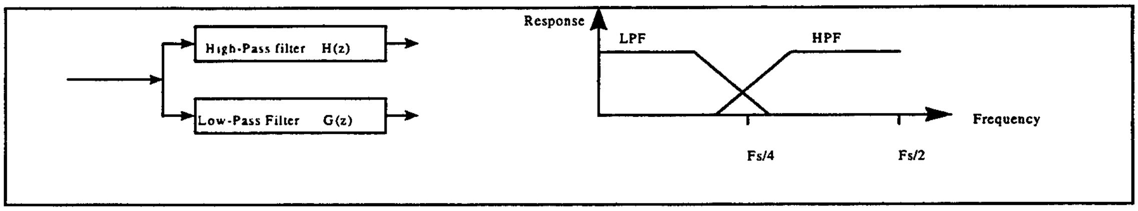

Figure 2.8: Schematic representation of Quadrature Mirror Filters (QMF), from Ref. [6].

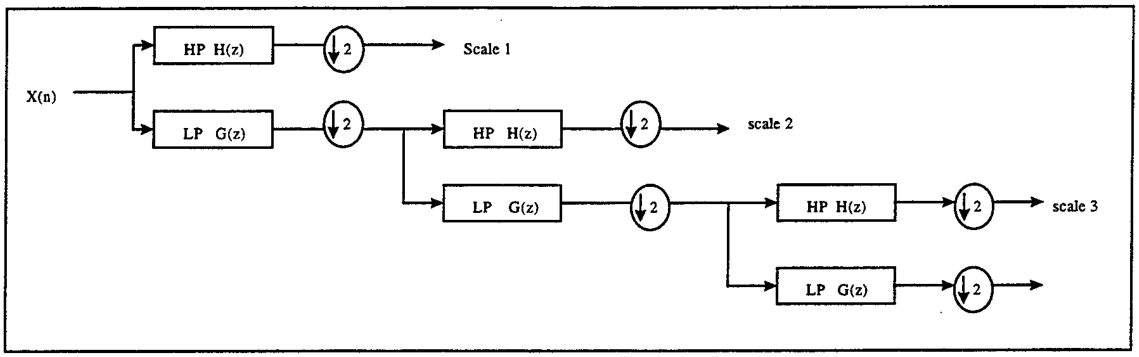

Figure 2.9: Schematic representation of the Mallat Algorithm, from Ref. [6].

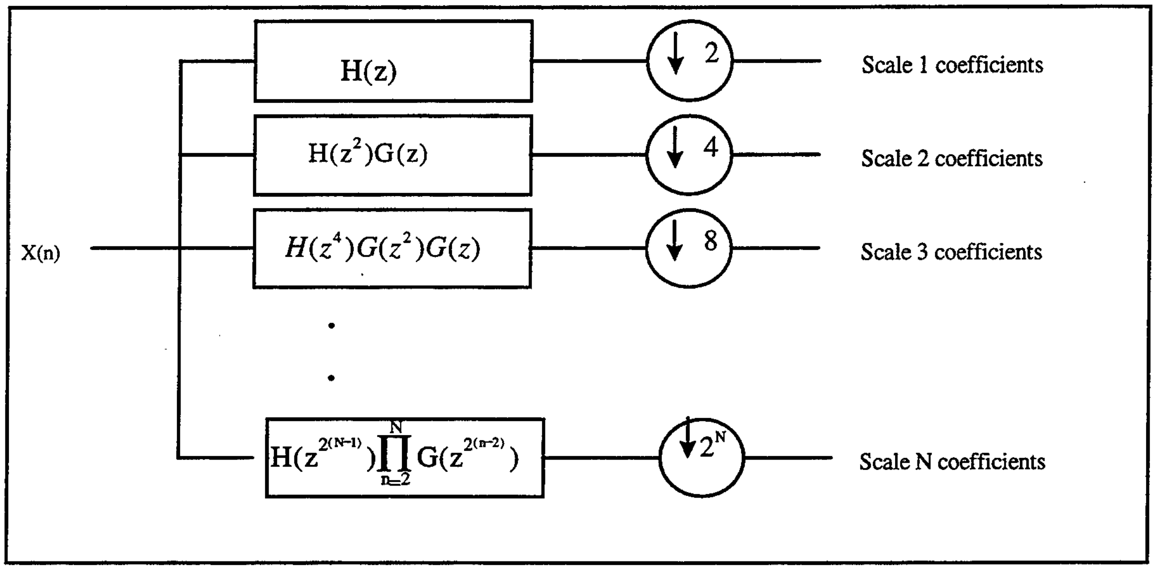

Figure 2.10: Discrete Wavelet Transform via the Filter Bank.

The decimated DWT described above leads to an orthogonal decomposition of a time domain signal only if the lowpass and highpass filters are chosen properly. Further details regarding these properties may be found in [6,9,11].

5. DWT and The Time-Varying Property

A potential drawback in the definition of the DWT is the fact that the transform is shift variant, due to the decimation operations. "Shift variant" means that the DWT coefficients obtained from the shifted time domain signal are different from the coefficients obtained from the non-shifted signal. This property of the DWT makes it difficult to use the DWT parameters as feature parameters for signal classification [12, 13, 15], as proper synchronization of the signals to be classified would be needed prior to applying the DWT decomposition. The shift variant property of the DWT coefficients associated with a linear chirp signal can be seen in Figure 2.11. Note that only a 10 step shift in the time domain signal results in drastically different DWT coefficients.

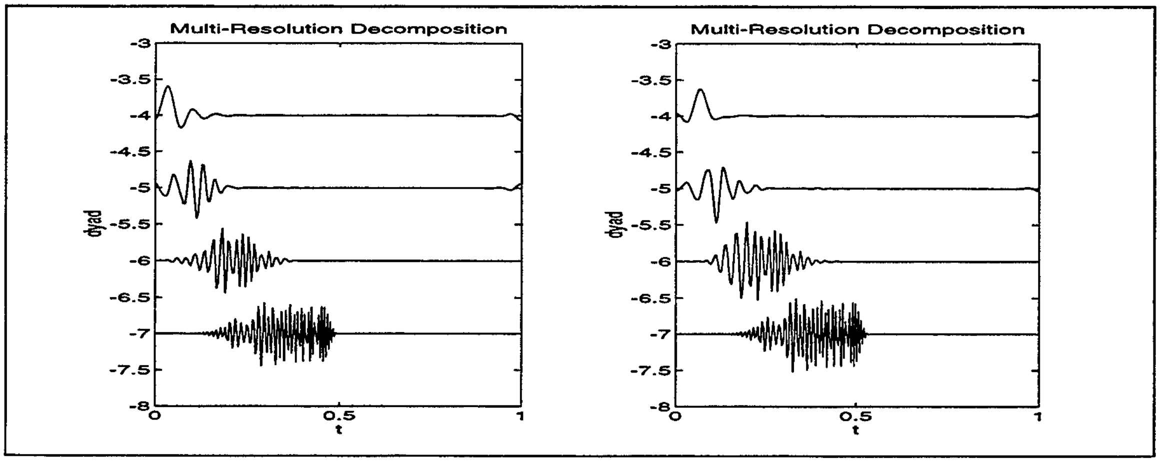

Figure 2.11: DWT coefficients of a linear chirp (left figure) and of a shifted version (right figure).

One of the methods to address DWT shift variance is cycle-spinning [14]. Basically, cycle spinning efficiently computes the averaged DWT coefficients obtained from successive shifted versions of the original time signal. Another method uses a target-entropy value to eliminate the time-variant property [15].

C. WAVELET PACKETS

Understanding the wavelet transform is a key point to understanding the wavelet packet (WP) decomposition. The DWT can be represented as a tree structure, as shown in Figure 2.12. This tree structure can be extended by passing the high-pass section of the data through quadrature-mirror filters, as was done for the low-pass portion of the data. This operation will divide the upper frequency band into two parts. Repeating this operation for each successive scale leads to the complete tree structure, as shown in Figure 2.13. The octave j associated with the scale 2^j is shown at each level. The outputs of highpass and lowpass filter combinations at each level are called “nodes.” The node numbering is performed from left to right starting from 0 at every scale so the node number (1,0) is the node at scale 1 which covers the frequency axis from 0 to Fs/4, while node (1,1) covers the frequency axis from Fs/4 to Fs/2, where Fs represents the sampling frequency. Figure 2.13 shows the node locations and the node numberings for the first 4 scales on the WP decomposition tree.

1. Basis Selection

The decomposition obtained with the full tree is redundant, as every parent node can be replaced by its two children nodes [16,17]. For example, consider the lowpass node at scale 1 (nodes number (1,0)). The information obtained at this node may also replaced with that of its two children nodes (2,0) covering the frequency band [0, Fs/8] and (2,1) covering the frequency band [Fs/8, Fs/4]. Actually, it is the inherent redundancy present in the WP decomposition that usually leads to better performances than those obtained with the WT. The WP decomposition allows for the selection of a "best" non-redundant decomposition among $2^{2^{(j-1)}}$ possible decompositions, where j is the maximum possible number of scales of a given signal [17]. The specific criterion involved in the "best" selection is left to the user who matches it to the specific application at hand, provided that it leads to a complete non-redundant coverage of the frequency axis. Note that one of the WP decomposition schemes is the DWT. Another possible decomposition is given in Figure 2.14. It is clear that the decomposition shown in Figure 2.14 has good time resolution at low frequencies. All the possible decomposition schemes form a complete orthogonal basis [17]. Next, Chapter III introduces the two feature extraction methods that are used in this work.

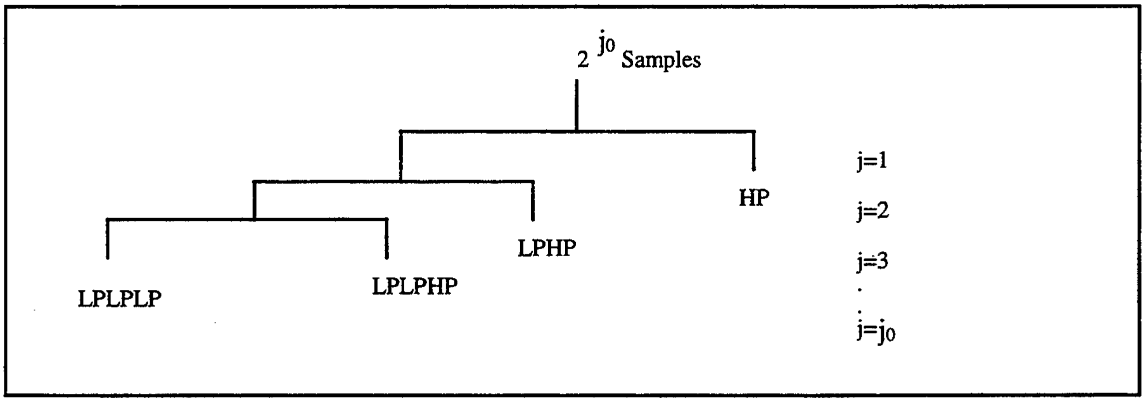

Figure 2.12: DWT tree structure.

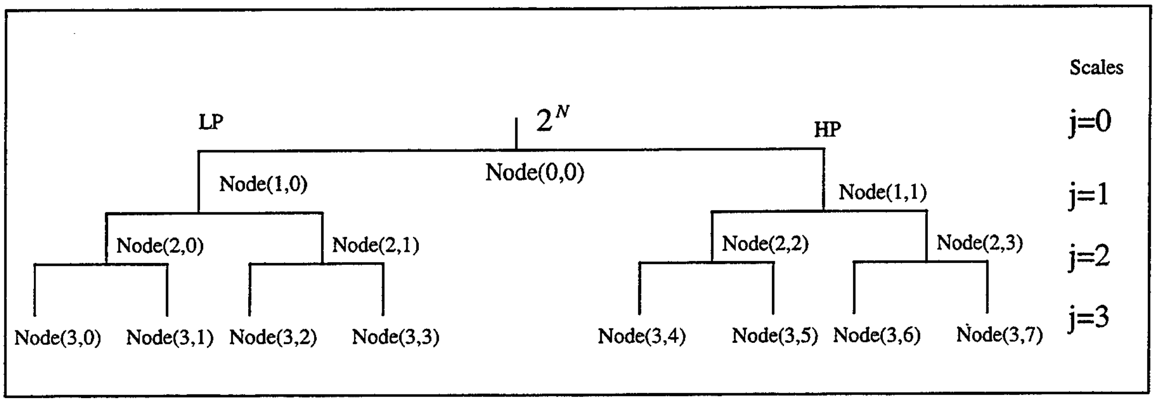

Figure 2.13: Complete wavelet packet decomposition tree structure.

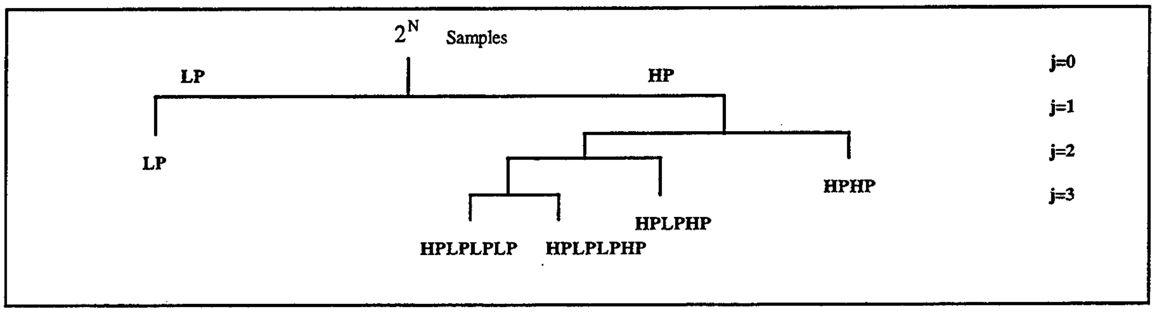

Figure 2.14: One possible wavelet packet decomposition scheme.

III. FEATURE EXTRACTION METHODS

In any classification task, extracting relevant features is key to good performance. Ideally, the extracted features should reveal some unique non-redundant characteristics that are most effective in discriminating between classes. This chapter presents two major methods for feature extraction. First, we consider the Local Discriminant Bases (LDB) scheme. It is designed to find the best distinguishing local basis in the wavelet packet decomposition (WPD) tree using a user-specified discriminant criterion [17]. Next, we investigate the Power Method which uses power values associated with the WPD nodes as features [21].

A. LOCAL DISCRIMINANT BASES METHOD

The Local Discriminant Bases (LDB) algorithm was originally proposed by Saito [17] in an effort to obtain a suitable basis in the WPD tree for feature extraction. It is similar in concept to the WP-based Best-Basis (BB) signal compression algorithm originally proposed by Wickerhauser [17,18] which selects a non-redundant wavelet basis from the entire WP decomposition. However, the LDB basis selection criterion is designed to extract a basis which best discriminates between signal classes, while the BB scheme identifies a basis which best compresses the information. Further details regarding the BB algorithm for compression applications may be found in [17, 26].

1. Discriminant Measures

Let us first briefly present the basis selection process involved in the BB scheme prior to discussing that of the LDB algorithm as they are conceptually related [17]. Both methods first expand a signal into a redundant library of orthogonal bases using the wavelet packet decomposition (or local trigonometric bases). A non-redundant basis which minimizes a user-defined information cost is then identified in the full WPD tree using the divide-and-conquer algorithm. In the case of the BB scheme, the user-defined selection criterion evaluates each node compression capability by its entropy. Recall that the Shannon entropy is commonly used as it measures the flatness of an energy distribution (few significant coefficients will be present at a given node when the entropy is low) [18]. Such a criterion is useful in signal compression applications where the goal is to represent the signal information using the least number of parameters. However, this selection criterion is not well matched to classification applications where the goal is to select the nodes that will best discriminate, i.e., will be most effective in showing the differences between various signal classes. So the main difference between the BB and the LDB scheme is in the choice of the selection criterion, as the identical divide-and-conquer scheme is then used in both cases to extract a non-redundant basis from the packet decomposition.

Various alternatives exist to choose discriminant measures. However, they all try to measure statistical distances among signal classes. Assume that $\mathrm{A}=\{a(i)\}_{i=1}^{M}$ and $\mathrm{B}=\{b(i)\}_{i=1}^{M}$ are two non-negative sequences normalized to one, so that $\sum_{i=1}^{M} a(i) = 1$ and $\sum_{i=1}^{M} b(i) = 1$. The discriminant measure function should measure how differently A and B are distributed. One of these functions is the relative entropy function (also known as the cross entropy, or the Kullback-Leibler Distance) and defined as:

$$I(\mathrm{A},\mathrm{B}) = \sum_{i=1}^{M} a(i) \log \frac{a(i)}{b(i)}, \tag{3.1}$$

with the convention log(0)= $-\infty$, log(x/0)= $+\infty$ [17].

Note that I(A ,B) is always greater or equal to zero as long as A and B are normalized to 1 [4]. However if A and B are not normalized to 1, the equation

$$I(\mathrm{A},\mathrm{B}) = \sum_{i=1}^{M} a(i) \left| \log \frac{a(i)}{b(i)} \right| \tag{3.2}$$

can be used instead of Equation 3.1 to avoid getting negative relative entropy values.

Note that I(A ,B)=0 if and only if A =B, while I(A ,B) gets large as A and B differ. Let us consider a simple example to illustratre this concept. Let three sequences $a(n)$, $b(n)$ and $c(n)$ be defined as:

$$a(n) = \left|\sin(0.5\pi n)\right|, \tag{3.3}$$

$$b(n) = \left|\sin(0.5\pi n)\right| + 2,$$

$$c(n) = \left|\sin(0.5\pi n)\right| + 10,$$

where n=0,1…31.

Since these three sequences are not normalized to 1, the relative entropy function, defined in Equation 3.2, is used to measure how much $a(n)$ differs from $b(n)$ and c(n), leading to:

$$I(a(n),b(n)) = 7.6339,$$

$$I(a(n),c(n)) = 16.6623.$$

It is clear that a(n) and $c(n)$ differ more than $a(n)$ and $b(n)$ do, due to the larger DC level present in $c(n)$. The only disadvantage of the relative entropy is the fact that it is not symmetric. However, symmetry is preferred in numerous applications [17]. Thus, the symmetric relative entropy function is defined as:

$$J(\mathrm{A},\mathrm{B})=I(\mathrm{A},\mathrm{B})+I(\mathrm{B},\mathrm{A})\,. \tag{3.4}$$

Other measures can also be used in the LDB method. Another possible measure is the norm-2 distance [17] defined as:

$$I(\mathrm{A},\mathrm{B})=\|A-B\|_2^2=\sum_{i=1}^{M}((a(i)-b(i))^2\,. \tag{3.5}$$

The efficiency of these distance measures varies as to the general behavior of the classification problem at hand. It is obvious that these two measures are defined for pair-wise comparisons, i.e., when there are only two classes. A different version of these measures must be employed with more than two classes. A potential solution relies on the pair-wise calculation of the distances defined as:

$$R=\sum_{i=1}^{C-1}\sum_{j=i+1}^{C}J(p^{(i)},p^{(j)}), \tag{3.6}$$

where $p^{(c)}$ is the sequence belonging to class c, and C is the number of classes considered.

2. Energy Maps and Relative Entropy Calculations

The first step in the LDB algorithm is the calculation of the time-frequency energy map of each signal class at hand [17]. Let $\{x_i^{(c)}\}_{i=1}^{N_c}$ be a set of training data signal vectors of length N that belongs to signal class c, and N_c be the number of signals in that signal class. The wavelet-based normalized energy map, E_c, obtained for this signal class is defined as:

$$E_c(j,k,l) = \frac{\sum_{i=1}^{N_C} (\Psi^T_{j,k,l} x_i^{(c)})^2}{\sum_{i=1}^{Nc} \left\| x_i^{(c)} \right\|^2}, \qquad (3.7)$$

where $\Psi_{j,k,l}$ is one of the basis functions associated with the node (*j,k*).

Recall that the scale number *j* corresponds to the depth of the tree decomposition and is defined in the range 0 to *J*, where $J \leq \log_2(N)$. The index *k* corresponds to a specific frequency band obtained at scale *j*, and is defined in the range 0 to $2^j - 1$. Finally, the index *l* corresponds to the time shift applied at the scale *j* and is defined in the range 0 to $N2^{-j} - 1$. Note that the normalization is crucial when the number of training signals in each signal class varies [17].

The second step in the LDB scheme is the computation of relative entropy values, denoted as *R*(*j,k*), associated with the node (*j,k*). For this purpose we apply Equation 3.6, which leads to:

$$R(\mathrm{j,k}) = \sum_{c1=1}^{C-1} \sum_{c2=i+1}^{C} J(\{E_{c1}(j,k,l)\}_{l=0}^{l=N2^{-j}-1}, \{E_{c2}(j,k,l)\}_{l=0}^{l=N2^{-j}-1}). \qquad (3.8)$$

Note that the relative entropy function defined in Equation 3.2 is used in the computation of Equation 3.8 as neither $\{E_{c1}(j,k,l)\}_{l=0}^{l=N2^{-j}-1}$ nor $\{E_{c2}(j,k,l)\}_{l=0}^{l=N2^{-j}-1}$ are normalized sequences.

3. The LDB Basis Selection Criterion

The last step in the LDB scheme is the selection of the "best" discriminating basis among all possible bases given by the WP decomposition. The selection criterion uses the relative entropy values obtained with Equation 3.8. Recall that this basis selection is very similar to that of the BB algorithm, with the exception that the BB method minimizes the total Shannon entropy, while the LDB maximizes the total relative entropy value to select the suitable basis. Thus, the LDB selection method can be summarized as follows:

1- Calculate the relative entropy values associated with each node according to Equation 3.7.

2- Compare the relative entropy value of a parent node to that of the sum of its two children nodes by starting from the bottom of the decomposition tree and marking the one or ones with the largest relative entropy.

3- Combine the highest relative entropy marked nodes into a basis, by starting from the top of the decomposition tree.

An example will be given next, to clarify the LDB selection method. Consider Figure 3.1, which shows relative entropy values obtained at each node for a given WP decomposition. The LDB scheme compares relative entropy values of each parent and children nodes. The higher relative entropy value obtained between a parent and its children is assigned to the parent node, as shown between parentheses in Figure 3.2. Next, the parent or the children nodes with a higher relative entropy value is marked with an asterisk, as illustrated in Figure 3.2. Finally, the topmost nodes marked with asterisks form the selected LDB that will be used in the feature extraction task, as shown in Figure 3.3.

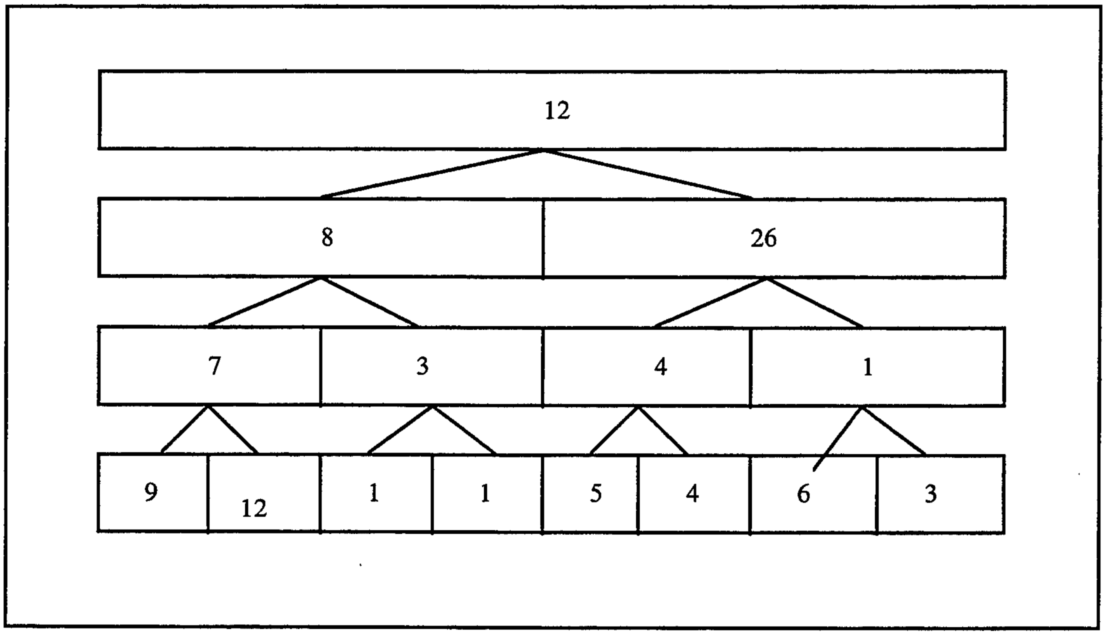

Figure 3.1: WPD tree with relative entropy values assigned to each node.

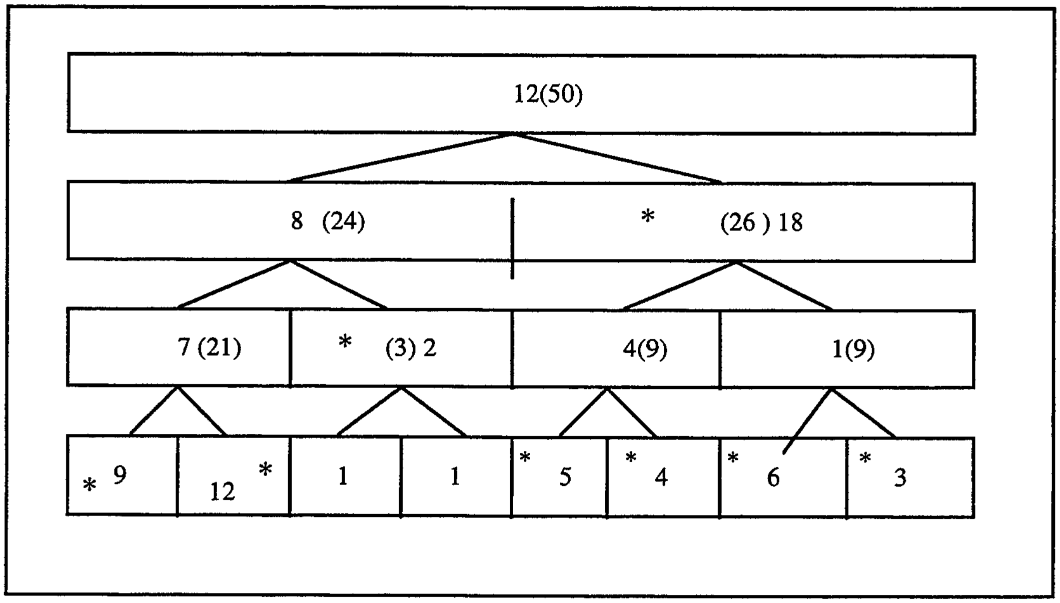

Figure 3.2: Result of step 2 in the LDB algorithm.

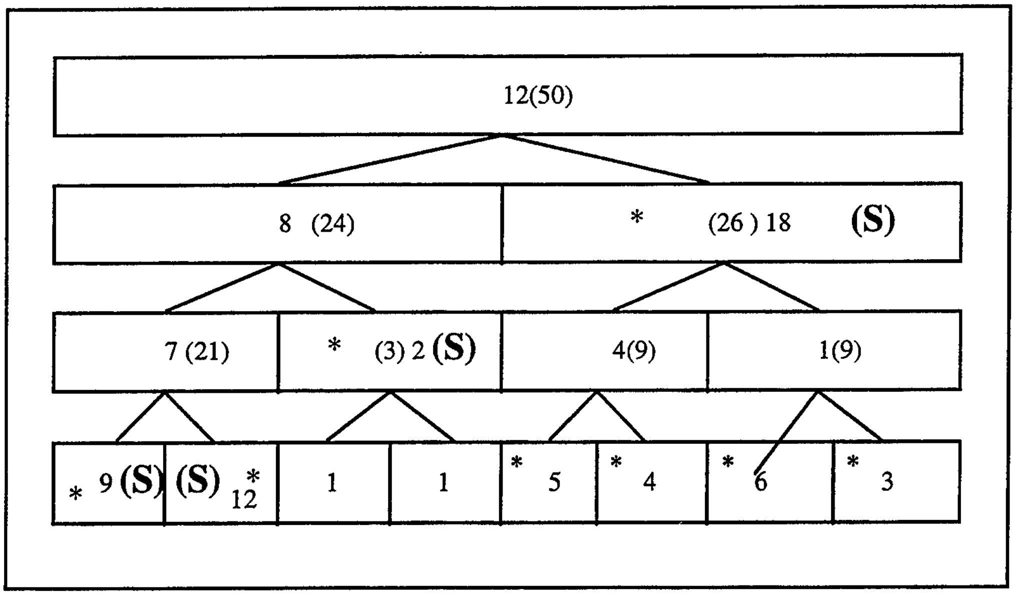

Figure 3.3: Selected nodes (Marked with **(S)**).

Recall that the basis selection criterion is guaranteed to select a complete basis set out of possible $2^{2^{(J-1)}}$ bases, where $J \leq \log_2(N)$, covering the full frequency axis. Further details regarding the LDB scheme may be found in Saito [17].

4. Feature Selection

Once the LDB decomposition is computed, the selected basis is applied to decompose the various signals belonging to known classes, and the resulting wavelet coefficients are used as feature parameter for inputs to train a classifier. However a large number of feature parameters exponentially increases the amount of data needed to train and validate the classifier. This problem, originally stated by Bellman, is known as the "curse of dimensionality," and will be considered further in later chapters [20]. Thus, it is usually preferable to use only a small subset of the WP nodes selected by the LDB scheme. Selecting these specific coefficients is a difficult problem. Most schemes available are somewhat problem dependent and a general methodology that can be applied to every problem is still an area of open research. The feature selection method used in the LDB scheme relies on the relative entropy values of the basis functions $\Psi_{j,k,l}$ in the selected LDB decomposition, which is denoted as $R(j,k,l)$ and can be computed using

$$R(j,k,l)=\sum_{c1=1}^{C-1}\sum_{c2=i+1}^{C}J(\{E_{c1}(j,k,l)\},\{E_{c2}(j,k,l)\}). \tag{3.9}$$

Note that if the signal length is N then the total number of basis functions $\Psi_{j,k,l}$ in the selected LDB will be N as well, and every basis function will produce one coefficient for a given signal. The basis functions $\Psi_{j,k,l}$ are ordered as to their relative entropy values $R(j,k,l)$, and $K<N$ of them are selected to be used in the feature extraction task.

5. Examples

Let us consider two examples to illustrate the capabilities of the LDB method in extracting features. The first example consists of two signal classes while the second one consists of four signal classes.

a) Two Signals Class Example

We first consider linear and quadratic chirp signals of length N=32=2^5. Variations in each signal are obtained by introducing some small random variation in the chirp specific frequencies. Thus, the general expression for signals belonging to the linear chirp class is given by:

$$x_L(n) = \sin(\frac{\pi(0.35 + \Delta f)n^2}{32}), \tag{3.10}$$

where Δf is a uniform random variable U[0,0.1] and n=0,1,....., 31.

The general expression for the signals belonging to the quadratic chirp class is given by:

$$x_Q(n) = \sin(\frac{\pi(0.217 + \Delta f)n^3}{1024}), \tag{3.11}$$

where Δf is a uniform random variable U[0,0.07] and n=0,1,....., 31.

Figure 3.4 shows the time domain representations obtained for noise-free linear and quadratic chirp signals. Finally, additive white Gaussian noise with SNR=-5 dB is added to each signal. The LDB scheme was implemented using 40 signal trials for each class. Five noisy sample waveforms for each signal class are shown in Figure 3.5. The LDB decomposition algorithm leads to the selection of nodes (2,0), (2,1), (3,4), (3,5), (5,24), (5,25), (5,26), (5,27) and (3,7). Relative entropy values obtained at each of these nodes are given in Table 3.1. The resulting frequency partitioning obtained with the LDB scheme is shown in Figure 3.6. Note that, this time-frequency tiling shows the 32 selected bases covers the whole frequency axis. However as mentioned earlier, using only a subset of the features selected is better for the classification step. The feature subset was selected by ordering the basis functions by decreasing entropy values, and choosing the top five. These five top suitable basis functions selected belong to the two top nodes in the tree, nodes (2,0) and (2,1), and are shown in Figure 3.7.

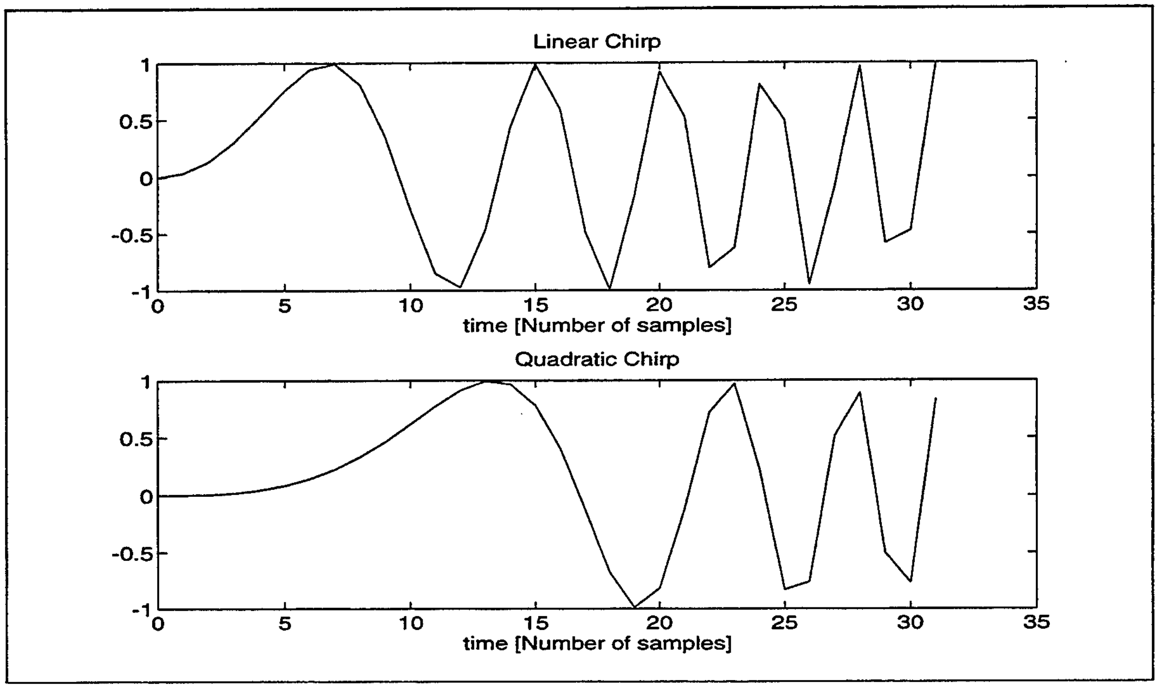

Figure 3.4: Time domain representations of linear and quadratic chirp signals.

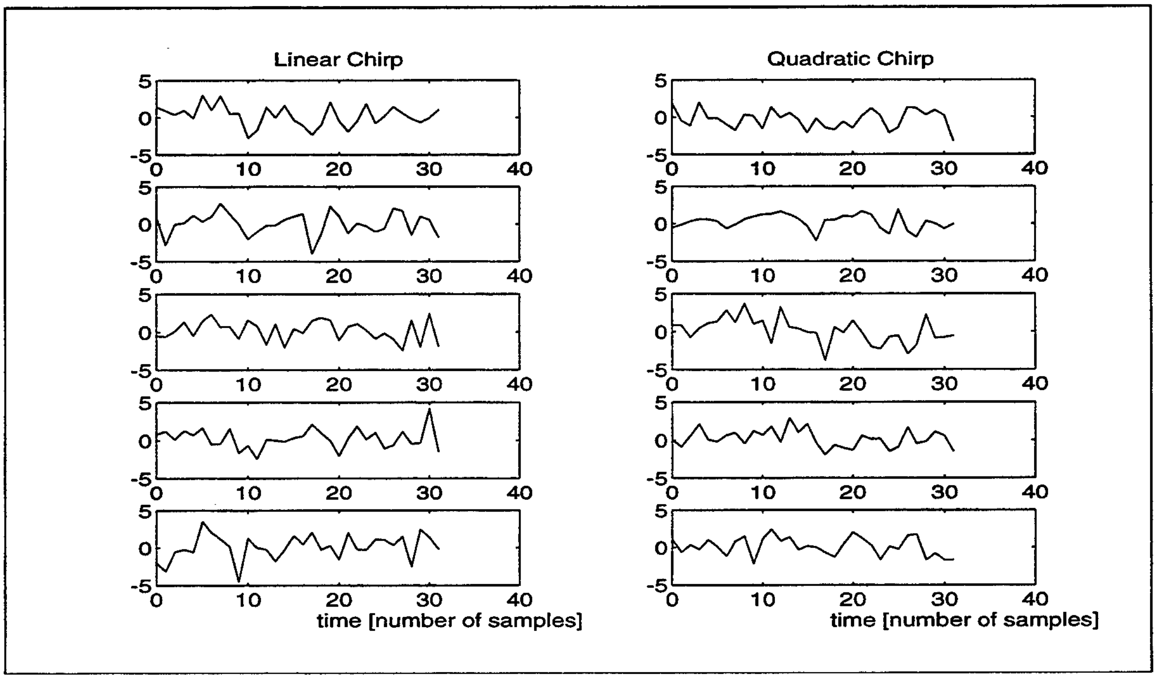

Figure 3.5: Five noisy trials for linear and quadratic chirp signals, used in example 5a; SNR=-5dB.

Node number	(2,0)	(2,1)	(3,4)	(3,5)	(5,24)	(5,25)	(5,26)	(5,27)	(3,7)	Total R.E.
Relative Entropy (R.E.)	**0.2**	**0.12**	**0.07**	**0.03**	**0.014**	**0.005**	**0.001**	**0.02**	**0.12**	**1.064**

Table 3.1: Relative entropy values obtained at the LDB selected nodes for example 5a.

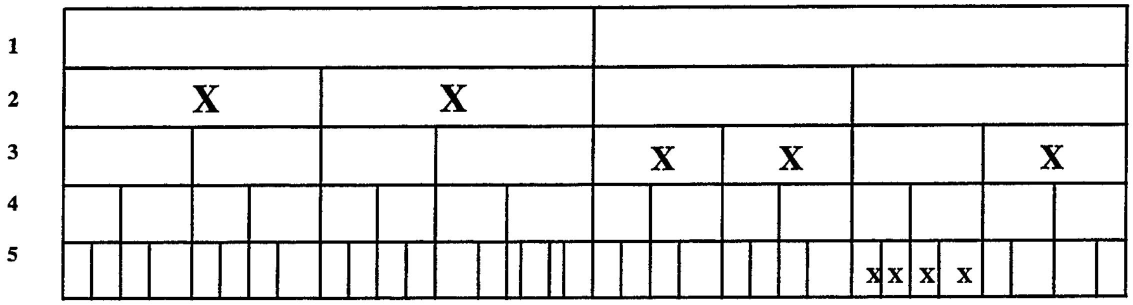

Figure 3.6: LDB time-frequency partitioning for example 5a. Selected frequency bands are indicated with "X".

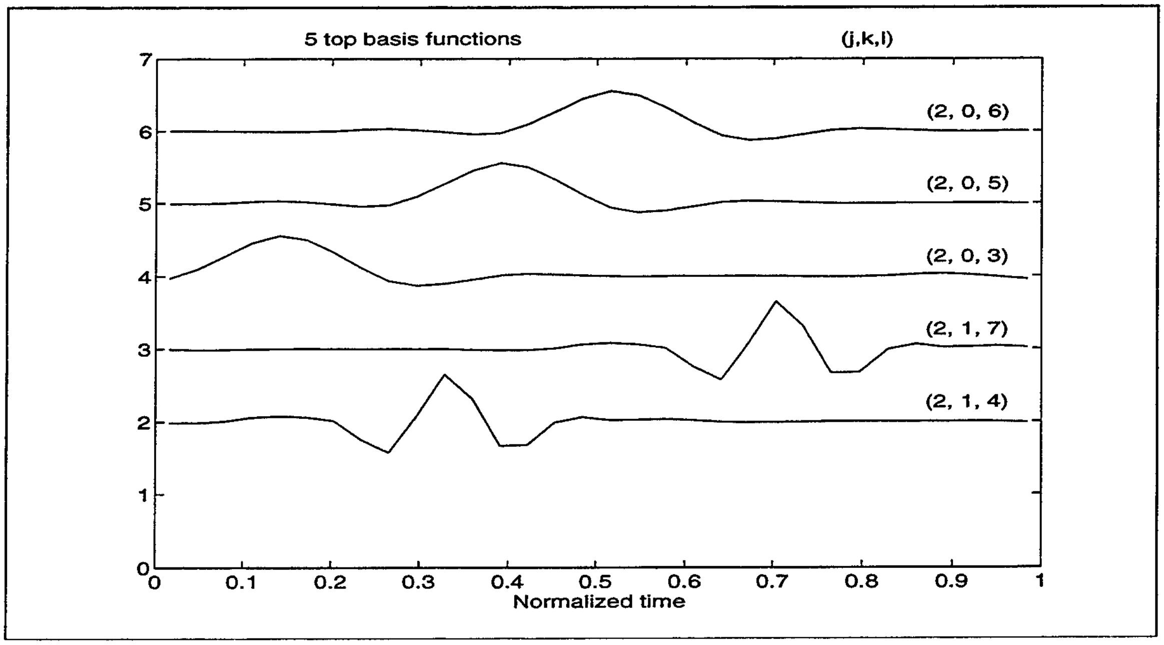

Figure 3.7: Top 5 LDB basis functions obtained for example 5a.

b) Four Signals Class Example

Next, four signal classes are considered: linear and quadratic chirps (already used in the previous example), and low and high frequency sinusoidal signals. The signal length is kept at N=32=2^5 samples. As in the previous example, a random variability in the high and low sinusoidal signals is introduced by adding a random component to the signal carrier frequency. Thus, the high and low frequency sine expressions are given by:

$$x_H(n) = \sin(\pi n(0.7 + \Delta f_1)), \quad x_L(n) = \sin(\pi n(0.2 + \Delta f_2), \qquad (3.12)$$

where Δf_1 and Δf_2 are uniform random variables defined as U[0,0.2] and n=0,1,.....,31.

Figure 3.8 shows the time domain representations obtained for noise-free high and low frequency sine signals. Additive white Gaussian noise is also added to the sequences with resulting SNR=-5 dB. Five noisy high frequency sine and low frequency sine signal trials are shown in Figure 3.9. Forty training signals per signal class are used in the experiment, corresponding to a total of 160 signals. The LDB scheme leads to a frequency partitioning based on the following selected nodes: (2,0), (3,2), (3,3), (3,4), (3,5), (4,12), (5,26), (5,27), (5,28), (5,29) and (4,15). The frequency partitioning obtained is shown in Figure 3.10. Table 3.2 lists the relative entropy values obtained for the selected nodes. Once again it is obvious that these selected nodes cover the entire frequency axis. At this point the 32 basis functions were ordered in decreasing relative entropy values, and the top five selected to be used in a feature extraction task. The top 5 basis functions are plotted in Figure 3.11.

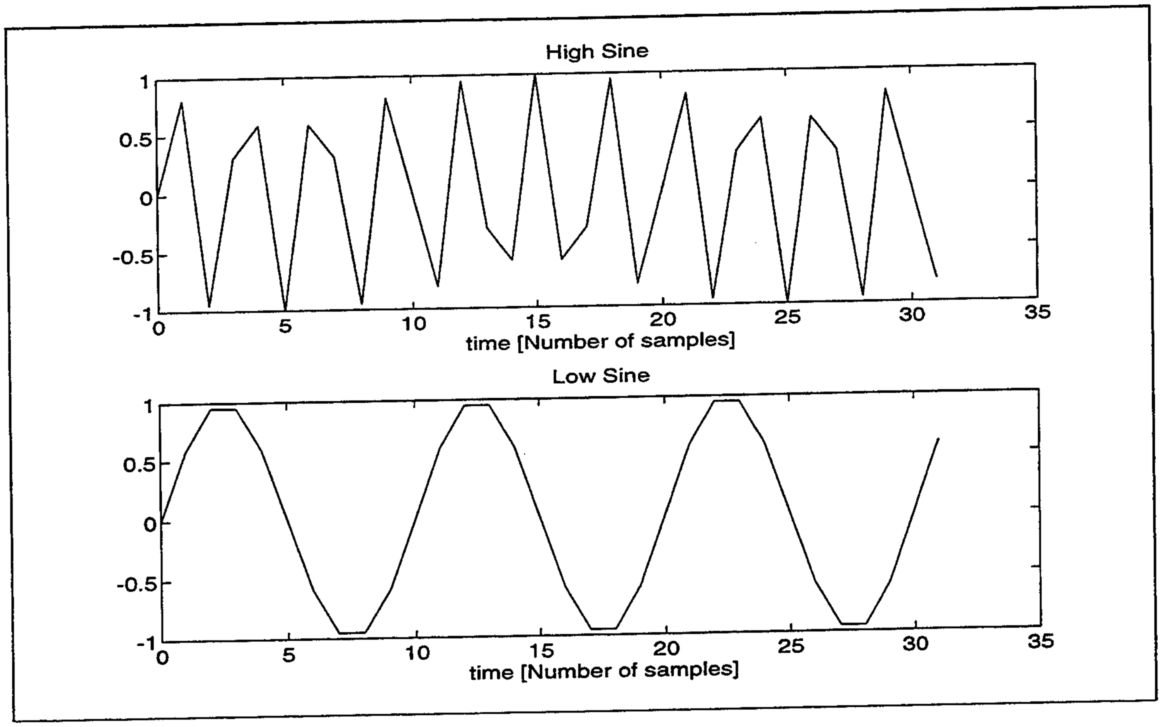

Figure 3.8: Time domain representations of high and low frequency sine signals.

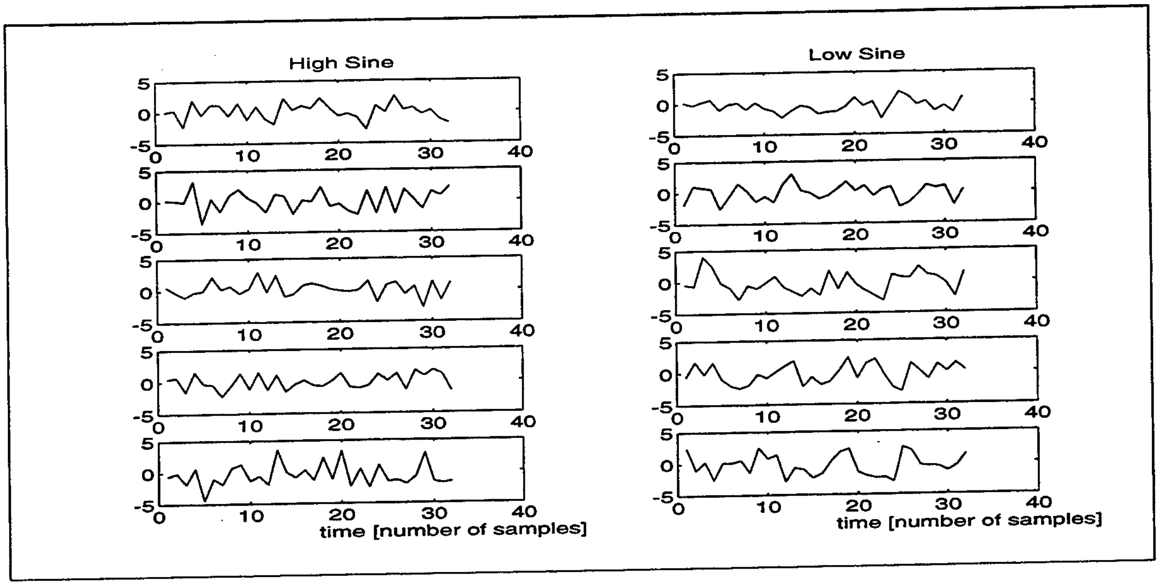

Figure 3.9: Five noisy trials for high and low frequency sine signals, used in example 5b; SNR=-5dB.

Node Numbers	(4,15)	(5,29)	(5,28)	(5,27)	(5,26)	(4,12)	(3,5)	(3,4)	(3,3)	(3,2)	(2,0)	Total R.E.
Relative Entropy (R.E.)	**0.07**	**0.05**	**0.1**	**0.07**	**0.06**	**0.31**	**0.31**	**0.27**	**0.28**	**0.46**	**0.7**	**2.7**

Table 3.2: Relative entropy values obtained at the LDB selected nodes for example 5b.

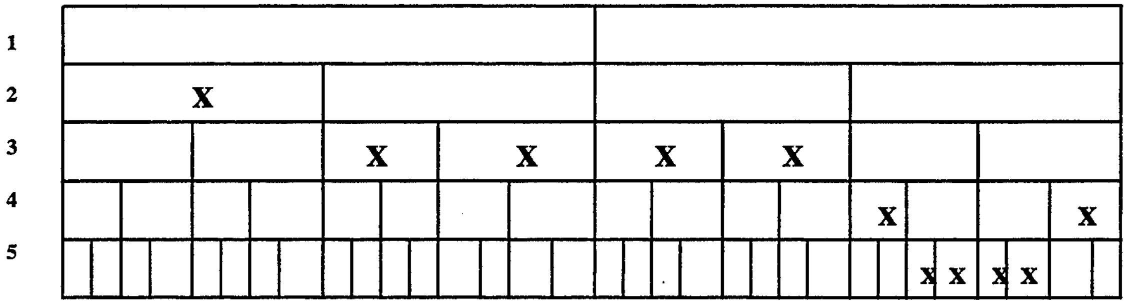

Figure 3.10: LDB time-frequency partitioning for example 5b. Selected frequency bands are indicated with "X".

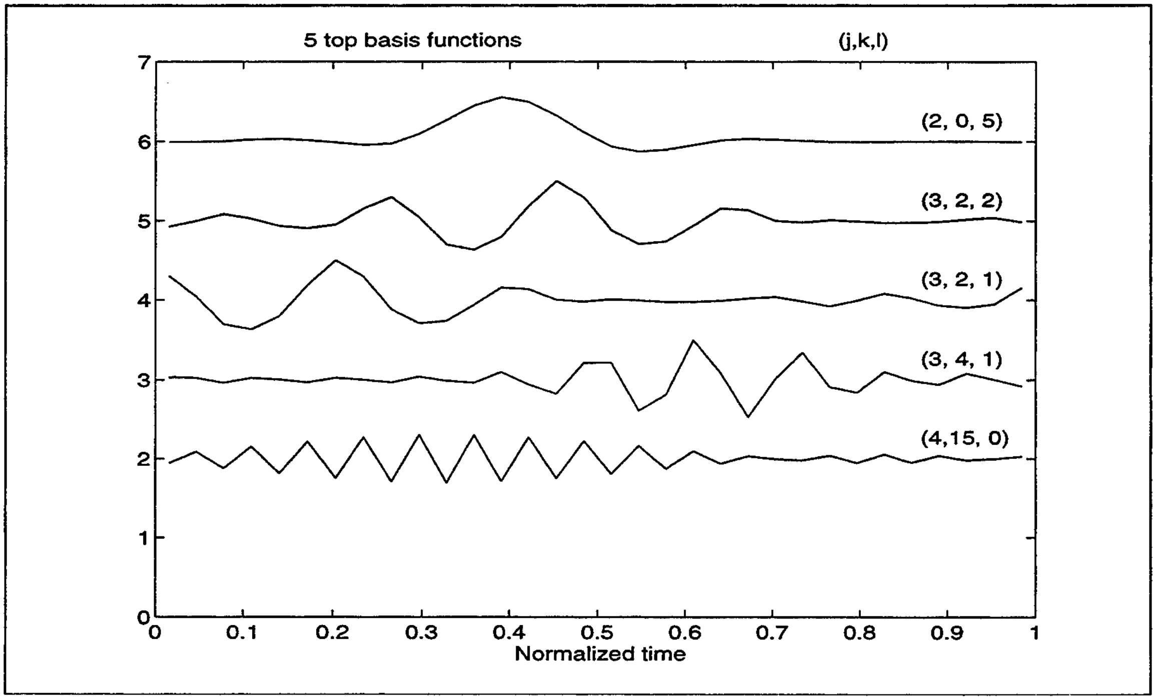

Figure 3.11: Top 5 LDB basis functions obtained for example 5b.

6. LDB Algorithm Drawbacks

a) Large Class Size

The first problem observed in the LDB based feature extraction tasks is the probable loss of performance when the number of signal classes is more than two. There may be two reasons to this particular problem: 1) a basis that discriminates all the signal classes is not guaranteed to exist and 2) even though such a basis may exist, the LDB scheme is not guaranteed to find it. Obviously, there is nothing that one can do in the first case. However, there may be hope in the second situation. Recall that the cost function used for measuring dissimilarities between classes averages pair-wise distances between the classes considered. For example, when dealing with three classes, the total relative entropy *R(j,k)* expression obtained at node (j,k) becomes:

$$R(j,k)=\sum_{c1=1}^{2}\ \sum_{c2=i+1}^{3} J(\{E_{c1}(j,k,l)\}_{l=0}^{l=N2^{-j}-1},\{E_{c2}(j,k,l)\}_{l=0}^{l=N2^{-j}-1})\,. \qquad (3.13)$$

Thus, *R(j,k)* averages out the 3 pair-wise relative entropy values obtained between signal classes 1, 2, and 3. Assume that one of the nodes is very good at discriminating signal classes 1 and 2 (that is, the relative entropy for these two signal classes is high), but the same node is poor at discriminating signal classes 1 and 3 (that is, the relative entropy value for these two signal classes is low). However, this specific node may still be selected, because the double summation present in the total entropy formula in Equation 3.13 averages out the pair-wise contributions. A potential "better discriminating" node might be one which is not as good in discriminating signal classes 1 and 2, but which is better in discriminating 1 and 3, that is, this node may be equally good in discriminating these three signal classes.

We further illustrate this problem with the simple 4 signal classes example considered earlier in Section 5b. Recall that 11 nodes were selected in the LDB selection implementation step. Pairwise entropy values obtained at the selected node (2,0), plotted in Figure 3.12, reveal the problem. Recall that pair-wise entropy calculations include relative entropies obtained for class pairs {1,2}, {1,3}, {1,4}, {2,3}, {2,4} and {3,4}. Figure 3.12 shows that the relative entropy value obtained between the first and second signal classes (equal to 0.25) is significantly higher than that obtained for the other pair-wise class computations, meaning the features found using this node will be very effective in differentiating between classes 1 and 2. However, it also shows that node (2,0) is not good at discriminating between classes 3 and 4, as the relative entropy between third and fourth signal classes is quite a bit lower than those obtained with the other pair-wise computations. Note that information regarding this variability in the quality of the discrimination capability will be lost after averaging all pair-wise contributions.

Another example of this particular problem is visible in Figure 3.13 which plots relative entropy values obtained at the parent node (2,1), and the sum of entropy values obtained at its two children nodes (3,2) and (3,3). Figure 3.13b shows that the children nodes are quite good at discriminating signal classes {2,4} and {3,4}, but poorer at discriminating signal classes {2,3}, due to the variations in the corresponding relative entropy values obtained for the given pair-wise comparisons. Figure 3.13a shows that the magnitudes of the entropy values obtained at the node (2,1) do not vary as much, meaning that the features selected for this node will be "equally good" to discriminate between all signal pairs. Therefore, given that the magnitude of the relative entropy values obtained at the parent or the sum of its children are similar, there is the likelihood that node (2,1) would be better suited to discriminate between the various classes than the children nodes are, where more variability in discriminant quality is visible. However, recall that the cost function designed to choose between parent and children node averages out all pair-wise contributions, and thus, disregards the effects due to unbalanced pair-wise contributions. In the example shown in Figure 3.13, the LDB selection algorithm selects the children nodes (3,2) and (3,3) as their total relative entropy nodes is higher than that of the parent node.

One possible solution to this problem may be to avoid averaging out the various pair-wise contributions altogether, as is done with the original LDB selection scheme in Equation 3.6. A proposed alternative might be to take into consideration the consistency with which the features obtained at a given node are at discriminating between class pairs. Thus, ideally, the pair-wise relative entropy values obtained at a given node should be: 1) high and 2) similar in magnitude, meaning the features are equally good to discriminate between all classes taken pair-wise. Thus, the distribution of relative entropy values obtained at a given node should be as high and flat as possible. One possible candidate to measure the flatness of a data distribution is the Shannon Entropy (SE) function defined as [18]:

$$Q(x)=\sum_i |x_i|^2 \log\frac{1}{|x_i|^2} . \qquad (3.14)$$

The SE function has a high value when the data sequence x has a flat distribution. The new node selection method can be described briefly as:

Step 1- Energy maps of each signal classes are calculated, as in the original LDB selection method.

Step 2- Pair-wise relative entropy values are computed at each node, and the Shannon entropy function computed to evaluate the flatness of the distribution of pair-wise contributions. The resulting SE value is placed at each node.

Step 3- Use the divide-and-conquer algorithm to find the best non-redundant basis which maximize the total SE value.

Now let's apply this new SE selection criterion to the previous node selection problem described in Example 5b, and Figure 3.13. Recall that even though node (2,1) looked better than its children nodes in discriminating between class pairs overall, the original LDB selection algorithm chose the children nodes due to the higher sum of their relative entropy value. Table 3.3 lists the Shannon entropy values obtained at the parent node (2,1) and its two children nodes (3,2) and (3,3). Results show that node (2,1) has a higher SE value then that obtained by summing the contributions obtained by its two children nodes. Therefore, node (2,1) would have been chosen. Unfortunately this algorithm also has its own problems because it doesn't take into account the actual magnitudes of the pair-wise relative entropies. Thus, a flat relative entropy distribution may be selected even though the values are all quite small, meaning the features obtained at that node are equally bad in discriminating two classes at a time. Thus, we would need to define some type of lower bound on the pair-wise relative entropy values and take it into account in addition to comparing parent and children Shannon entropy values. However, at this time a successful combination of these two criteria which improves the classification rate has not been isolated.

Finally, the user needs to keep in mind that judging a selected LDB basis on the behavior of only a single node may also be misleading. A selected node may be quite poor at discriminating two signal classes, but the other selected nodes may compensate for this problem to a certain extent.

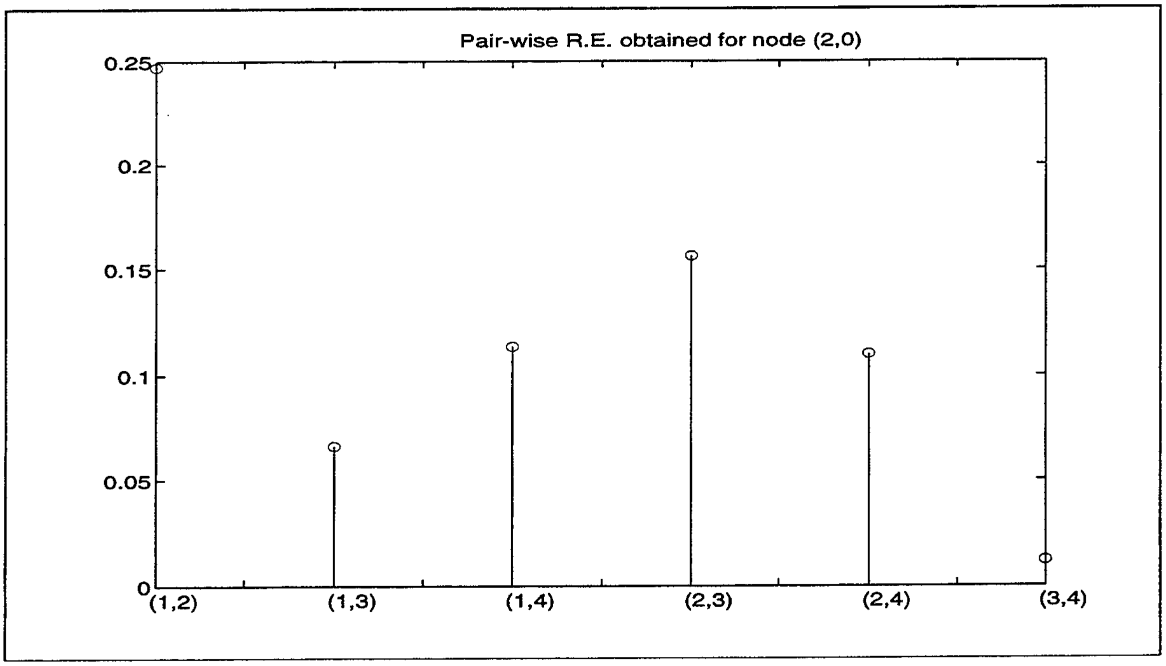

Figure 3.12: Pair-wise relative entropy values of node (2,0).

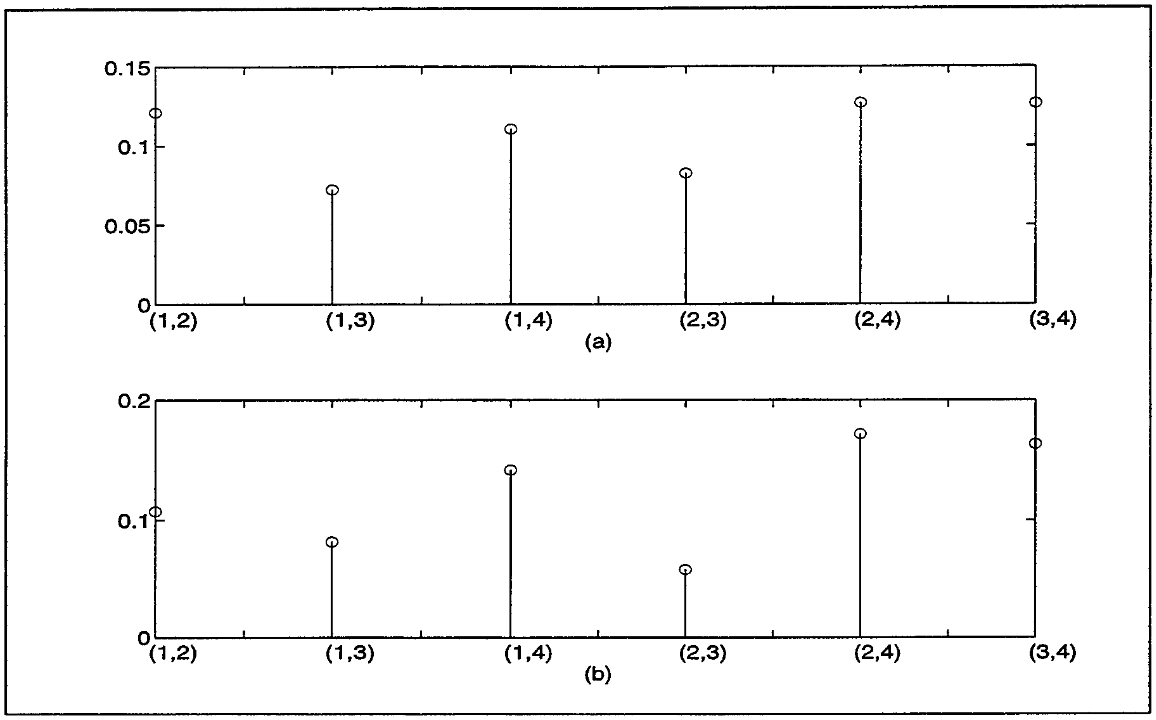

Figure 3.13: (a) Relative entropies of node (2,1), (b) Total Relative entropies of node (3,2) and (3,3).

Node Numbers	(3,2)	(3,3)	Total	(2,1)
Shannon Entropy.	**0.089**	**0.036**	**0.125**	**0.134**

Table 3.3: Shannon Entropy values of various nodes.

b) Dimension Reduction Problems

Another potential problem of the LDB algorithm comes from the dimension reduction process. Note that using all the basis functions obtained from the LDB scheme to define input features to a classifier may result in a large feature set, and make the classification task more difficult. As mentioned earlier, the number of features should be kept as small as possible. Therefore, one needs to select a subset of the LDB basis functions which contain only the most relevant information. The selection process originally proposed by Saito selects the first K basis functions with the highest relative entropy values [17]. However as discussed earlier, when the number of classes is high (say larger than 8 or 9), the averaging process present in the computation of the relative entropy values, may lead to the selection of nodes which are poor in discriminating some of the classes pair-wise. In addition, selecting only a subset of the basis functions may worsen the classification performance by preventing some of the non-selected basis functions to compensate for selected ones with poor isolated pair-wise discrimination capabilities.

Figure 3.14 illustrates this problem by plotting the pair-wise relative entropy values obtained for the top five basis functions, determined by the original LDB scheme, in Example 5b, note that x axis labels 1, 2, 3, 4, 5 and 6 correspond to pair-wise signal classes {1,2},{1,3},{1,4},{2,3},{2,4} and {3,4} successively. It is clear that most of the basis functions have difficulties in discriminating classes 1 and 3, that is, the features obtained using these specific basis functions will be very similar for the first and third signal classes. As a result, this poor discrimination quality will make the classification task more difficult to differentiate between classes 1 and 3.

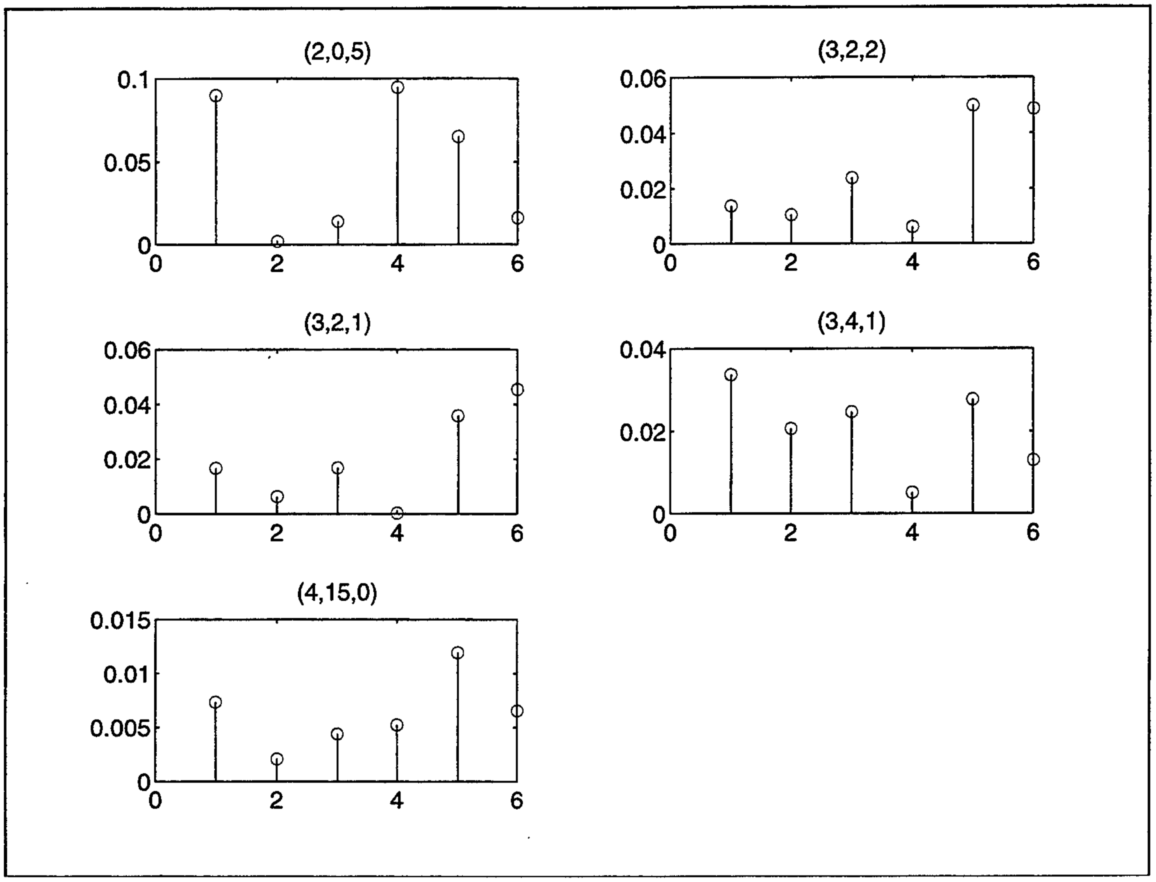

Figure 3.14: Pair-wise relative entropy values of some selected basis functions in example 5b. Basis function numbers are at the top of each plot.

c) Synchronization Issues

Another problem with the LDB comes from the fact that wavelet packet decomposition is shift-variant. As a result, a slight time shift in the signal may cause drastic changes in the wavelet coefficients. Saito addressed this issue by introducing cycle spinning [25], and showed some improvements in the classification performances. Further details may be found at the end of Chapter III.

7. Summary

In this section, we showed that the cost function used to measure dissimilarities between classes in the original LDB basis selection process becomes less and less meaningful as the number of classes increases, and illustrated what some of the main resulting problems are with a few basic examples. In addition, we showed that this cost function may further impair the extraction of a small set of relevant features, and ultimately worsen the classification performances. We proposed an alternate basis selection criterion based on measuring the flatness of the pair-wise relative entropy value distribution and showed that it could potentially improve results, if used in combination with a lower bound on the Shannon entropy function.

B. POWER METHOD

As mentioned earlier, one of the main drawbacks with the LDB scheme is the fact that it requires some time-domain signal synchronization prior to the decomposition step to insure that slightly shifted signals belonging to the same class will be categorized as such. Some additional robustness in the classification process (with respect to the time shifting issue) can be obtained by considering the average energy obtained at each node, instead of the individual wavelet coefficients. In this section we consider two such energy-based approaches after defining the node-based features considered.

1. Energy Maps And Feature Extraction

Consider a vector $\underline{x}$ of length N. The average energy (i.e., power) contained in $\underline{x}$ is given as:

$$P_x = \frac{1}{N} \underline{x}^T \underline{x} \; . \tag{3.15}$$

The full WPD of the signal $\underline{x}$ of length N leads to a spectral partitioning with a total number of nodes (i.e., frequency bands) $TNF = \sum_{j=0}^{\log_2(N)} 2^j$. The power contained in each frequency band (i.e., at each node) can then be computed using Equation 3.15. The power obtained at the node (d,b) in the WPD tree will be denoted as p(d,b), which leads to the power map of a given signal, as illustrated in Figure 3.15.

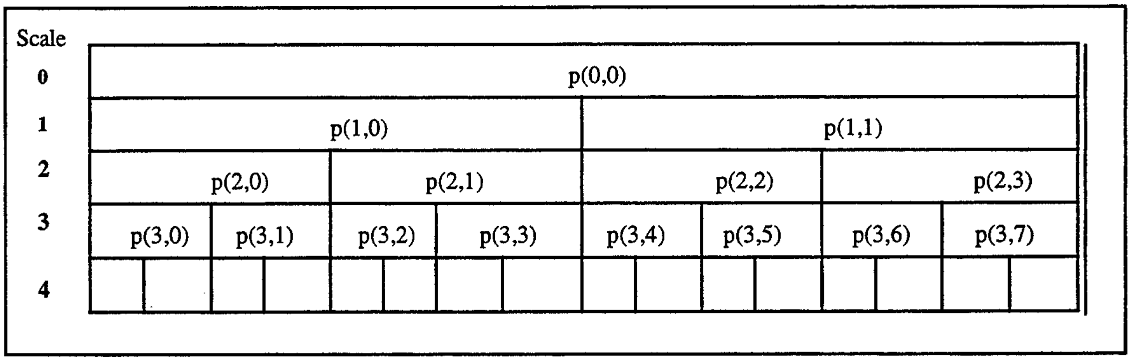

Figure 3.15: Power values associated with each node of the WPD.

Let us consider the following example to illustrate the power mapping concept. Assume that a given signal $x(n)$ is defined as:

$$x(n) = \sin(0.1\pi n), \;\; n = 0,1,2,.....255. \tag{3.16}$$

The WPD decomposition leads to the power map shown in Figure 3.16. Note that, high energy bins are represented in dark colored areas while light areas represent the low energy bins. Recall that average energy values show the power of a given signal in the frequency range of the associated node. Thus, the power centered around the frequency

0.1 Hz is more highly focused at high scales, which correspond to narrower filter bands. The basic idea behind the power method is to consider the average energy values as features. Note that there will be 2^j number of average energy values at the scale j due to the WPD tree structure. For example, p(0,0) represents the first average energy feature. As scale 0 represents the time domain signal, p(0,0) is also the power of the given signal.

Note however that this feature set contains much redundant information, as a parent node and its two children nodes carry the same frequency information. Thus, selecting only non-redundant information becomes essential when reducing the number of feature parameters. Four feature selection schemes are considered next.

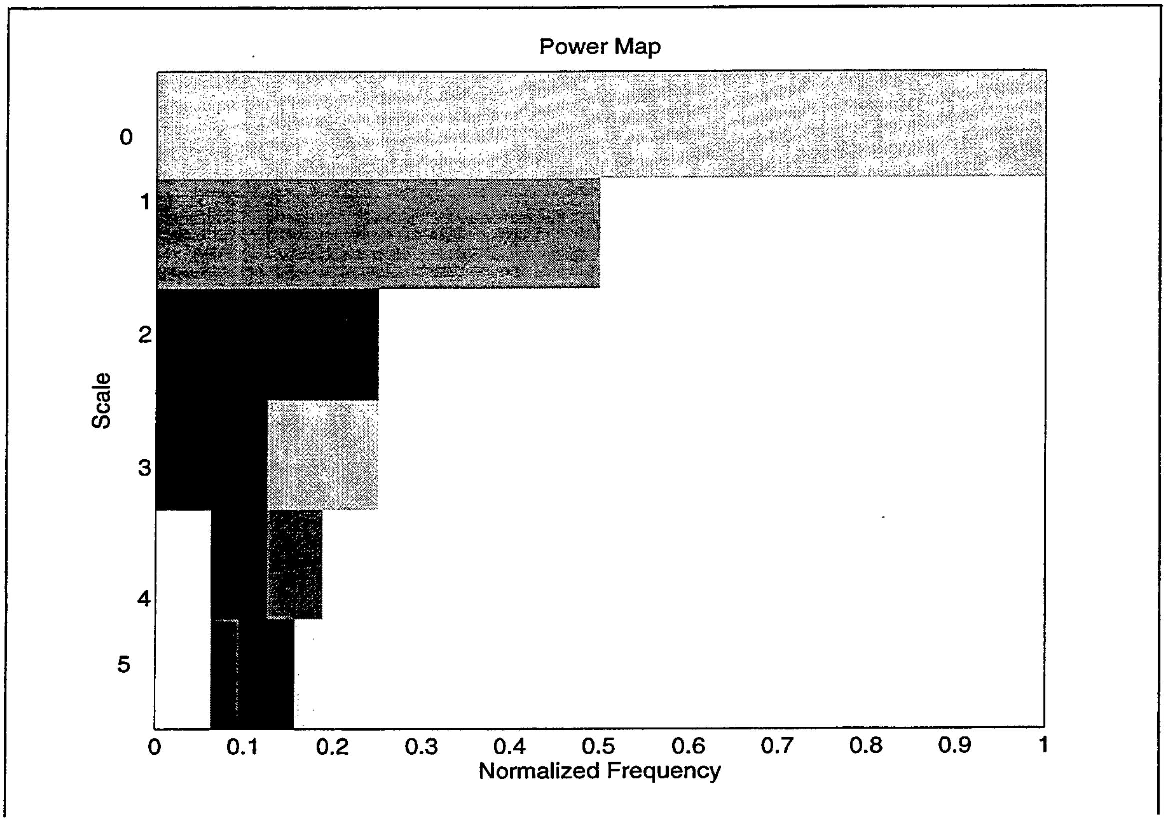

Figure 3.16: Power map of a sine signal.

2. Feature Selection

Four feature selection approaches are considered to reduce the dimensionality of the classifier. The first one, originally proposed by Learned and Willsky [21], uses the SVD information obtained from the power mapping, the second one selects the most within-a-class consistent features, the third one selects the most-discriminating features and the last one uses the same nodes that the LDB feature extraction method selects for feature extraction task.

First, following Learned and Willsky [21], let us define the "power matrix" which will be used in the derivations. Consider the vector $\underline{p}(n,t)$ which contains all power values of the *n*th signal in the signal class *t*. The power matrix P_t of the signal class *t* is given as [21]:

$$P_t = \left[\underline{p}(1,t), \underline{p}(2,t), \underline{p}(3,t), \ldots\ldots\ldots\ldots \underline{p}(N_t,t)\right], \tag{3.17}$$

where N_t is the number of training signals in the signal class *t*.

a) Learned and Willsky's (LW) Feature Extraction Scheme

Learned and Willsky's scheme is a two-step process that searches for the dominant power nodes that lead to the "best" separation between classes.

(1) Dominant singular vector identification. The first step identifies the dominant left singular vector (i.e., associated with the largest singular value) of each single class power matrix P_t. Once identified, the dominant singular vector is used to represent the signal class *t*. Note that a large gap between the first two

singular values indicates that the dominant singular value might be sufficient in representing the information contained in the class. Thus, Learned and Willsky investigated the presence of a dominant singular vector by evaluating the following singular value difference ratio:

$$\Delta\sigma_t = \frac{\sigma_{t,1} - \sigma_{t,2}}{\sigma_{t,1}}, \tag{3.18}$$

where $\sigma_{t,1}$, and $\sigma_{t,2}$ are respectively the largest and second largest singular values of the matrix P_t defined from the signal class *t*. The ratio $\Delta\sigma_t$ is defined between 0 and 1, and a value close to 1 denotes the existence of a dominant singular value and associated singular vector.

(2) Node Selection. Once each class dominant singular vector is identified, Learned and Willsky proposed to identify the nodes that contain significant information for a given class by selecting the components of the dominant singular vector that are within a given percentage of the maximum component. Thus, "significant node" locations are obtained by selecting those corresponding to the singular vector coordinates which lie within 20% of the maximum singular vector component. The full feature set is then obtained by combining significant feature sets obtained for all given classes.

Finally, non-redundant information is kept by insuring that the full feature set does not contain both parent and children nodes. When such a case occurs, Learned and Willsky chose to keep only the parent nodes, and disregarded any children nodes. This decision insures that the size of the full feature set is kept small, while it provides good inter-class separation.

(3) Example. We apply the Learned and Willsky scheme to a two-signal class problem next. The signals are obtained from the Wavelab.700 package [7]. The first signal (called "MishMash" in Wavelab.700) is a combination of high frequency sine, linear, and quadratic chirps and defined as:

$$x(n) = \sin(\frac{\pi n^3}{3N^2}) + \sin(0.6902\pi n) + \sin(\frac{\pi n^2}{8N}), \qquad (3.19)$$

where N is the length of the signal and $n=1,2,3,\ldots\ldots N$.

The second signal (called "Doppler") is defined as:

$$x(n) = \sqrt{\frac{n}{N}(1-\frac{n}{N})}\sin\left[\frac{2.1\pi}{\frac{n}{N}+0.05}\right], \qquad (3.20)$$

where N is the length of the signal and $n=1,2,3,\ldots\ldots N$.

Figure 3.17 plots the signals. A signal length equal to 32 ($=2^5$) was considered. Additive white gaussian noise was added to get an SNR=10dB. Forty training signals were created per class. Figure 3.18 plots 5 trials of each resulting noisy signal. Note that, the size of the power matrix for each signal class is 63x40, as the total number of nodes is 63 ($\sum_{j=0}^{\log_2(32)} 2^j$) for a signal length of 32, and there are 40 training signals per class. The SVD of the power matrix obtained for each class is computed next. Figure 3.19 plots the power map obtained for one sample trial in each signal class. The singular values obtained for each signal class are shown in Figure 3.20. Next, difference ratios $\Delta\sigma_t$ were calculated using Equation 3.18 to evaluate whether one can find a dominant singular vector for each class. The difference ratios for the "Mishmash" and "Doppler" signal classes are computed using Equation 3.18 and found respectively equal to be 0.87 and 0.84. Note that these relatively high values indicate that one may be sure about the existence of a "true" dominant singular vector. Figure 3.21 plots the dominant singular vectors obtained for the two signal classes.

Next, the significant nodes for each signal class are found by selecting the singular vector coordinates which are within 20% of the maximum singular vector coordinate value. Figure 3.21 shows that values at indexes 35 and 38 are significant for the Doppler class, and the value at index 53 is significant for the Mishmash class. These indexes respectively correspond to nodes (5,3), (5,6) and (5,21) in the WPD tree. The corresponding node locations in the WPD tree are shown in Figure 3.22. Therefore, the full feature set is selected as the combination of both class-specific features, and shown in Figure 3.22. Note that there is no redundancy of information contained in the feature set, as it doesn't contain both parents and children. Thus, all three nodes will be used as features parameters for inputs to a classifier.

Figure 3.23 plots the training data feature set in 3D and Figure 3.24 shows its three projections onto the three main planes. In general, good classification performance is expected when the feature sets are contained in nonoverlapping clusters. Figures 3.23 and 3.24 show that features associated with each class are somewhat clustered, they do not seem to overlap. As a result, one would expect relatively good classification performances.

(4) Potential problems. The main drawback behind this feature selection scheme is that it relies on the existence of a dominant singular vector for each signal class considered. However, in some cases, the gap between the first two largest singular values may be too small to have a dominant singular vector. In such a case, the validity of the Learned and Willsky's approach becomes somewhat questionable. For example, this situation could occur in low SNR environments, or when the properties of the signals change significantly from trial to trial.

Let us illustrate some of the drawbacks with a simple example. An example will be given to prove this situation. The two-signal class (Mishmash/Doppler) example considered earlier was implemented for different SNR values and the corresponding difference ratios $\Delta\sigma_t$ are shown in Table 3.4. Reduced feature sets for a SNR of -10 dB are plotted in Figures 3.25 and 3.26. Results show that as the SNR decreases, the difference ratio decreases as well, endangering the feature selection approach of this method. In addition, the features associated with each signal class become more overlapped, thereby decreasing the likelihood of a good classification performance.

Further, note that the properties of the noise-free training signals were not changed in this example. In practice, one cannot usually expect such an ideal behavior, as some in-class variation will be observed. As a result, we consider a two-signal class case: high and low frequency sine. Additive white gaussian noise was added to the signals for an SNR level equal to 0 dB. Forty signals per class were used to generate the power matrices associated with each signal class, and the difference ratios obtained were 0.8097 and 0.8175. Next, the digital frequencies are changed 10% randomly around the carrier frequency to simulate a change in the characteristics of the signals and the difference ratios were 0.4034 and 0.3899. At this point, it becomes difficult to justify the existence of a dominant singular vector to select the useful features.

Finally note that: 1) selecting the nodes with the highest power values may not guarantee that the reduced feature sets will be well clustered and won't overlap, in such a case reducing the feature set may result in further information loss; and 2) this scheme assumes that the nodes with the highest power carry the most discriminant information, which may not necessarily be true.

SNR	20db	10db	0db	-10db
$\Delta\sigma_1$	**0.9587**	**0.87**	**0.6787**	**0.601**
$\Delta\sigma_2$	**0.9607**	**0.84**	**0.6579**	**0.62**

Table 3.4: Difference ratios $\Delta\sigma_t$ for different SNR values.

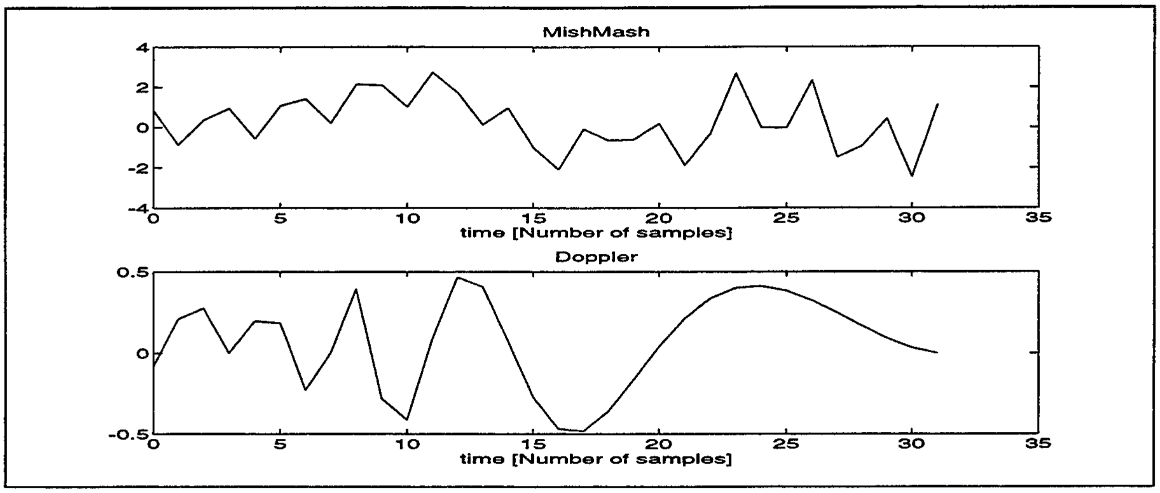

Figure 3.17: Noise free "Doppler" and "Mishmash" signals. After Ref. [7].

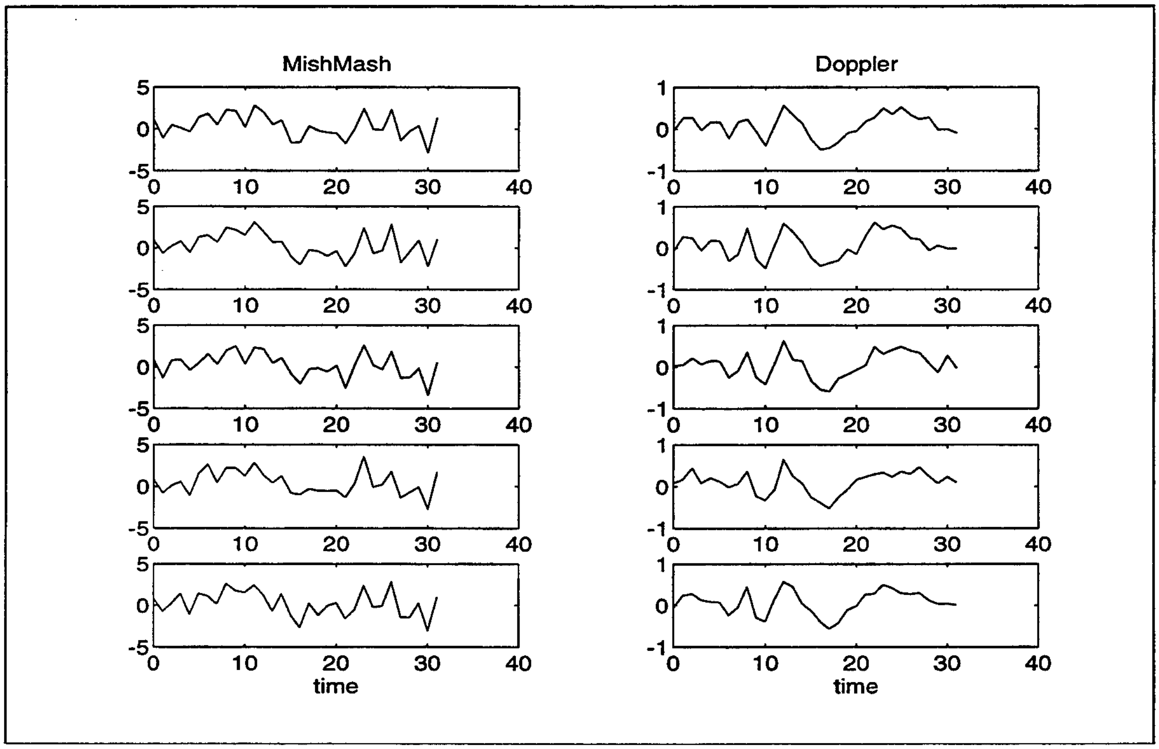

Figure 3.18: Five noisy trials for "Doppler" and "Mishmash" signals ; SNR=0dB.

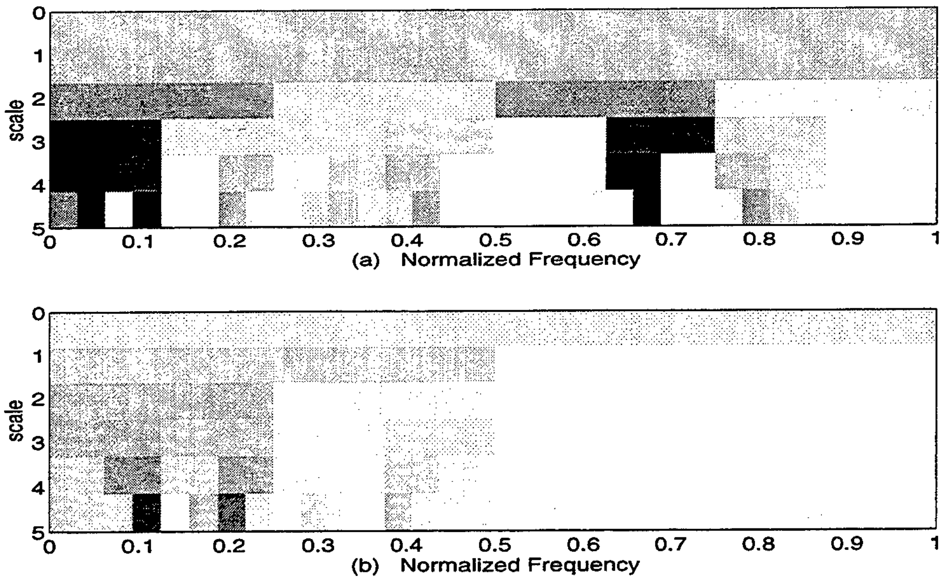

Figure 3.19: Power maps of Mishmash (a) and Doppler (b) signals.

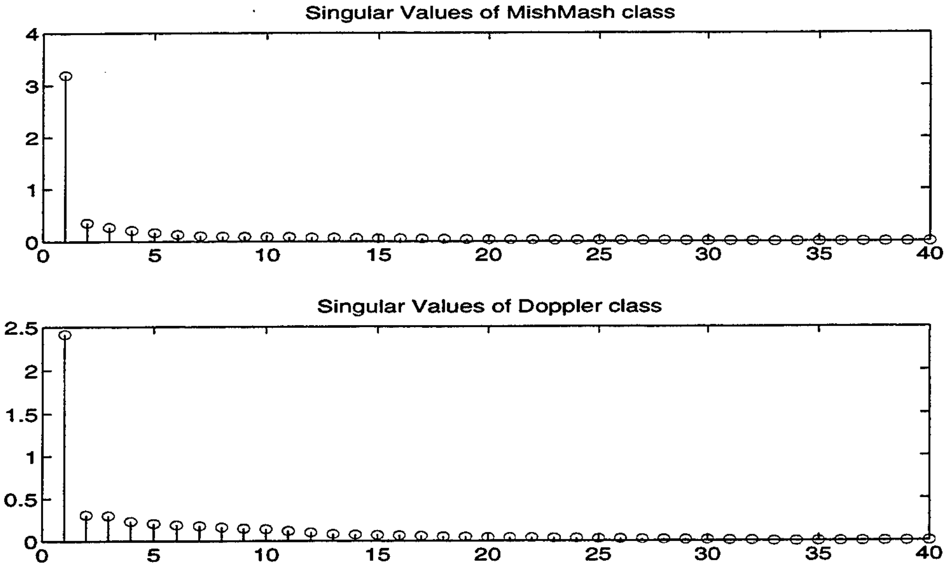

Figure 3.20: Singular values for Mishmash and Doppler classes.

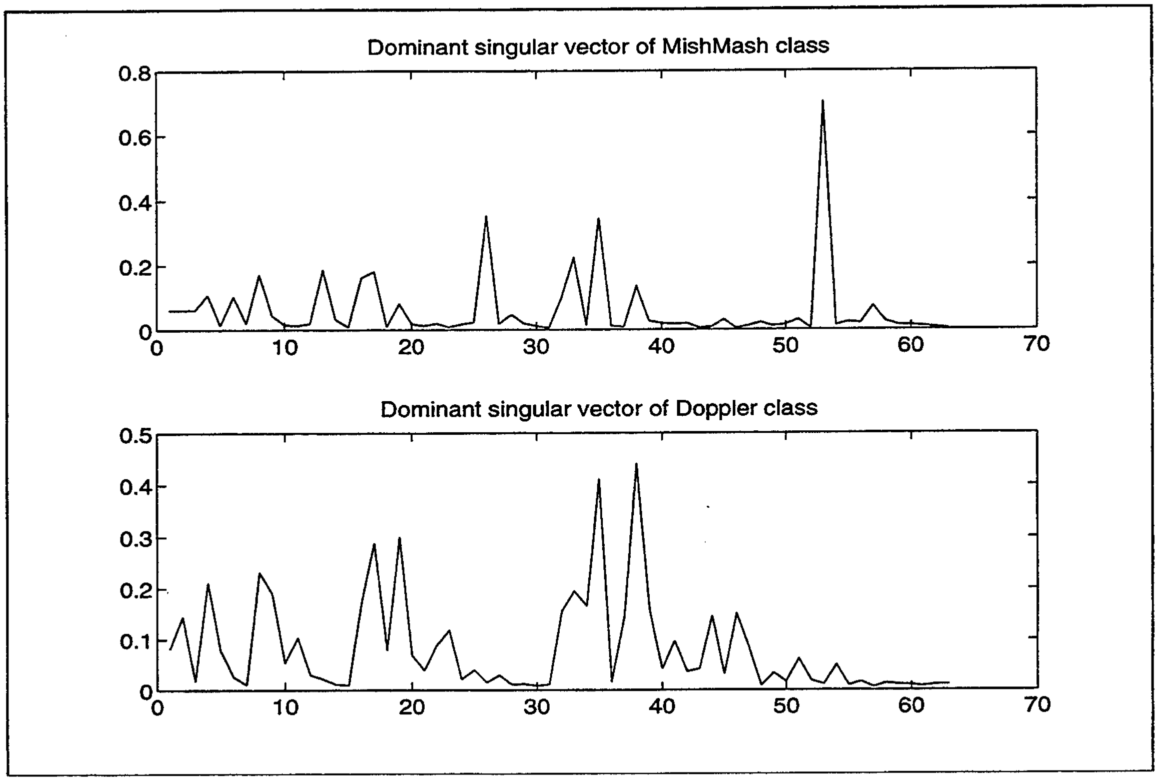

Figure 3.21: Dominant singular vectors of two signal classes.

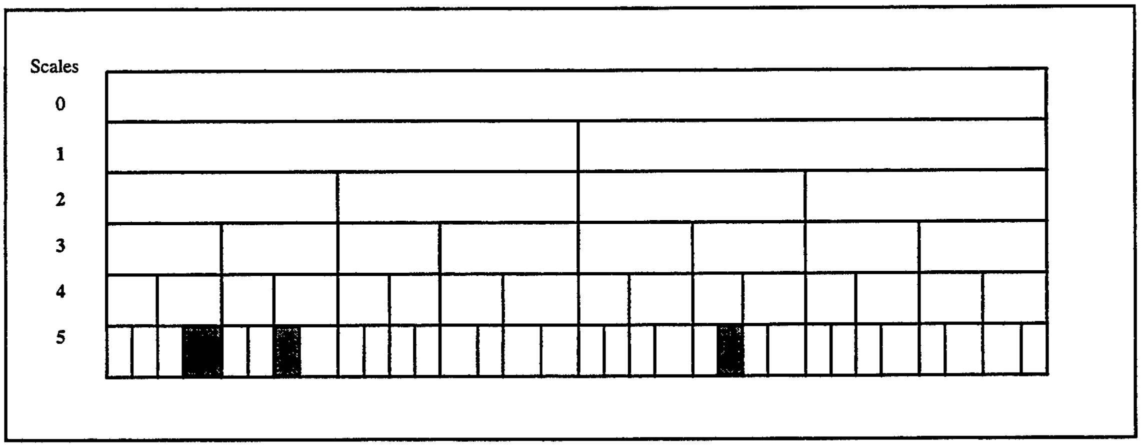

Figure 3.22 Selected nodes in the WPD scheme.

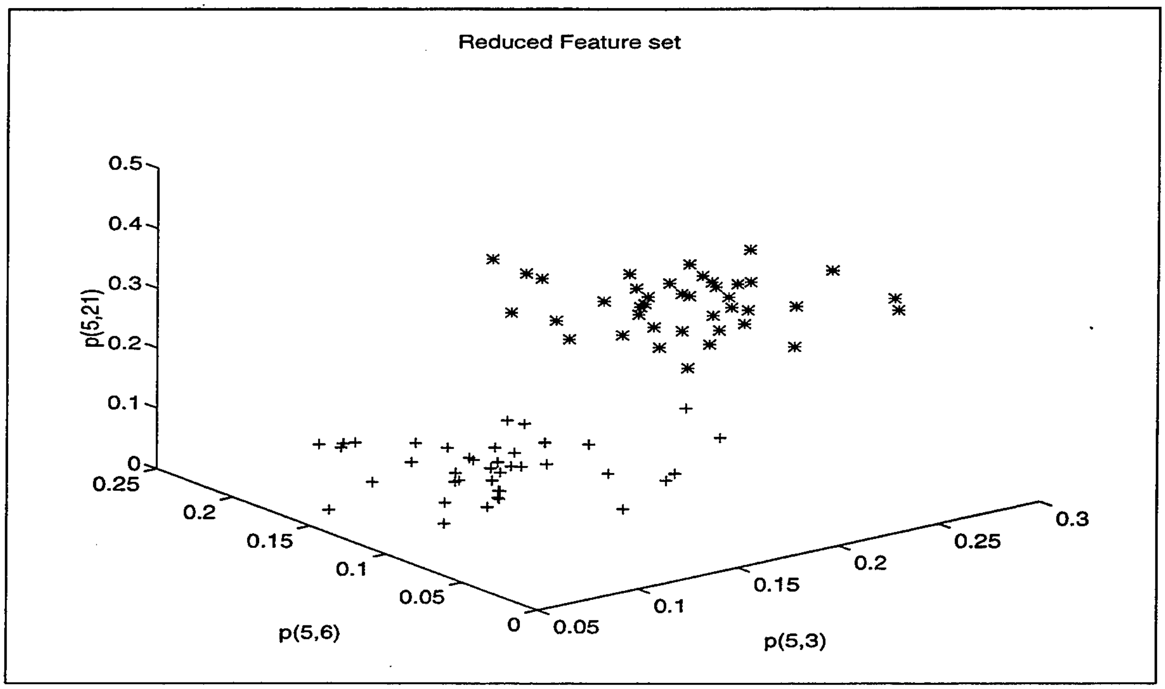

Figure 3.23: Reduced feature set in 3D, 2 signal-class example,SNR=10dB; LW feature extraction scheme.

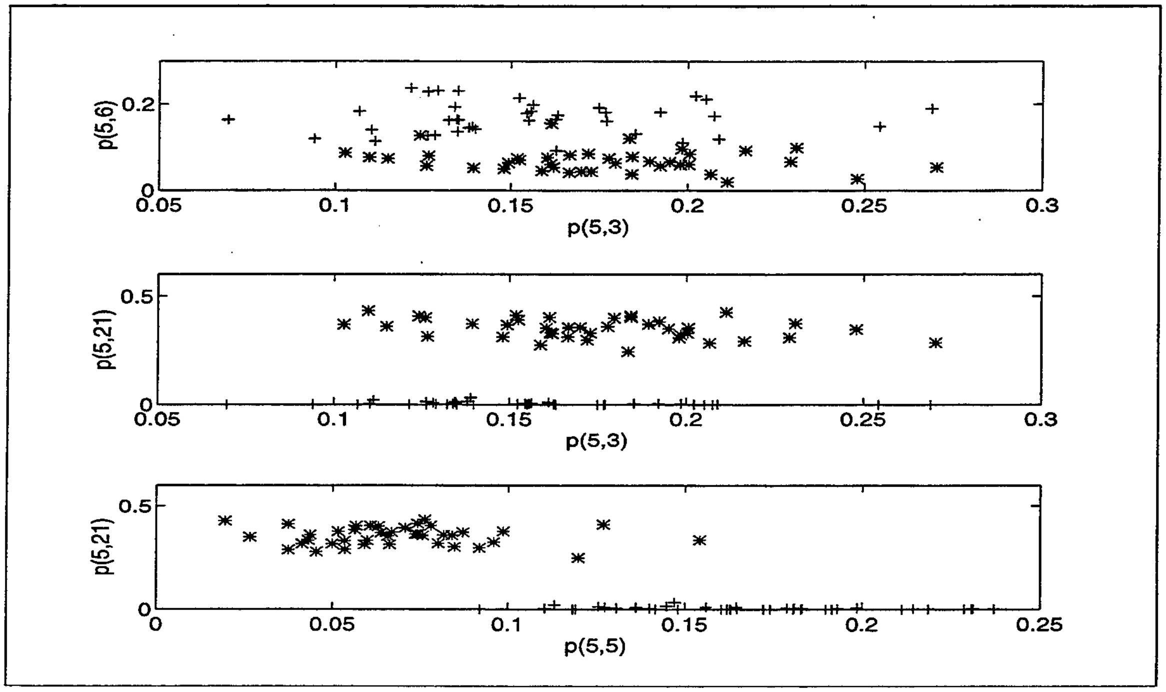

Figure 3.24: Three projections of Figure 3.23.

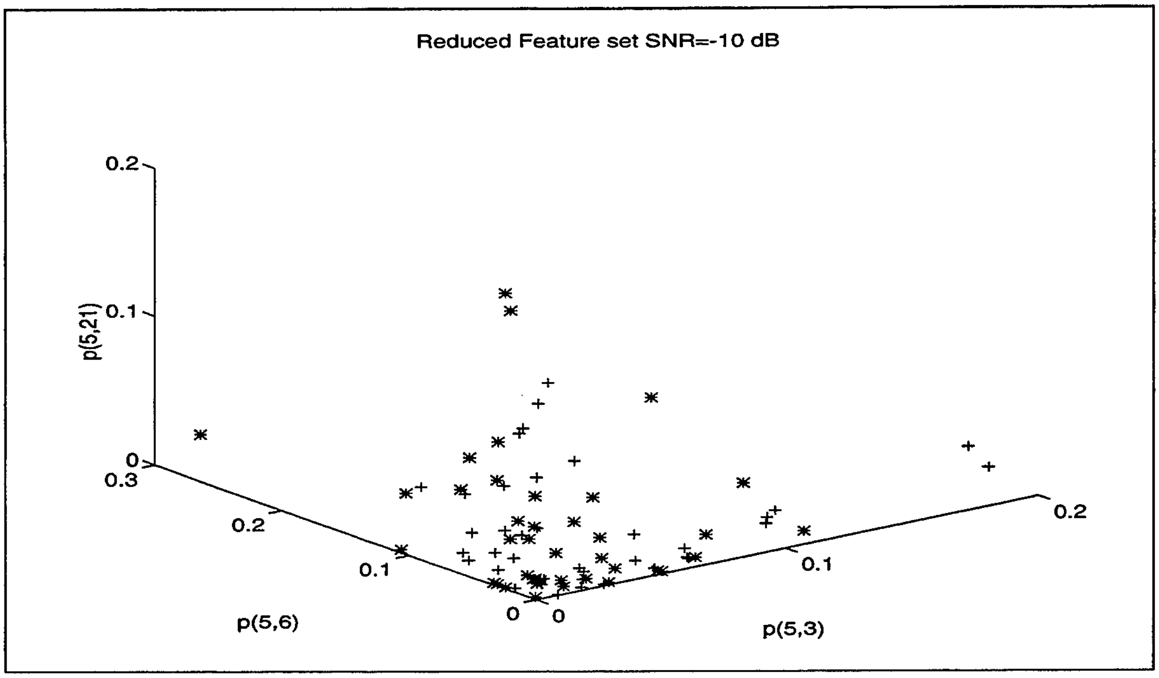

Figure 3.25: Reduced feature set in 3D, 2 signal-class example, SNR=-10dB; LW feature extraction scheme.

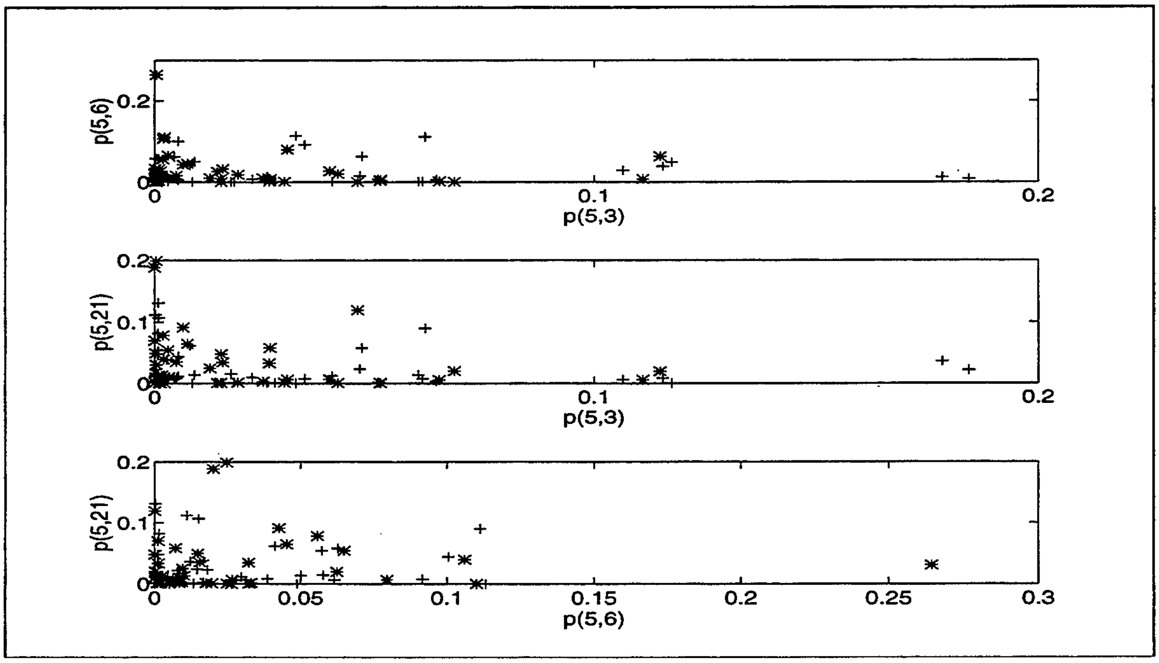

Figure 3.26: Three projections of Figure 3.25.

b) Most Consistent Feature Extraction Scheme

The scheme presented next was considered to avoid the problem due to the estimation of a dominant singular vector. Here, we select as "reliable" in-class features those which vary the least within each class. Thus, variances across the rows of the power matrix P_t of a signal class t are calculated and ordered in decreasing order (recall that each row corresponds to the same node location for all training signals of a given class). Note that the nodes with the most consistent information (i.e., those with similar power from signal trial to trial) are those with the smallest variances. These nodes will be those selected as they are the "most consistent" features of a given class. Thus, we select Q nodes with the smallest variances.

We illustrate this scheme with the two-signal class (MishMash/Doppler) example considered earlier. Variances across the rows of each signal class power matrix were calculated, and the top 5 least varying node indexes selected for each signal class. Indexes [2, 3, 6, 7, 15] and [2, 3, 5, 7, 11] are respectively selected for the classes MishMash and Doppler. The resulting node indexes used in the feature extraction task are [2, 3, 5, 6, 7, 11, 15] as indexes 2, 3, and 7 are common to both classes. These node indexes correspond to the nodes (1,0), (1,1), (2,1), (2,2), (2,3), (3,3) and (3,7) respectively, and their locations are shown in Figure 3.27. Note that the parent-children node situation is not taken into account in this method. The problem with this procedure comes from the fact that the actual magnitudes of the discriminant values are not taken into account in the selection process. Thus, the method may select nodes with small discriminant values consistent over all classes considered.

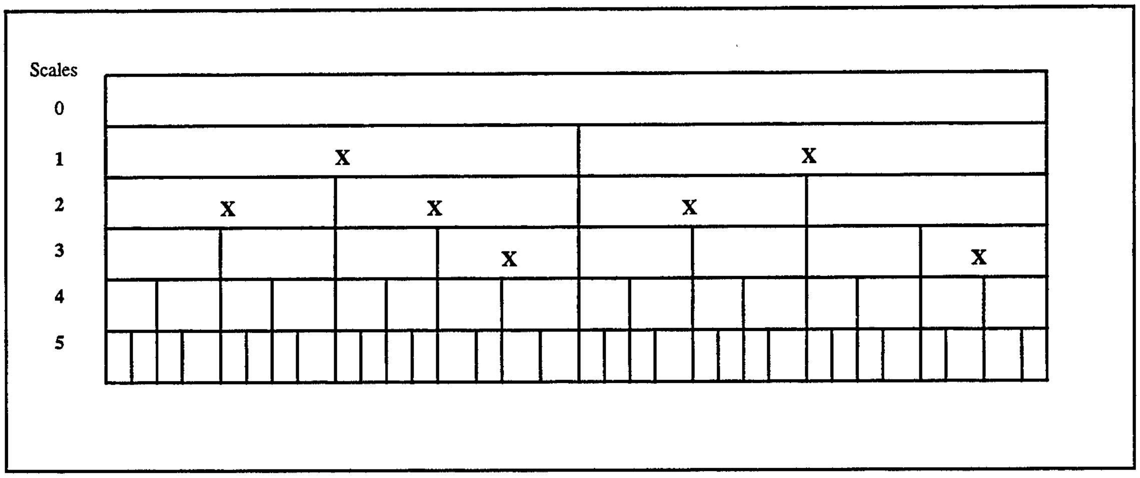

Figure 3.27: Subspaces selected by the most consistent feature extraction scheme.

c) Most Discriminating Nodes Feature Extraction Scheme

The method considered here is very similar to the LDB feature reduction scheme where the basis functions are selected according to their relative entropy values. One average feature set is first identified for each class. Next, relative entropy values for each node are calculated using Equation 3.8. Last, the nodes with high relative entropy, i.e., those which supply discriminating features, are selected.

This scheme is illustrated with the two-signal class example used earlier corresponding to the Mishmash and Doppler signal classes with SNR=10dB. First, averaged feature sets obtained for each signal class are calculated and shown in Figure 3.28. Then the relative entropy values for all 63 nodes are calculated. The top three nodes are (5,21), (4,10), (5,6), and the corresponding time frequency partitionings are shown in

Figure 3.29. Note that in this method, we do not care if a parent-children node situation exists, and the number of top nodes selected is left to the user. The reduced feature set is plotted in Figures 3.30 and 3.31. Note that the two signal classes are clustered and do not overlap, thus a good performance can be expected.

However, note that this method suffers from the same potential drawbacks as the LDB scheme does when the number of classes is larger than two. It is also based on the averaged pair-wise relative entropy value, as given in Equation 3.5, which significance becomes more and more questionable as the number of classes increases.

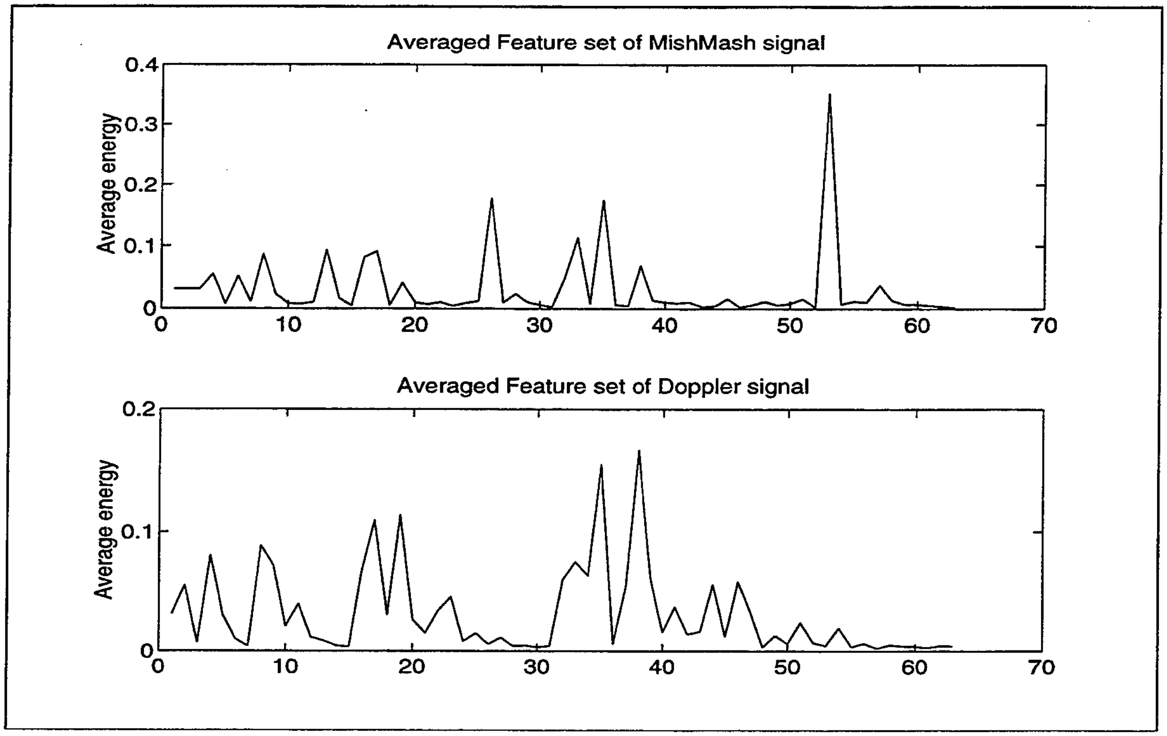

Figure 3.28: Average feature sets for Mishmash and Doppler signals. Most discriminant nodes feature extraction scheme. SNR=10 dB.

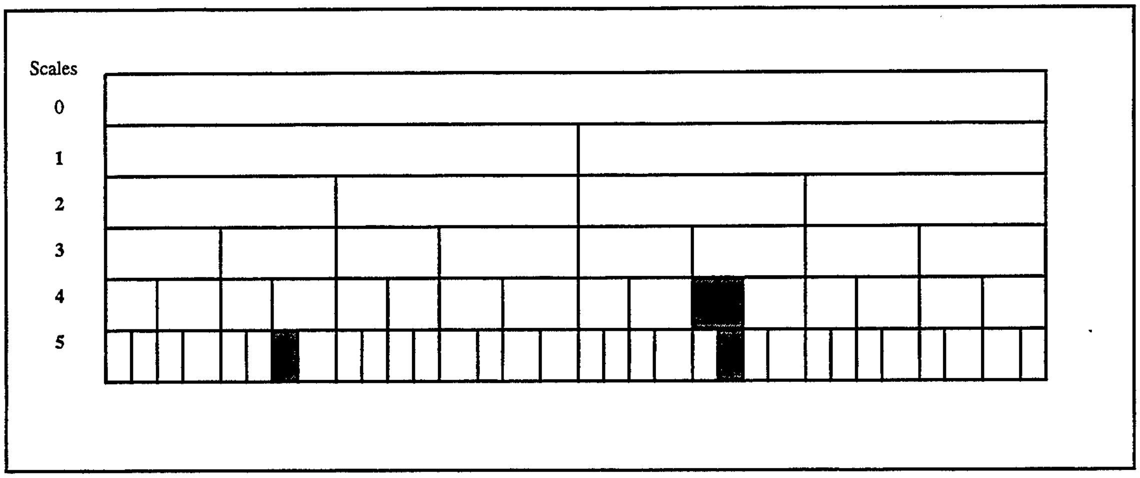

Figure 3.29: Subspaces selected by the most discriminating dimension reduction scheme.

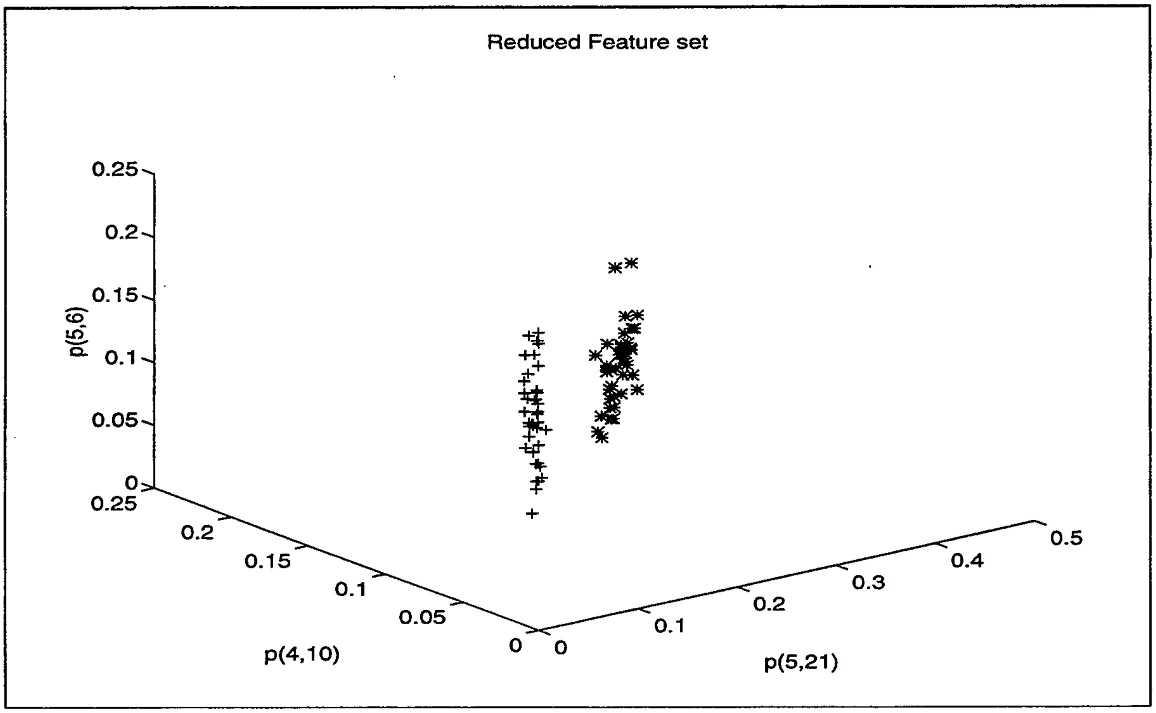

Figure 3.30: Reduced feature set in 3D, 2 signal-class example, SNR=10dB; Most Discriminating feature extraction scheme.

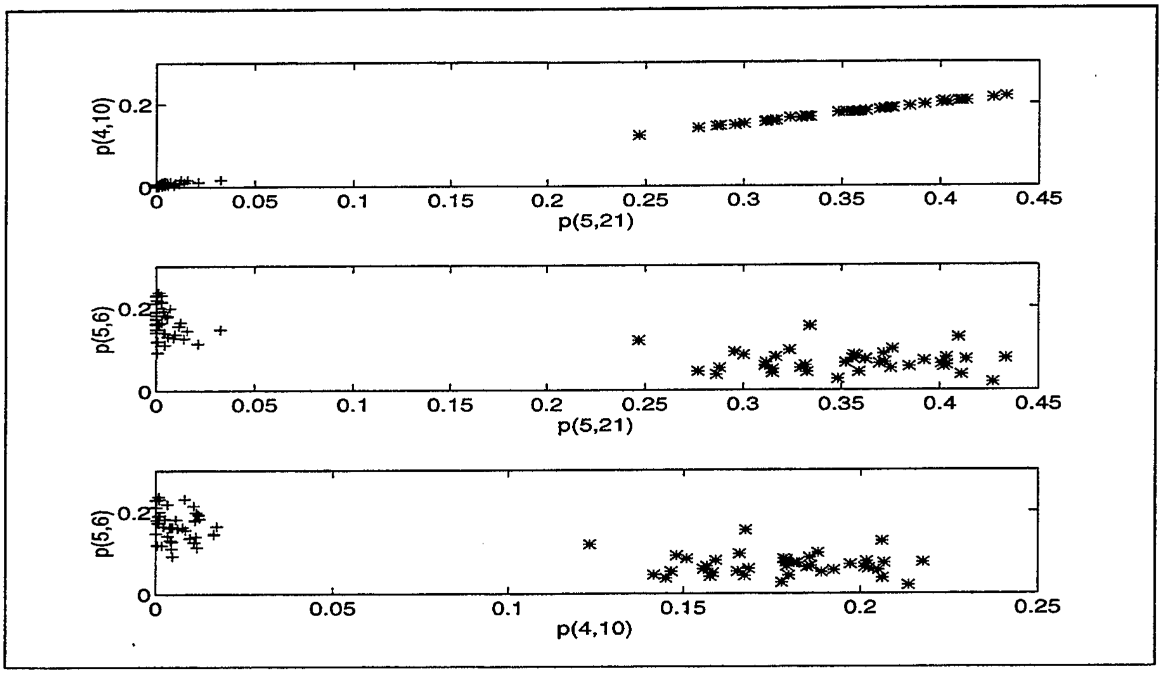

Figure 3.31: Three projections obtained from the 3D reduced feature set given in Figure 3.30.

d) LDB Based Dimension Reduction Scheme

This dimension reduction scheme simply considers using the power features associated with the nodes selected by the LDB feature extraction method. First, the LDB feature extraction method is used to extract a basis which best discriminates between signal classes. Next, the WP nodes associated with the selected basis are used to extract the power features from each given signal. The only drawback of this scheme is the fact that the number of features this scheme selects depends on the number of nodes that constitute the selected LDB basis, meaning that the number of nodes selected in this scheme may be higher than expected.

3. General Problems With The Power Method

The Power method is based on power quantities defined at each node. This averaging operation results in loss of time resolution, which may be a problem when precise timing information is needed to separate otherwise similar signal classes. However, this problem can be alleviated by defining short-time energy quantities at each node to re-introduce some timing information [21].

C. TIME SHIFTING ISSUES WITH LDB AND POWER METHODS

Recall that the Power method was found to be more robust to time-shift problems encountered in classification tasks than the LDB method is. We will illustrate this behavior by comparing each set of coefficients obtained from a given signal with Power and LDB schemes and the set of shifted coefficients obtained from a time-shifted version of the signal. The signal selected for this simple experiment is a linear chirp signal of length 128 that is zero padded with 256 zeros.

The Power and LDB coefficients are computed for the chirp, with a maximum scale decomposition of 7. The chirp is then successively time-shifted by 1 to 60 sample numbers, and the Power and LDB schemes used to computed new sets of coefficients for each time shifted versions.

Next, the nodes shown in Figure 3.32 are selected as if they were obtained using the LDB method. Figures 3.33 and 3.34 plot the squared errors obtained between the coefficients of the original signal and the shifted coefficients of the time-shifted signal for

LDB and Power method respectively, for successive time shifts between 1 and 60 samples.

Results show that as the shift increases, the difference between the original coefficients and the shifted coefficients increases when using the LDB scheme, while it remains much smaller when using the Power Method. This result is to be expected as additional time robustness is added in the Power method with the averaging operation conducted at each selected node, while no such averaging is done with the LDB scheme.

Next, Chapter-IV presents the two classification tools considered in this work: Classification Trees (CT) and Back-propagation Neural Networks (BP) .

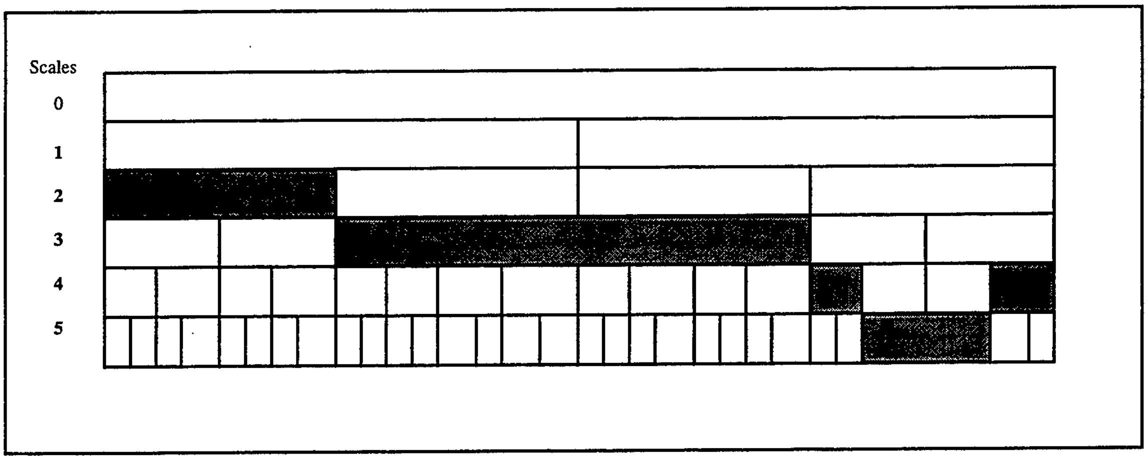

Figure 3.32: Subspace selected for showing the effects of time shifts in LDB method.

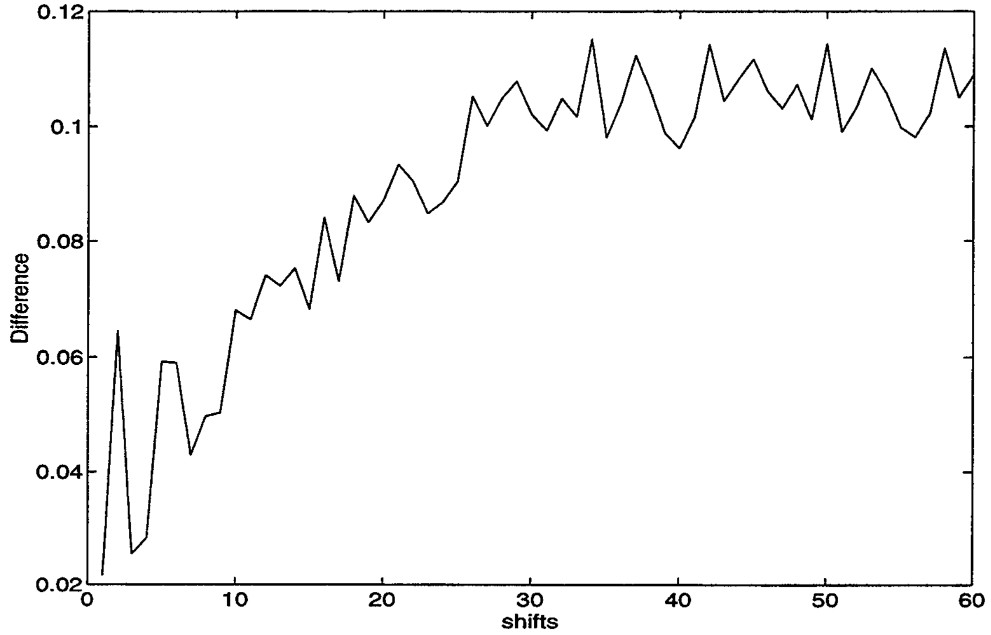

Figure 3.33: Difference values (LDB Method).

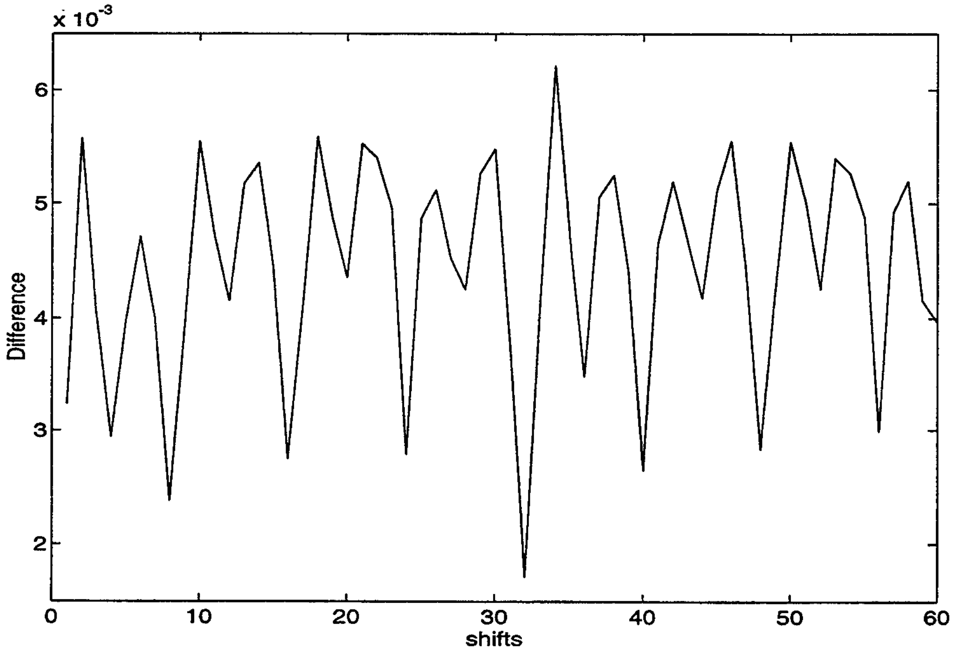

Figure 3.34: Difference values (Power Method).

IV. CLASSIFICATION TOOLS

In this chapter, we first review two pattern classification types considered in our study: Classification Trees (CT) and Back-propagation Neural Networks (BP NN). Next, we discuss how feature clustering and overlapping may affect classifier performances.

A. CLASSIFICATION TREES

1. Tree Structure

Tree-based methods have been used extensively over a long period of time in some areas such as botany, social sciences, and medical diagnosis, while only more recently in others, such as statistics and pattern recognition [17, 28]. A classification tree can be viewed as a set of if-else binary decision rules which partition the feature space into non-overlapping rectangular regions corresponding to the tree leaves. Numerous tree decision schemes have been developed over the years, however, they all are variations of the same type of algorithm which uses a top-down search through all possible solutions [27]. In this work we consider only one type of decision trees, classification and regression trees (CART) originally proposed by Breiman [28], and thus restrict our discussions to this structure only. Further details on decision trees may be found in [28,30].

CART are binary trees designed to assign a class label to a given feature vector. Let us consider a simple example to illustrate the concept behind CART. Assume that a feature vector of length N is equal to $\underline{X}=\{x_i\}_{i=1}^{N}$. For example, this set of features may have been obtained by the LDB or the Power schemes mentioned in the previous chapter. The CART scheme produces a tree based on individual features presented to the tree. For example, the first split (i.e., at the top of the tree) will be determined by a question like: "is x_i less then a given threshold?" Such a question will result in a partitioning of the feature sets and creation of a left and a right leaf to the classification tree. At this point, each of the child branches may be further split according to new decision rules. By convention, all nodes with children are called internal nodes. This splitting process may be repeated until a node without any children is reached, or the number of feature vectors belonging to the children nodes falls below a user-specified value. This final node is called a terminal node. Feature vectors belonging to that final node are then assigned a class label during testing. By convention, internal nodes are shown as circles and terminal nodes are shown as rectangular boxes, as illustrated in Figure 4.1. Notice that CART is called a binary decision tree, because every internal node has only two children nodes. Trees with more than two children per node have also been proposed, however, binary trees are the most often used because they are simple and can be easily built using a given training data set.

The key point in the creation of the CT is the selection of good decision (or splitting) rules at every internal node. Thus, this issue will be discussed next.

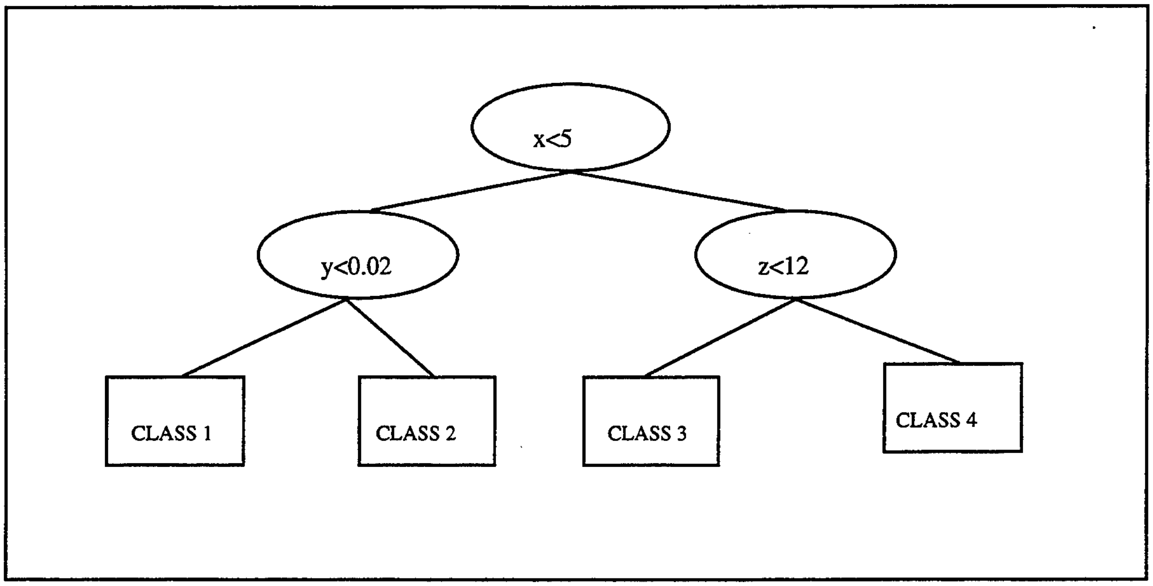

Figure 4.1: Sample Classification tree.

2. Tree Growing

As mentioned earlier, the key point in the tree construction lies in the definition of specific decision rules. The main idea behind this process is to identify the "best" question, or decision rule, for each split, where "best" should lead to the following resulting behavior:

> For all possible feature sets in the training set, identify the feature (i.e., x_i) and the decision rule threshold value which will lead to two children nodes with purer features, where "purer" means the number of feature vectors belonging to one signal class clearly outnumbers the others.

As a result, an index of impurity is defined so that it has a minimum at zero when all the given feature vectors belong to the same single class (the purest case), and reaches a maximum when the feature vectors equally belong to different signal classes (the worst case). For example, let us assume that there are J signal classes and the probability that a given feature set belongs to signal class j is denoted as p_j, then $\sum_{i=1}^{J} p_j = 1$. The impurity measure of a node t is expressed as :

$$E(t) = \Phi(p_1, p_{2,} p_3, \ldots p_j). \tag{4.1}$$

For practical purposes p_j is calculated using the equation:

$$p_j = \frac{N_j}{N}, \tag{4.2}$$

where N_j is the number of feature sets that belongs to signal class j and N is the total number of feature vectors at any internal node.

The Impurity function Φ should satisfy [27]:

$$\Phi\,(1/j, 1/j, 1/j, \ldots\ldots 1/j) = maximum, \tag{4.3}$$

$$\Phi\,(1,0,..0) = \Phi\,(0,1,0,\ldots0) = \ldots\ \Phi\,(0,0\ldots.1) = 0.$$

One possible candidate for the function Φ is the entropy function defined as:

$$\Phi(p_1, p_2, p_3, \ldots p_j) = -\sum_{i=1}^{J} p_j \ln p_{j.} \tag{4.4}$$

Once the impurity function is selected, for each given node, the scheme considers individual decision rules for each feature x_i contained in the feature set, i.e., it investigates decision rules of the type: is "$x_i <$ threshold ?" satisfied.

Obviously, the specific choice of threshold values used in the decision rules will have a clear impact on the overall tree structure. Thus, for each feature, the scheme iteratively investigates what the gain in purity (defined below) is for successive threshold values covering the full range of each feature parameter. For example, the S+ [38] tree software program which was used in this study to implement CART, has a default option which covers the complete range of each feature parameter contained in the feature vectors in given increment of the given range.

At this point, the scheme investigates whether the 2 resulting children nodes obtained with a given decision rule are purer than their parent node. The cost reduction value, that measures the purity gained after a specific decision rule is implemented, is computed using the formula:

$$\Delta E(x_i^s, t) = E(t) - p_l E(t_l) - p_r E(t_r), \qquad (4.5)$$

where s denotes the decision rule, and p_l and p_r are the percentage of trial cases in node t that belong to the left or right children node (i.e., branch) after the splitting rule s is invoked.

Therefore, at a given tree node, cost reduction values are successively computed for each possible splitting rule s invoked on each feature parameter x_i, and the decision rule with the highest cost reduction value selected for that leaf. This selection means that the decision rule chosen at a given node divides the feature set so that the children nodes are purer than their parent node. This process is repeated for successive children nodes until a pre-determined purity level is reached at any node. The node then obtained is called a terminal node, and a class label assigned to it. The class assignment is based on the class i with the highest probability p_i obtained at that given node.

Let us illustrate these concepts on a simple example. Let us assume that 25 feature sets are used to construct a classification tree to classify 5 different classes. Further, let us assume that after growing a CT, a final node has the following class distribution:

16 feature sets belong to class 1,

2 feature sets belong to class 2,

3 feature sets belong to class 3,

1 feature sets belong to class 4,

3 feature sets belong to class 5.

Thus, the resulting probabilities for each signal class obtained at this node are: 16/25, 2/25, 3/25, 1/25, and 3/25. As a result, the node is labeled as class 1. At testing, feature sets are assigned to a given class, and these probability quantities allow the user to measure the confidence with which the testing sets are classified in a given class. For example, should a test signal be assigned to node 1 during testing, one can say that belongs that class with a probability of 16/25. Note that this unknown signal may also belongs to signal class 2 with a probability of 2/25, etc.

3. Example

Assume that our training feature set has two features per signal and there are two feature sets for each of the two signal classes:

Class 1 with features ({1.5, 0.5},{2.5, 1.5})

Class 2 with features ({3.5, 2.5},{0.5, 3.5})

Figure 4.2 plots the location of the feature sets. The tree growing process starts at the root node by computing the impurity value obtained :

$E(t)$=-0.5ln(0.5)-0.5ln(0.5) = 0.693147.

Next cost reduction values measuring the purity gained after a specific decision are computed in succession for each feature for threshold values covering the feature value range. For example, Figure 4.2 shows that threshold values investigated for feature 1 are in the range [.5 3.5]*[2.5 0.5]. Next, cost reduction values $\Delta E(x_1^s, t)$ and $\Delta E(x_2^s, t)$ are computed for each decision rule, such as $x_1 \leq 1, x_1 \leq 2$, $x_1 \leq 3$, $x_2 \leq 1$, $x_2 \leq 2$, etc. Their values are listed in Table 4.1.

Decision Rule "s"	Cost Reduction Value
$x_1 \leq 1$	0.2157
$x_1 \leq 2$	0
$x_1 \leq 3$	0.2157
$x_2 \leq 1$	0.2157
$x_2 \leq 2$	0.6931
$x_2 \leq 3$	0.2157

Table 4.1: Cost reduction values for various decision rules.

Table 4.1 shows that the decision rule "$x_2 \leq 2$?" has the maximum cost reduction value and this decision rule will be assigned to the root node. Such a selection makes sense, because it can be viewed as drawing a perpendicular line from the y axis, which partitions the input space perfectly. The resulting CT can be seen in Figure 4.3 with class probabilities assigned to terminal nodes. Note that children nodes become terminal node for this example, because there is no need to partition these nodes again.

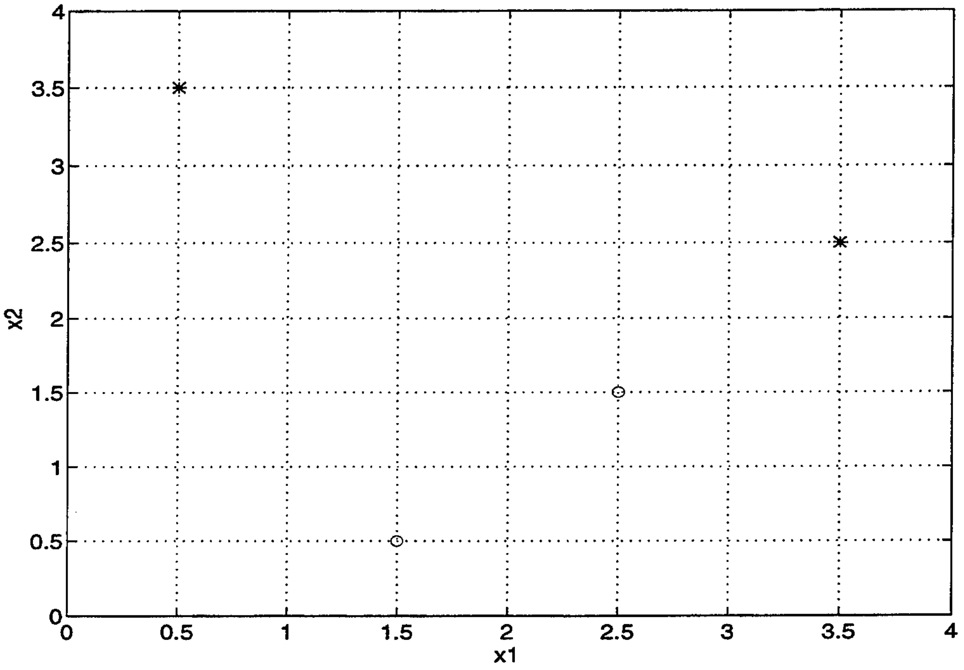

Figure 4.2: Feature set locations.

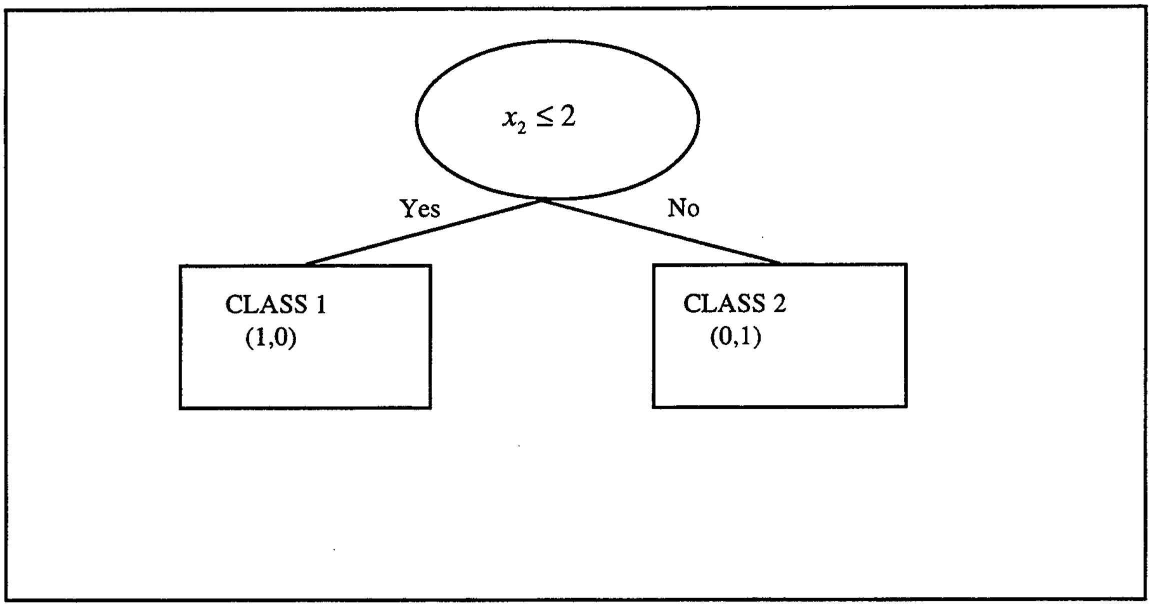

Figure 4.3: CT for the class clusters shown in Figure 4.2.

4. Problems with Classification Trees

The performance of the CT depends on the cluster locations of the signal classes. Recall that CT may be viewed as implementing a set of decision rules which are designed to separate the feature space into a set of non-overlapping rectangular regions containing class clusters. Thus, perfect classification will result when classes can be separated by perpendicular lines associated with each decision rule. However, when such separation is not possible, as shown in Figure 4.4, CT may have difficulty in separating classes even though the classes may be visually clearly separated and without any overlap. Eventually though, a tree-based classifier can be found to separate classes shown in Figure 4.4, but the resulting tree structure will be quite complex, as shown in Figure 4.5. In such cases, other types of classifiers may be better suited to the problem. Next, we consider BP neural network classifiers.

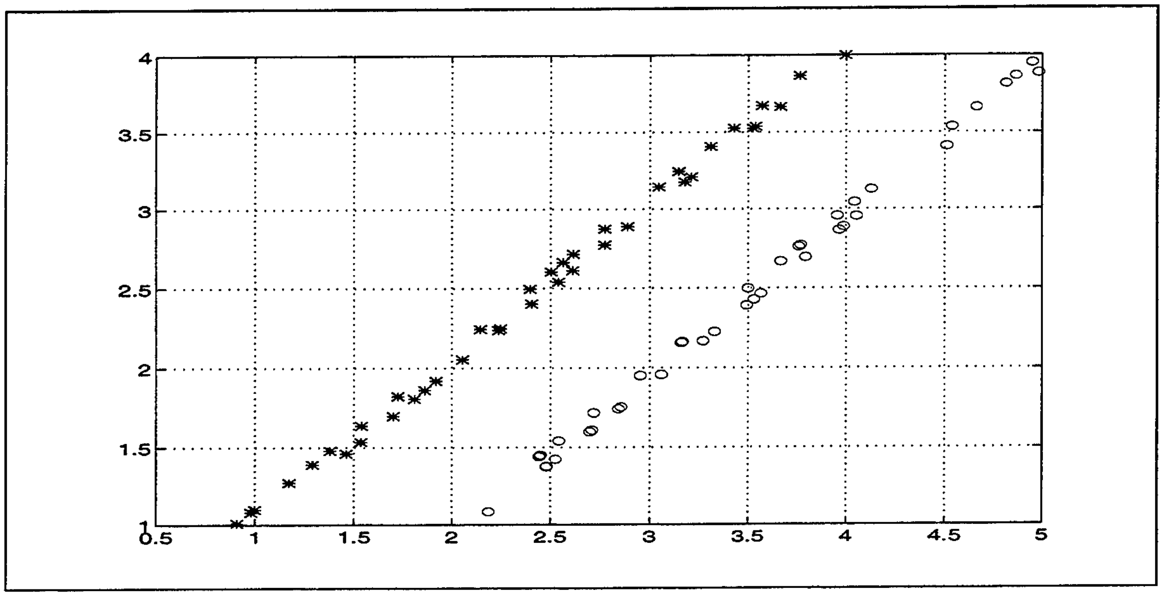

Figure 4.4: An example where clusters can not be easily separated by perpendicular lines.

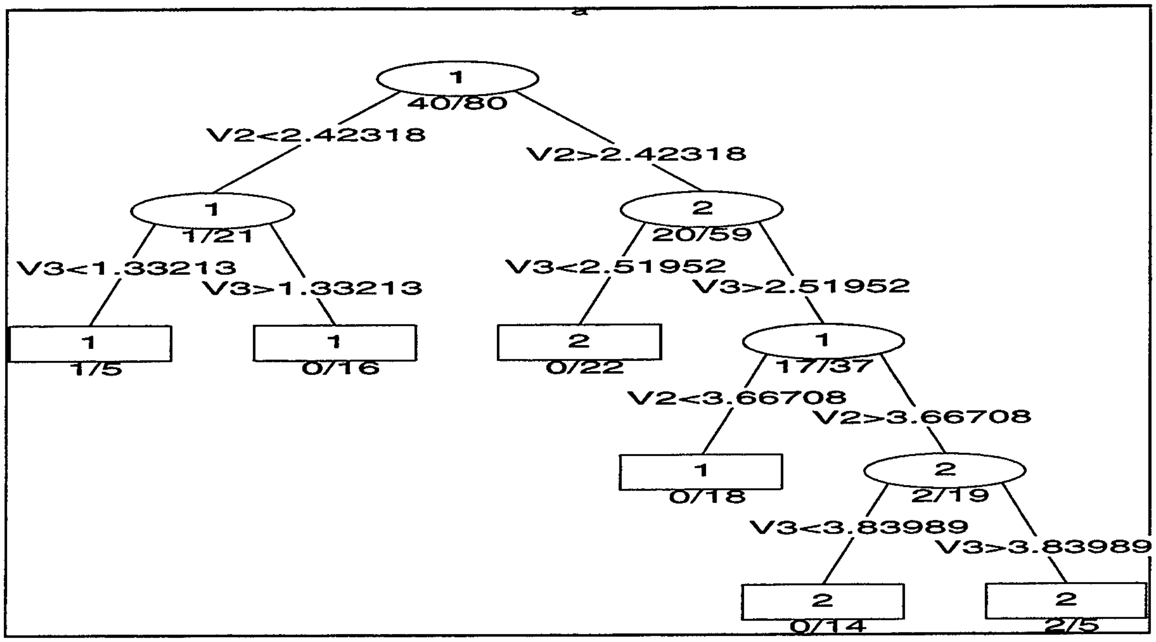

Figure 4.5: CT for the class clusters shown in Figure 4.4.

B. THE BACK-PROPAGATION NEURAL NETWORK

The back-propagation neural network is one of the most powerful classification tools available today [29]. It consists of one input layer, multiple computational layers, also known as hidden layers, and one output layer. In this study, we restrict our investigation and our discussions to one-hidden layer neural networks (NN) as they were shown to be sufficient for the data considered. However, this discussion can easily be generalized to k-hidden layer NNs. Figure 4.6 shows the general architecture of a NN implementation. Circular elements denoted as PE are processing elements. These elements are the building blocks of the neural network and Figure 4.7 shows their general diagram. The notation used in this diagram is explained below [24]:

- $x_j^{[s]}$ → output of the j^{th} PE in the layer s
- $w_{j,i}^{[s]}$ → weight of j^{th} PE in layer s that will be multiplied with input $x_i^{[s-1]}$
- $R_j^{[s]}$ → weighted summation of inputs to the j^{th} PE in layer s
- φ → Transfer function

As seen from Figure 4.7 $x_j^{[s]}$ can be computed as:

$$x_j^{[s]} = \varphi\left(\sum_i w_{j,i}^{[s]} x_i^{[s-1]}\right) = \varphi(R_j^{[s]}) \qquad (4.6)$$

The non-linear transfer function chosen in this study is the log-sig function defined as:

$$\varphi(x) = \frac{1}{1+e^{-x}}. \qquad (4.7)$$

This transfer function has a range of output values from 0 to 1. During the training process, target outputs are assigned to each input feature vector and the weights of the NN is adjusted so that its outputs best match the target outputs. In other words, the NN maps the input values to desired output values (target outputs). This process can be thought of as an error minimization process with respect to the NN weights, if the error is defined as the difference between target and NN outputs. In order to minimize this error, weight updates are needed during the training process. Basically, the error is first computed after the input vectors are applied to the NN and this error is back-propagated to compute the weight updates. Further details may be found in Ref. [24, 27].

In this thesis, we use the Neuralwork Professional II/Plus software to generate BP neural networks [31]. Thus, we briefly review the main options selected in this software.

1. Learning Rule

The Learning Rule (LR) is used to update the weights in order to decrease the error value. Note that some learning rules may outperform others, depending on the specific error surfaces obtained. In this thesis, we use the Normalized-Cumulative Delta learning rule. Further details regarding this LR may be found in Ref. [24].

2. MinMax Tables

Saturation of the selected transfer function may occur if the inputs to the NN are not properly scaled prior to applying to the transfer function. When the transfer function gets saturated, its derivative becomes nearly zero, which causes a zero weight update. In such a case, weights are not updated during the training process and the NN does not learn. To avoid this situation MinMax tables are generated prior to the training process using the training data set. The training data set is then scaled according to these tables to avoid staturation of the transfer function.

3. Classification Rate

The classification rate (CR) information is contained in a N by N confusion matrix, where N is the number of signal classes. Basically, this matrix shows the performance of the NN given the testing data set. After the NN is trained with the training data set and proper weights are obtained, a feature set belonging to a known class is presented to the NN. The outputs of the NN are computed and the PE at the output layer with the highest value selected as the class of the signal. Diagonal entries of the confusion matrix contain the correct classification decision percentages, while off-diagonal elements correspond to misclassification. Thus, perfect classification occurs when the confusion matrix is equal to the identity matrix. Overall classification rates are obtained by averaging each class classification rate, i.e., by averaging the confusion matrix diagonal elements.

4. Network Architecture

The type of NN architecture chosen in this study is a one-hidden layer, where the number of PEs in the hidden layer is chosen as 1/5 of the number of input features. We realize that there are several rules of thumb to select the number of PEs in the hidden layer. However, this study mostly deals with feature extraction tasks and using another architecture was shown to improve the performance only slightly. As a result, optimization of the neural network implementations was not considered further.

5. Training and Testing the Neural Network

Feature vectors are put into a matrix in row-wise fashion and target outputs are attached to each row in order to train and test the neural network with the NeuralWare software. Target outputs consist of a one in the correct signal class location and zeros otherwise. For example, the target vector equal to [0,0,0,0,0,1] denotes that there are 6 signal classes and this feature vector belongs to signal class 6. NN training is stopped when a desired average classification rate is reached or when this value converges to a user-specified value.

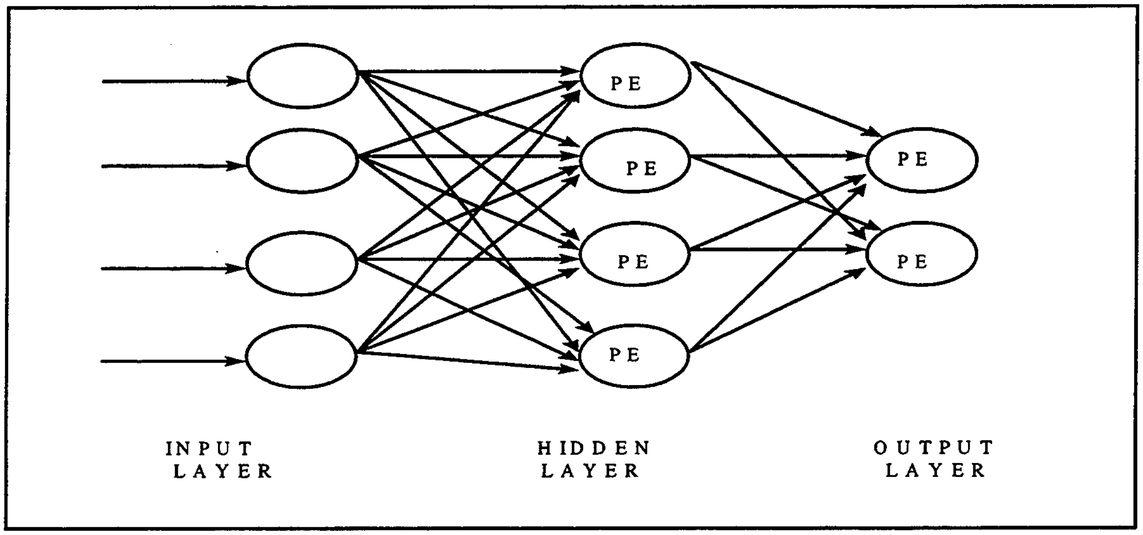

Figure 4.6: Typical back-propagation neural network with one hidden layer.

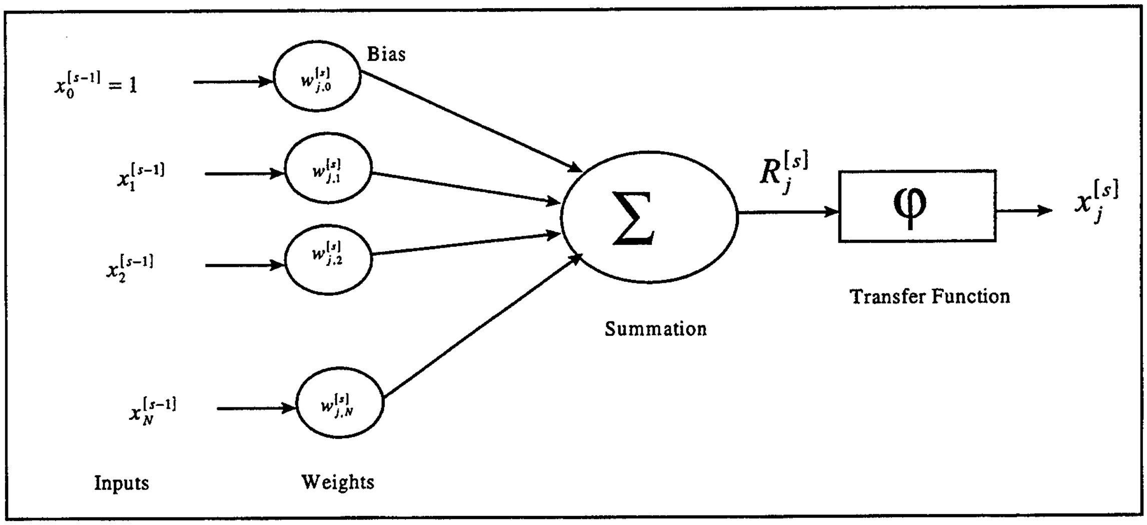

Figure 4.7: Processing Element (PE).

C. EFFECTS OF CLUSTERING AND OVERLAPPING ON THE PERFORMANCE OF CLASSIFIERS

We stated earlier in Chapter III that good classification performances should be expected when class clusters do not overlap. This section further considers class clustering and overlapping issues, and illustrate them on some synthetic classification examples.

1. Two Non-overlapping Signal Classes in 2D

Let us assume we have 2 classes of two-dimensional features. The first and second feature of class 1 are respectively defined as uniform random variables with densities U[1,3] and U[7,9]. The first and second features of signal class 2 are respectively defined as uniform random variables with densities U[4,6] and U[7,9]. These two signal classes are plotted in Figure 4.8. A BP neural network with the architecture 2-1-2 is used for this problem. The number of testing signals per class is fixed at 200, while the number of training data set is increased gradually to investigate its effects on the performance and average classification rates. The average classification ratios of this experiment are listed in Table 4.2.

It is obvious that, as the number of training data set increases, performance gets better. The BP neural network tries to figure out the cluster borders using the training data set. As the number of training data set increases borders become more and more accurate and performance gets better accordingly.

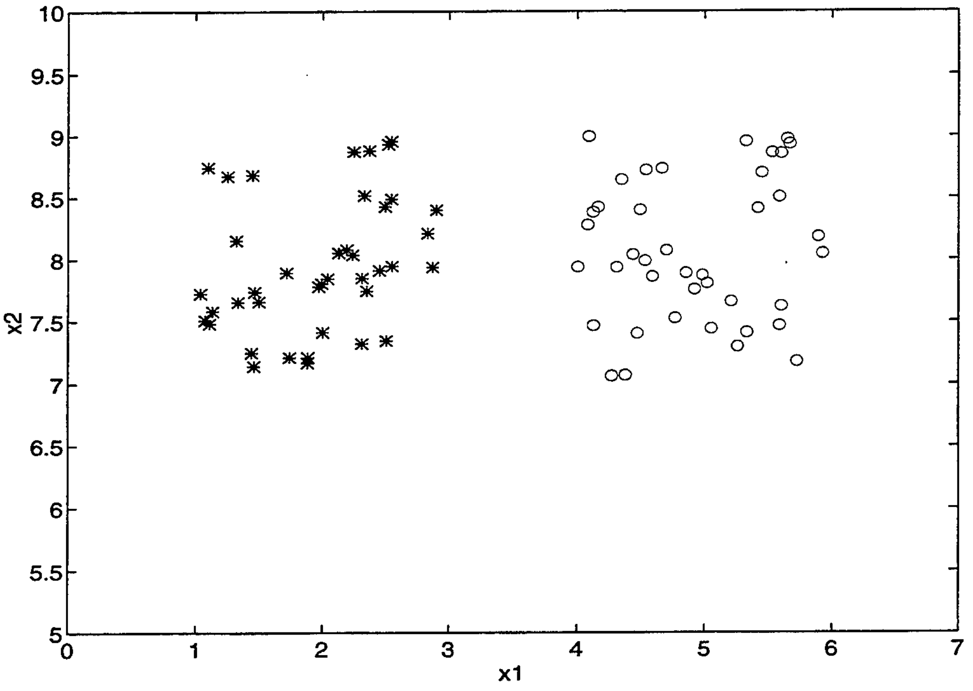

Figure 4.8: Cluster locations: two nonoverlapping signal classes.

The number of training signals per signal class	Average Classification rate (%)
1	77.5
5	100
10	100
20	100
30	100

Table 4.2: Classification rates versus number of training data set.

2. Two Overlapping Signal Classes in 2D

Assume we have two bi-dimensional signal classes. The first and second features of signal class one are uniform random variables with densities U[1,5] and U[7,10]. The first and second features of signal class two are uniform random variables with densities U[3,7] and U[7,10]. These two signal classes overlap, as shown in Figure 4.9. A BP 2-1-2 neural network is used as a classifier. In this example, we gradually increase the size of the training data set to investigate its effect on the performance and average classification rate. The testing data set is set at 200 testing signals per class.

Results listed in Table 4.3 show that the performance of the neural network does not lead to perfect classification when the size of the training set increases, as was noted in the previous example. This performance loss is due to the presence of class cluster overlaps.

The number of training signals per signal class	Average Classification rate (%)
1	57.25
5	80
10	75
20	81.25
30	82
50	81
100	82
200	82.5
500	81

Table 4.3: Classification rates versus number of training data set.

Similar experiments were also performed for more than 2 signal classes in 2D and in higher dimensions, leading to the same conclusions. Thus, this example showed that signal class clusters should be separate to insure that increasing the training data size improves classification performances. Conversely, increasing the training data size will not guarantee improvements in classification performances when data class information overlaps. In such cases, classification performances may reach a plateau and no longer improve.

At this point one may wonder how decreasing the dimension of the data set helps the neural network. Non-relevant feature parameters contained in the training feature sets impede the training process as more data will be required to train the NN. Thus, removing these features may enable the NN to learn class boundaries with fewer training data. Finally, note that the confusion matrix gives a good insight about the positions of data clusters. For example we may conclude that clusters belonging two signal classes overlap, when the NN does not differentiate the two classes well. We added a third signal class to our previous example to illustrate this comment. The first and second features of this added signal class are uniform random variables with densities U[-1 ,-2] and U[2,3], as shown in Figure 4.10. A NN with architecture 2-2-3 was used, and 100 training and 200 testing data set were applied. The resulting confusion matrix is shown below.

Average Class. Rate: 84.3%				
		True Class Label		
		1	2	3
Declared as Class	1	67.5	14.5	0
	2	32.5	85.5	0
	3	0	0	100

The confusion matrix shows that perfect classification is obtained for class three, as it does not overlap with any of the other classes, while performances degrade for the other two classes due to the partial overlap of their feature information.

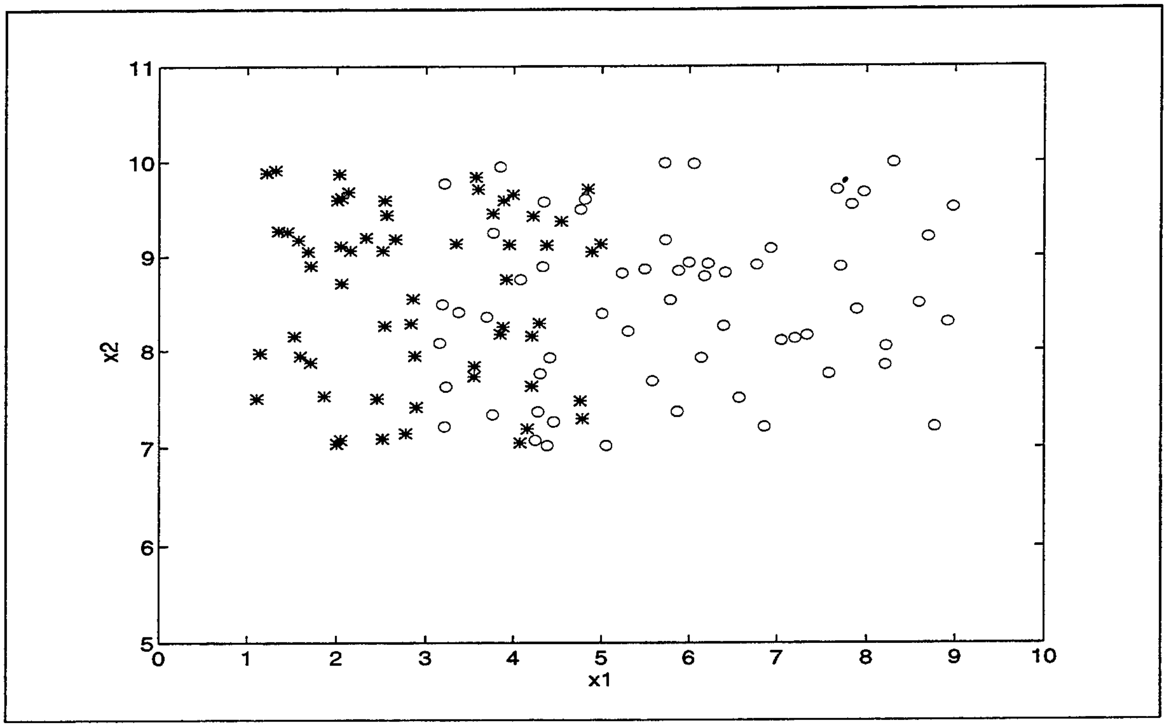

Figure 4.9: Cluster locations: two partially overlapping signal classes.

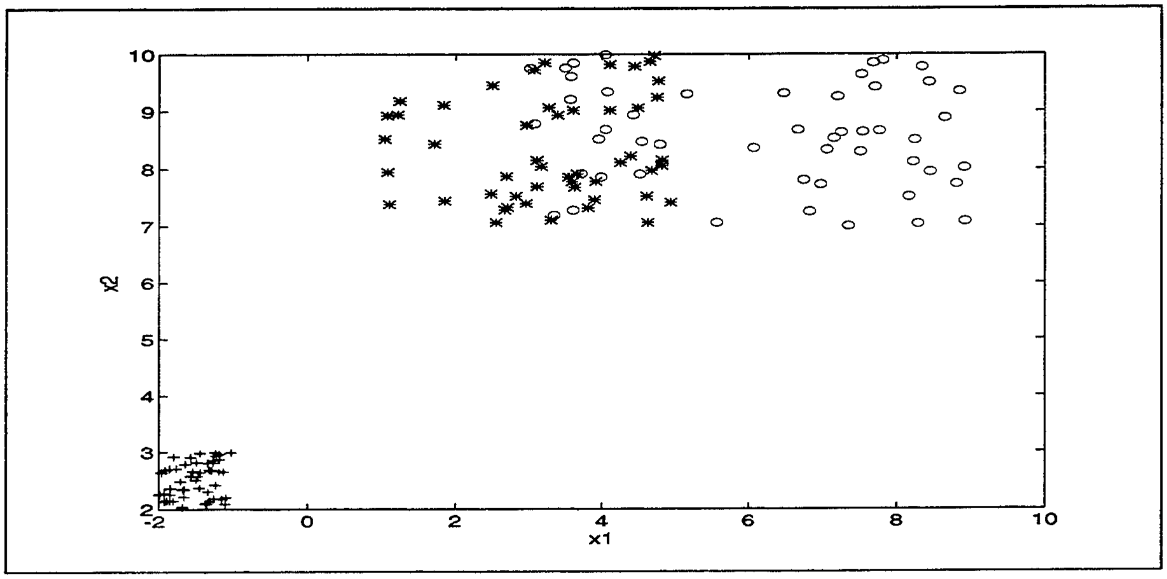

Figure 4.10: Cluster locations: two partially overlapping signal classes, one non overlapping class.

V. DIMENSION REDUCTION

This chapter first considers feature space dimensionality issues and their impacts on classification tasks. Next, it presents two dimensional reduction schemes and their application to classification tasks.

A. CURSE OF DIMENSIONALITY

Classification tasks require the user to extract features which contain as much discriminating information as possible. Such schemes may potentially lead to feature vectors of high dimension. However, the amount of training data needed to create good classifier performance grows exponentially with the dimension of the input feature space, as first discussed by Bellman who referred to the constraint as “curse of dimensionality” [20]. This constraint is especially applicable to PDF (Probability density function) based classifiers (like maximum likelihood classifier, etc.) and to a lesser extent to BP neural networks. Recall that in Chapter IV, we discussed the fact that non-relevant feature parameters contained in the training process, impede the training process as more data will be required to train a BP NN. As a result, removing these features may enable a BP NN to learn class boundaries with a smaller training data set. Therefore, reducing the dimension of the feature space is usually needed to obtain good classification performance in real-world problems where the amount of training data available may be somewhat restricted. Thus, classification schemes usually include the following few steps, as illustrated in Figure 5.1:

1-Feature extraction from the raw data,

2-Reduction of the feature space, using a suitable tool,

3-Classification based on the lower dimension feature space, using a BP, CT, or other classification tool.

There are several dimension reduction techniques available today, and some of these techniques were mentioned earlier in Chapter III. For example, projection methods select linear combinations of the features to emphasize class separation [33, 36, 37]. Projection pursuit methods look for a small dimensional projection (usually one- or two-dimensional) of the feature space which emphasizes some user-specified measure of interest. Next, we will first present the concept of projection pursuit, and then introduce two dimension reduction tools.

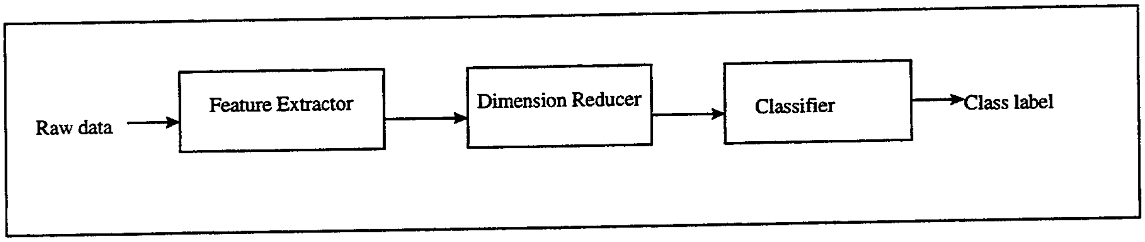

Figure 5.1: Feature reduction/Classification model.

B. PROJECTION PURSUIT

Reducing the feature space dimension can be obtained by considering projections of the high dimensional feature data set into a space with smaller dimensions. For example, consider a feature vector $\underline{a}$ of size (mx1) and a projection matrix P of size (mxn) where m>n. The vector $P^T \underline{a}$ can be viewed as a low-dimensional projection of the vector $\underline{a}$ of size (nx1). Projection pursuits (PP) schemes are designed to find the "most interesting" projection matrix P. Numerous choices for P are possible, depending on which projections are considered to be "interesting." In all cases, the basic concept behind PP schemes is to assign a projection index that indicates the degree to which each projection considered is interesting, and optimizes the index with respect to the parameters defining the projection matrix P. Thus, the core of this type of algorithm lies in the selection of the projection index. However note that, whatever the selected projection index is, information should not be lost during the dimension reduction process, as otherwise worsening in classification performances would result, assuming enough data is available to start with.

Next, we will discuss two specific projection pursuit methods with different projection indexes.

1. BCM Unsupervised Neural Network

a) Introduction

In this dimension reduction technique, "interesting" projections are motivated by an observation made by Diaconis and Freedman (1984) [32] who noted that most of the low-dimensional projections of high-dimensional clusters tend to be normally distributed [32]. This finding suggests that the information in the high-dimensional feature space is transferred to the directions which produce low-dimensional projections far from Gaussian [32, 33]. Thus, projections leading to low-dimensional distributions which are far from Gaussian will be considered as "interesting," and a projection index that will identify those projections will be employed in the feature reduction scheme.

The BCM network (referred to as BCM in the following) was originally developed by Bienenstock, Cooper and Munro in 1982 to examine the synaptic plasticity in visual cortex [32]. It was later adapted by Nathan Intrator to the dimension reduction concept [32]. BCM is an unsupervised neural network, meaning that during the training process no target output is assigned to the input feature vectors. BCM contains processing elements (PEs), as the BP neural network does. Figure 5.2 presents the operations involved in the BCM: a dot product of the input feature vector with the weight vector, followed by a nonlinear activation function. However, the PEs configuration is somewhat different from that used in the BP network, due to the lateral inhibition operation, which prevents any neuron from outperforming the others during the training process [34]. Basically, it insures that the weights associated to all the neuron outputs will be equally taken into account and will lead to all neuron outputs with distributions far from Gaussian. Selection of a proper lateral inhibition factor is still an open research subject, and still remains one of the main drawbacks for BCM-based classification schemes.

A sample BCM neural network with two PEs is shown in Figure 5.3. At this point, note that each PE output represents one of the reduced dimension features, and the weights associated with each PE can be considered as a projection of the input feature space into a one-dimensional space. Thus, if there are two PEs in a BCM configuration, then the dimension of the input space is reduced to two. As a consequence, the main idea behind the BCM projection selection scheme lies in finding the weights that will result in non-Gaussian distributed PE outputs.

b) Projection index

The projection index used in the BCM scheme for a single neuron and no activation function is called the Risk, *R(m)*, which is designed to measure the degree of skewness of the PE output values $\underline{x} \bullet \underline{m}$ from normality. The Risk function *R(m)* is defined as:

$$R(\underline{m}) = -\frac{1}{3}E\left[(\underline{x} \bullet \underline{m})^3\right] + \frac{1}{4}E^2\left[(\underline{x} \bullet \underline{m})^2\right], \qquad (5.1)$$

where $\underline{x}$ and $\underline{m}$ are the input and weight vectors respectively and the "•" operation is the dot product.

(1) Example. Let us consider a simple example to illustrate the concepts described above. Assume we have two one-dimensional classes. The first one contains data with Gaussian distribution N(0,1), while the second one contains data with uniform distribution U(1,2). One thousand data points are considered in each class, and their histograms plotted in Figure 5.4. The risk values obtained for the two classes are equal to 0.2653 and 0.1115, respectively, which illustrates the fact that the risk value is lower for non-Gaussian data.

c) BCM training process

The BCM training process is designed to find the PE weights that minimize the risk value. This training process can be viewed as trying to reach the bottom of a multidimensional risk surface defined in terms of the weight vector coefficients. For example, assume that the input feature dimension is two and the user wishes to reduce it to one using a BCM scheme with one PE. In such a case, the PE has two weight coefficients, and the dimension reduction involves a two-dimensional minimization of the risk surface R with respect to the two PE weight coefficients. The minimization scheme used in our implementation is the steepest descent method [29] which has the following weight update equation:

$$\underline{m}(n) = \underline{m}(n-1) + \mu \frac{\partial R}{\partial \underline{m}}, \tag{5.2}$$

where the learning μ is specified by the user, and *$\underline{m}$(n)* represents the weight vector at time sample *n*.

Thus, Equation 5.2 shows that the core of the training process lies in the estimation of the partial derivative expression for the risk function *R($\underline{m}$)*, which is considered next.

(1) Single Neuron Case. First, assume that this single neuron (PE) has no activation function $\Phi(x)$. The partial derivative of the risk function *R($\underline{m}$)* defined in Equation 5.1 with respect to the weight vector $\underline{m}$ becomes:

$$\frac{\partial R}{\partial m_i} = -E[(\underline{x} \bullet \underline{m})^2 x_i] + E[(\underline{x} \bullet \underline{m})^2] E[(\underline{x} \bullet \underline{m}) x_i], \tag{5.3}$$

where m_i and x_i are the i^{th} components of the weight and input vectors.

Now assume that the activation function Φ *(x)* is nonlinear. For example, a logsig function scaled between -10 and +10 was used in our BCM scheme. In such a case, the single BCM neuron output will be $\Phi(\underline{x} \bullet \underline{m})$ and the associated risk value becomes:

$$R = -\frac{1}{3}E[\Phi^3(\underline{x} \bullet \underline{m})] + \frac{1}{4}E^2[\Phi^2(\underline{x} \bullet \underline{m})]. \qquad (5.4)$$

It can be shown that the partial derivative of the risk function with respect to the weight coefficients is given by:

$$\frac{\partial R}{\partial m_i} = -E[\Phi^2(\underline{x} \bullet \underline{m})\Phi^{'}(\underline{x} \bullet \underline{m})x_i] + E[\Phi^2(\underline{x} \bullet \underline{m})]E[\Phi(\underline{x} \bullet \underline{m})\Phi^{'}(\underline{x} \bullet \underline{m})x_i], \qquad (5.5)$$

where $\Phi^{'}(\underline{x} \bullet \underline{m})$ represents the derivative of the activation function at the point $\underline{x} \bullet \underline{m}$.

Equation 5.5 completes the formulation of the single neuron BCM network, and the minimization can then be applied to compute the weights leading to the minimum risk value. Of course, expected value operators present in Equations 5.4 and 5.5 are replaced by mean operators in practical applications. Further, note a random initial weight value is to be selected at the beginning of the training process.

(2) Multiple Neuron Case. As mentioned earlier, interaction between different neurons can be introduced when the BCM network has multiple neurons by introducing lateral inhibition between the PEs, as illustrated in Figure 5.3 [32]. In such a case, the output of a k^{th} PE is defined as:

$$\tilde{c}_k = c_k - \eta \sum_{j \neq k} c_j \,, \qquad (5.6)$$

where η is called the inhibition factor, and $c_k = \underline{x} \bullet \underline{m}_k$.

Now let us derive the partial derivative expression needed for the weight update equation. As was done earlier in the single neuron case, we first assume that the activation function is linear, and then take the nonlinearity into account later on. Thus, the risk function obtained for the k^{th} PE contained in the BCM network with lateral inhibition is obtained by replacing $\underline{x} \bullet \underline{m}$ by $\tilde{c}_k$ in Equation 5.1, which leads to:

$$R_k = -\frac{1}{3}E[\tilde{c}_k^3] + \frac{1}{4}E^2[\tilde{c}_k^2]. \tag{5.7}$$

Thus, the total risk expression R obtained for a BCM network with N neurons, is defined as the sum of the individual risk functions [32]:

$$R = \sum_{k=1}^{N} R_k. \tag{5.8}$$

Next, it can be shown that the partial derivative of the total risk R with respect to the weight values m_k is obtained as [32]:

$$\frac{\partial R}{\partial m_k} = E[(\tilde{c}_k^2 - \tilde{c}_k E[\tilde{c}_k^2])x] - \eta \sum_{j \neq k} E[(\tilde{c}_j^2 - \tilde{c}_j E[\tilde{c}_j^2])x]. \tag{5.9}$$

Note that the minimization of the total risk function given in Equation 5.8 is designed to produce neuron output values with distributions far from Gaussian.

Next, taking the activation function into account, the laterally inhibited output of the k^{th} neuron can be expressed as:

$$\tilde{c}_k = \Phi(c_k - \eta \sum_{j \neq k} c_j). \tag{5.10}$$

The partial derivative of the total risk function can be shown to be equal to [32]:

$$\frac{\partial R}{\partial m_k} = E[(\tilde{c}_k^2 - \tilde{c}_k E[\tilde{c}_k^2])\Phi'(\tilde{c}_k)x] - \eta\sum_{j\neq k} E[(\tilde{c}_j^2 - \tilde{c}_j E[\tilde{c}_j^2])\Phi'(\tilde{c}_j)x]. \qquad (5.11)$$

which completes the derivation of the N-neuron BCM network update equation.

d) Summary

The training stage used in a BCM-based classification scheme has two parts:

1- Training of the BCM network,

2- Training of the classifier using the BCM outputs.

This process is illustrated in Figure 5.5a. Once the classifier is trained, testing feature vectors are fed into the BCM network to reduce their dimensionality, and the resulting BCM outputs fed into a classifier to obtain class labels, as illustrated in Figure 5.5b.

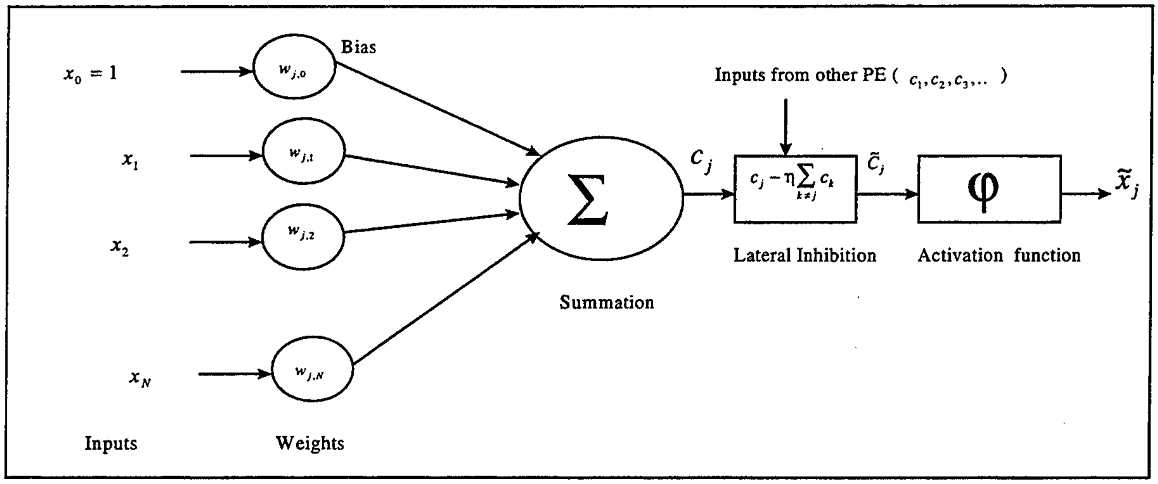

Figure 5.2: BCM neural network processing element (PE) diagram.

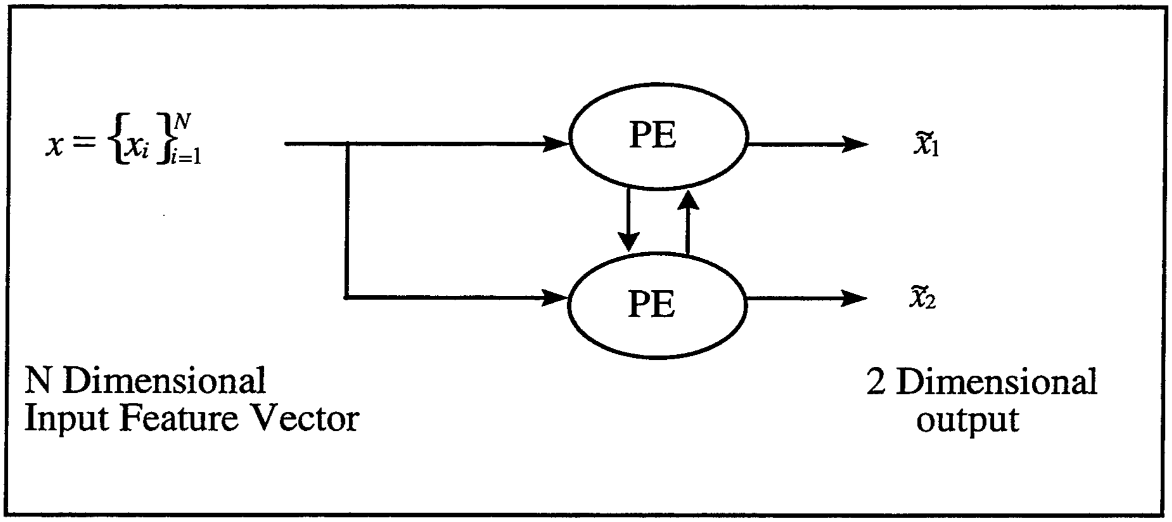

Figure 5.3: A sample BCM neural network with two PE's.

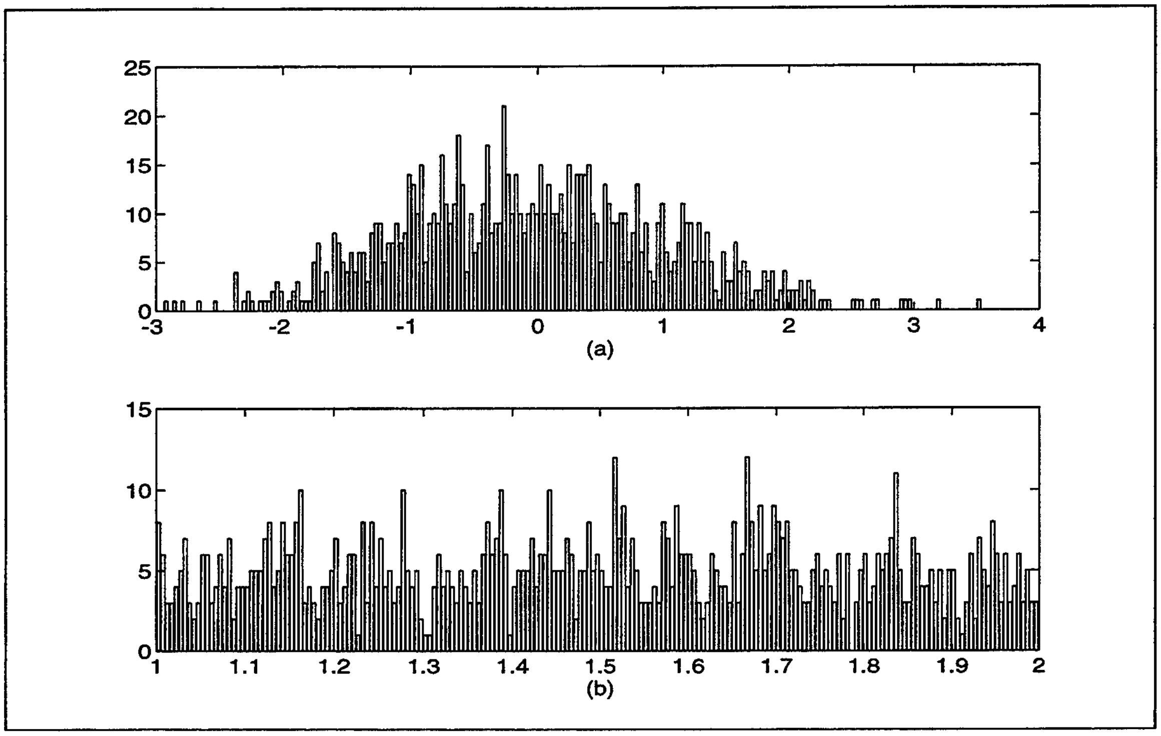

Figure 5.4: Data histograms for: (a) Gaussian data N(0,1), (b) Uniform data U(1,2) .

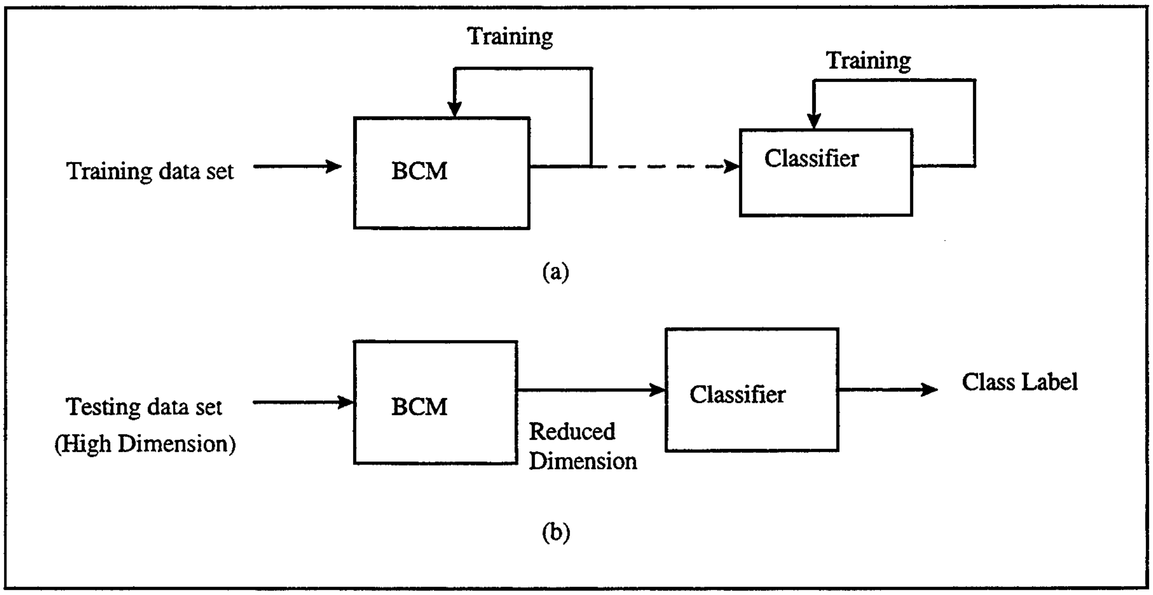

Figure 5.5: BCM based classification scheme: (a) Training phase, (b) Testing phase.

e) BCM-based Dimension Reduction Examples

(1) Dimension reduction from two to one. Consider two data classes which are represented by two features each. Assume that the class features are contained in the range [-1,0]*[0,2], and [0,1]*[0,1] respectively, as illustrated in Figure 5.6. Further, assume that there are 100 signals per signal class. Assume that we wish to reduce the dimension of the feature space from two to one, which requires one PE with one weight vector of length two. Initial weight values for the weight update equation are chosen equal to [1,1]. Using the training data set, the gradient value at the present weight location is computed using Equation 5.5 and the weights updated accordingly. Results show that the risk value converges to its minimum value after 20 iterations. Figure 5.7 shows the corresponding risk surface, risk contour and the trace of weight update during the training process. Note that the gradient value guided the process towards the minimum point of the risk surface. The upper plot in Figure 5.8 shows the values obtained for the BCM outputs as a function of the feature vector it was trained on. The first hundred feature vectors were selected from class one, while the next hundred from class two. This plot shows that the BCM output values obtained for feature vectors for class one and two are respectively centered around different values. Therefore, this example shows that the feature reduction operation preserved the class separability information, as the original 2D separate clusters are still separate in one dimension. The risk value obtained is equal to -0.2095. The lower plot in Figure 5.8 shows the histogram of the BCM output. Note that it is far from Gaussian.

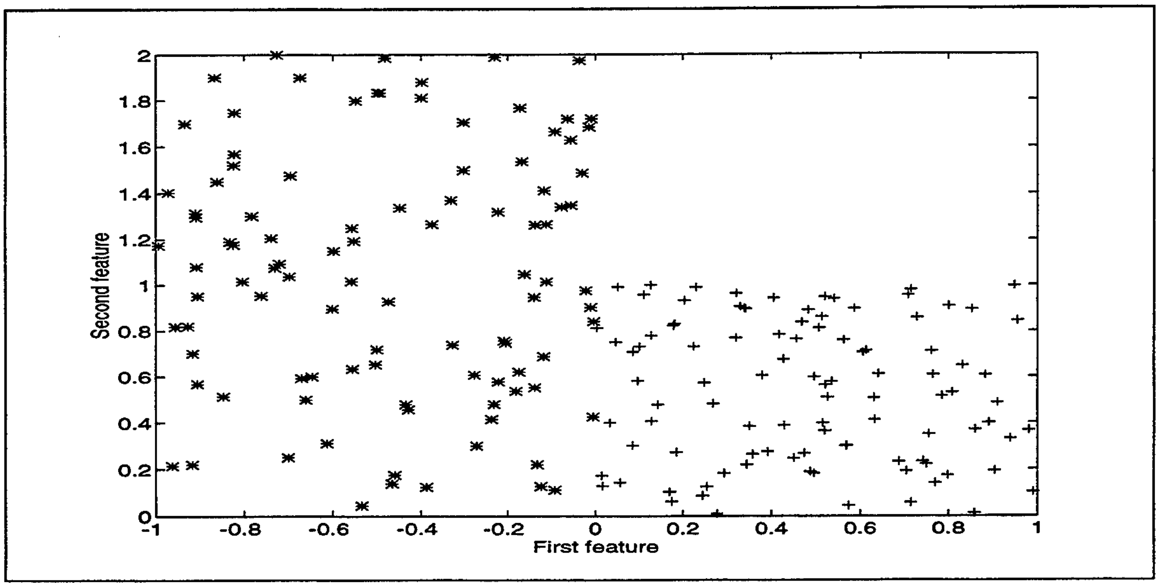

Figure 5.6: Cluster locations for the single neuron BCM implementation.

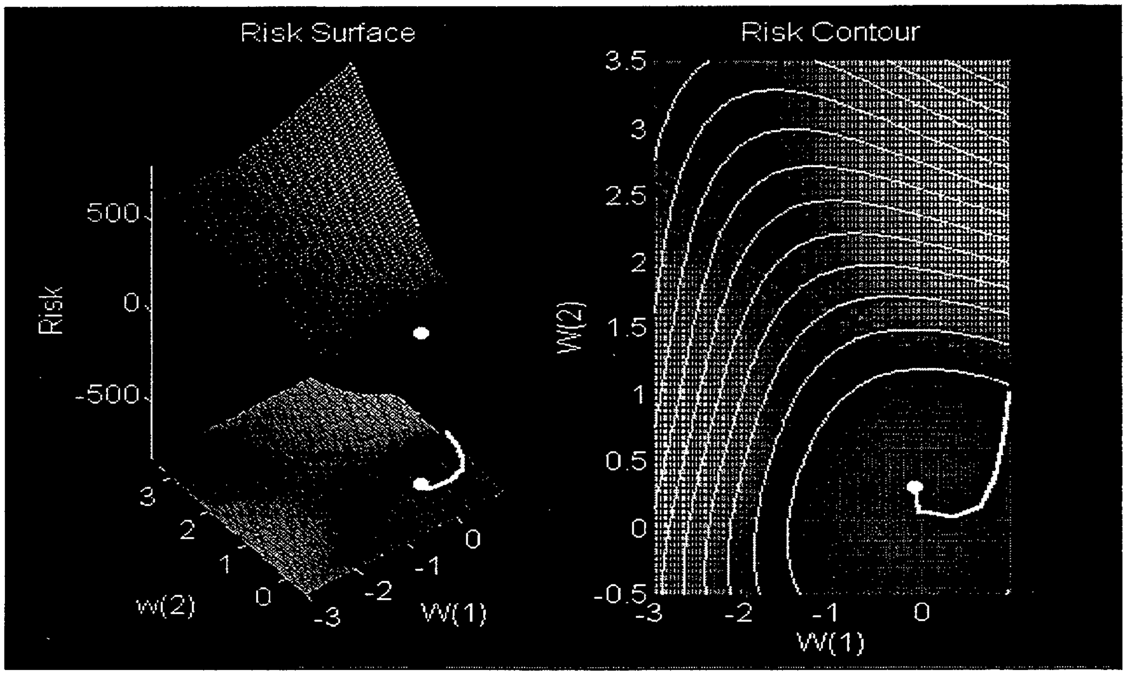

Figure 5.7: Risk surface, risk contour and trace of the weight update process.

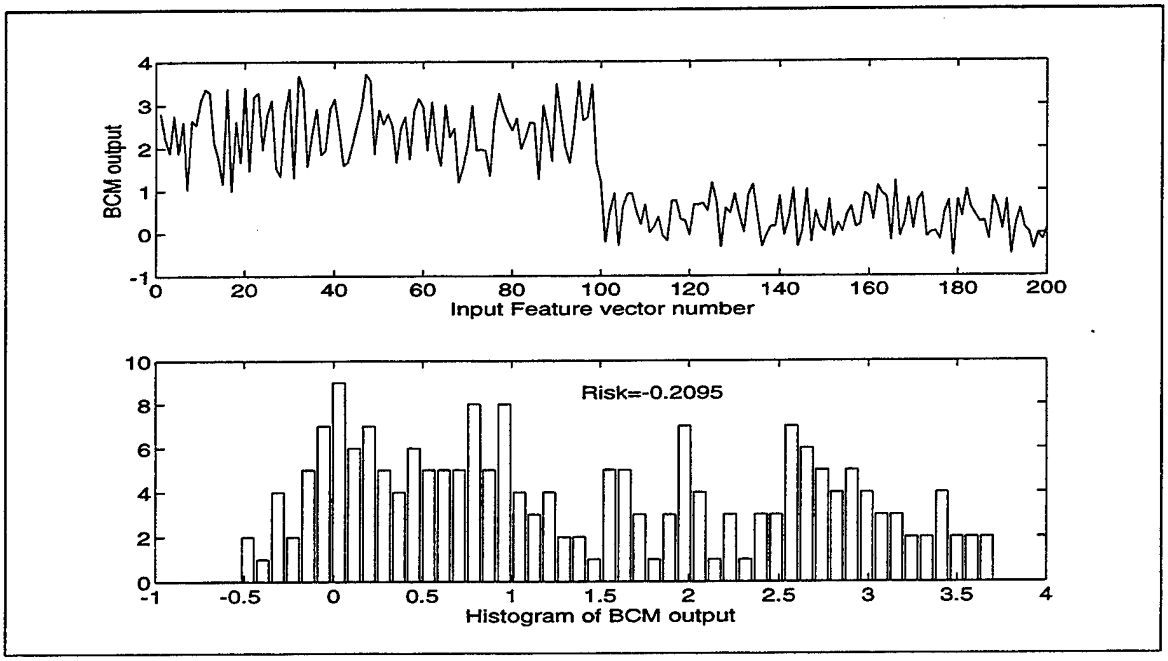

Figure 5.8: 2 Feature reduction using the BCM scheme; two-dimensional signal classes; BCM output values (top plot), output values histogram (bottom plot).

(2) Dimension reduction from three to two. Now consider the three signal clusters shown in Figure 5.9. Signals have three features so the original dimension is three. In this example, we reduce the dimension to 2 using a BCM configuration with 2 PEs. Thus, each PE has three weight components. Forty signals per signal class are considered during the training process, and the lateral inhibition factor 0.02 is selected. The total risk value converged to a minimum after 100 iterations.

Figure 5.10 plots the outputs of the BCM configuration for the given input features. Notice that we still have three separate clusters in 2D, meaning that the class separation information is still conveyed in the reduced dimension space. Figure 5.11 plots the histograms of each neuron output and shows the associated risk values.

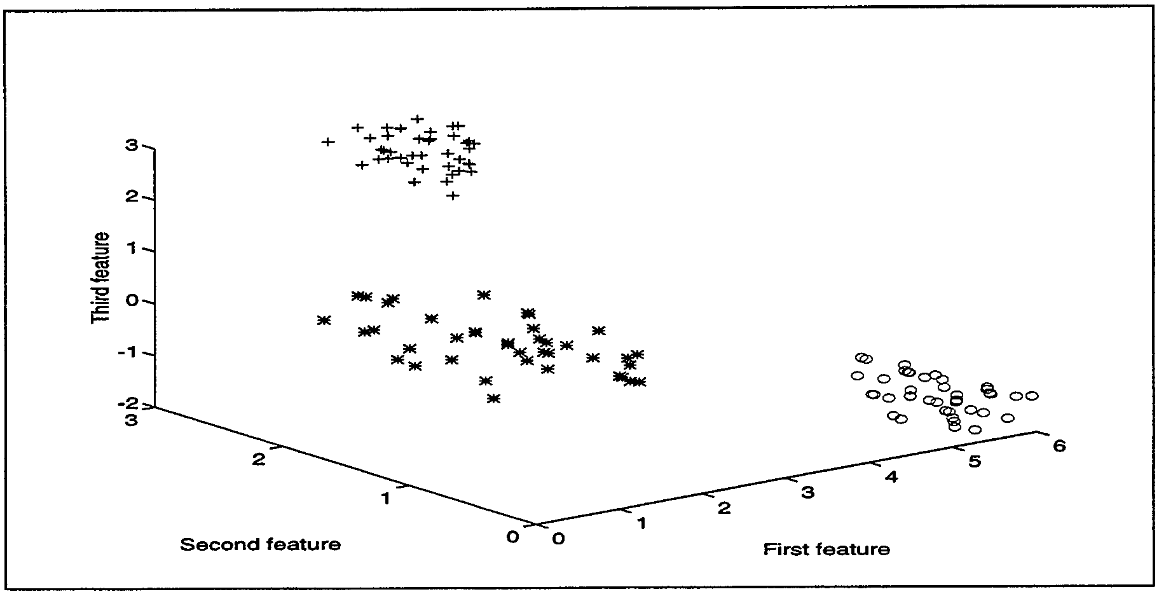

Figure 5.9: Cluster locations for the two-neuron BCM implementation.

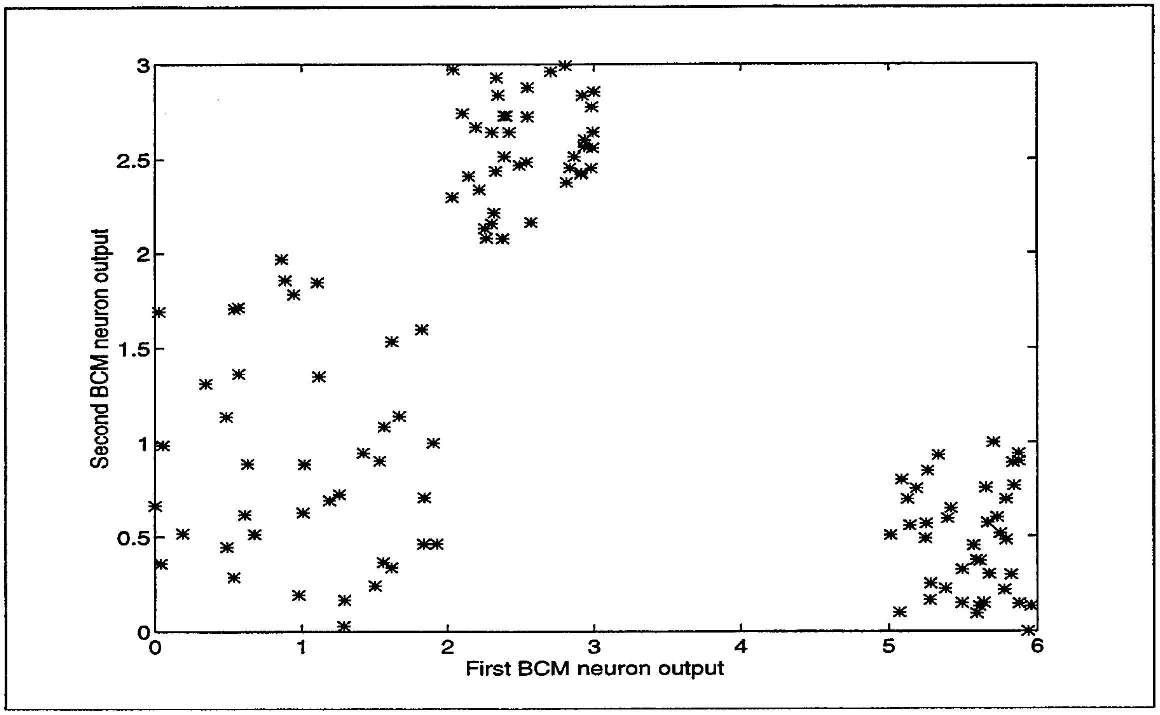

Figure 5.10: 2-neuron BCM output values, for data clusters given in Figure 5.9.

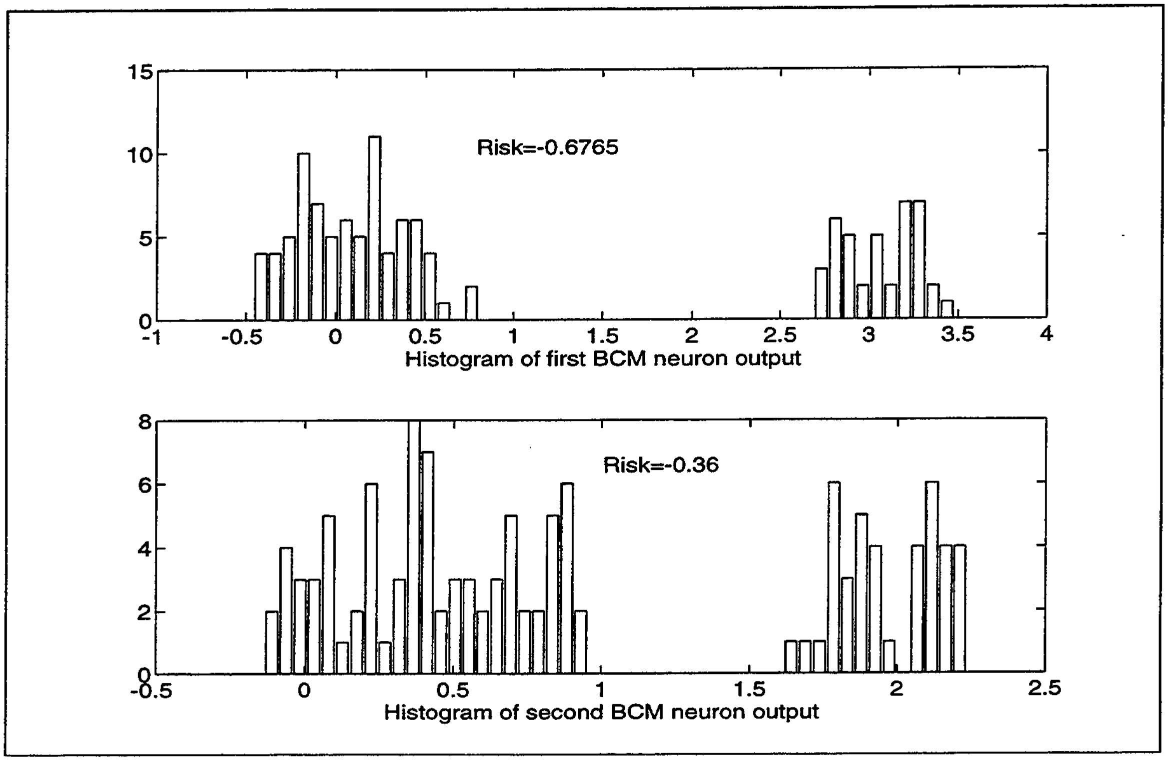

Figure 5.11: Histograms of the 2-neuron BCM output values.

f) BCM-Based Classification Scheme Example

In this example we illustrate the behavior of the BCM scheme used for feature reduction, the actual classification is done via BP as shown in Figure 5.5b. Two signals were considered: linear and quadratic chirps of length 512. Signal frequency characteristics were randomly changed 10% and white Gaussian noise was added to get a SNR of -5 dB. Forty training and 200 testing signals per signal class were selected. The Power method presented in Chapter III was chosen to extract the signal features. The first 8 scales were used, which resulted in 511 features per signal. Two classification schemes were considered, as shown in Figure 5.12. First, we considered a BP neural network with the configuration 511-100-2. Next, we considered a BCM network with configuration

511-10 (meaning the feature space dimension was reduced from 511 to 10, using a 10-neuron BCM network). Next, the reduced dimension features were used as input to a BP classifier of configuration 10-8-2. The Mathworks neural network toolbox package was used to implement the BP network [35]. The resulting classification rates obtained were 75% and 88% respectively.

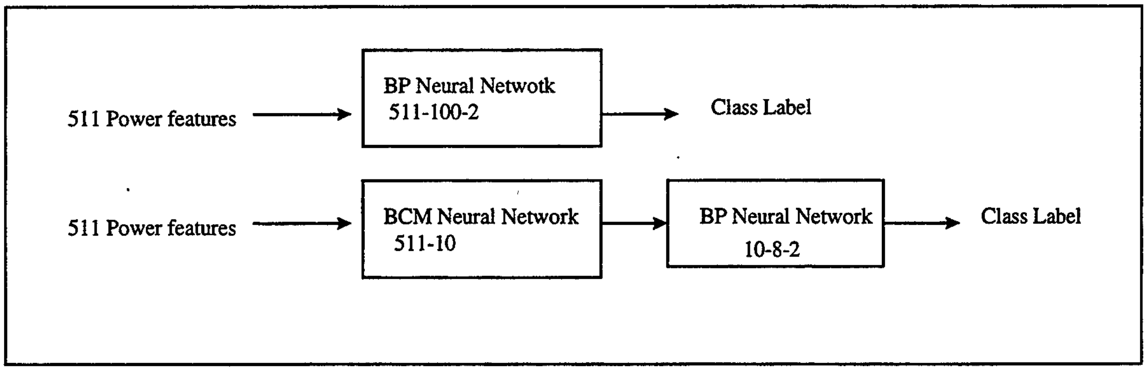

Figure 5.12: Two classification schemes considered in the first example.

g) BCM Drawbacks

The first problem encountered in the BCM is its slow convergence during training. For this reason, we adopted the variable learning rate with momentum algorithm in the weight equation update used during the BCM training phase [29,31]. Basically, this algorithm adjusts the learning rate according to the shape of the risk surface. Thus, the learning rate increases when the risk surface is smooth and decreases when the iteration occurs in a portion of the surface with a steep slope.

The second problem is the fact that the risk surface may have multiple minima due to the cubic expression in the risk equation. Therefore, the algorithm may stop at an undesirable local minimum, depending on the choice of the initial weight values. The likelihood of stopping at a local minimum may be decreased by running the scheme several times with different initial weight values, and selecting that which leads to the lowest risk value. However, running the scheme multiple times is expensive and does not guarantee the global minimum will be reached.

Finally, the third problem is the selection of a useful lateral inhibition factor which remains an open research area.

Next, we will consider a different projection pursuit algorithm scheme where projections that best discriminate between signal classes are considered as "interesting."

2. Mean Separator Neural Network

a) Concept

This particular neural network deals with one-dimensional projections that best separate two signal classes. One sample PE of this neural network is shown in Figure 5.13. Notice that it is identical to a PE used in a BP neural network. Assume that there are N training data sets per signal class, denoted as $x = \{\underline{x}_i\}_{i=1}^{i=N}$ and $y = \{\underline{y}_i\}_{i=1}^{i=N}$ respectively. We define the *mean-difference (MD)* projection index for this neural network as:

$$MD = -(E[\Phi(\underline{w} \bullet \underline{x})] - E[\Phi(\underline{w} \bullet \underline{y})])^2, \tag{5.12}$$

where Φ is the activation function and $\underline{w}$ is the weight vector.

In addition, the logsig function scaled between 10 and -10 is used as the activation function, as was considered earlier in the BCM setup. During the training process this projection index is minimized iteratively. The partial derivative of the projection index with respect to the weight vector is obtained as:

$$\frac{\partial MD}{\partial w_i} = -2(E[\Phi(\underline{w} \bullet \underline{x}] - E[\Phi(\underline{w} \bullet \underline{y})])(E[\Phi'(\underline{w} \bullet \underline{x})x_i] - E[\Phi'(\underline{w} \bullet \underline{y})y_i]). \quad (5.13)$$

Thus, the scheme minimizes the *MD* projection index given in Equation 5.12 in terms of the weight vector coefficients, as done in the BCM scheme. Note that each PE of this neural network can only be trained to distinguish between two signal classes. However, the following alternatives can be used in implementations dealing with more than two classes:

Alternative 1: Train each PE to distinguish one signal class from the rest (we call this alternative as the Class_x/Class non-x formulation). Then, there will *N* neurons in a *N*-signal class classification problem and outputs of these neurons can be fed into a classifier.

In addition, this network can also be used as a stand-alone classifier when followed by a decision scheme, as illustrated in Figure 5.14, even though it was originally designed for dimension reduction purposes only. For example, assume there are *N* signal classes and *N* neurons are trained using the Class_x/Class_non-x alternative. After training is completed, the training feature vectors belonging to each class are fed into the *N*-neuron *mean separator* network and the mean of each neuron output values computed. These *N* mean values constitute the reference values for each class and are denoted as r_i^n, where, 1<*i*< *N*. During testing, unknown feature vector $\underline{p}$ is fed into the *mean separator* network and its output values o_i computed. The distance c^n between the output values obtained for the testing feature vector and the reference output values obtained for each class is computed as:

$$c^n = \sum_{k=1}^{N} (r_k^n - o_k)^2 , \tag{5.14}$$

where 1<*n*< *N*.

Class labeling is obtained by selecting the class which leads to the smallest distance between reference and testing output values..

Alternative 2: Train each PE to distinguish two signal classes pairwise. For example, in a three-class case, the first neuron is trained to distinguish between classes 1 and 2, the second one trained to distinguish between classes 1 and 3, while the third neuron is trained to distinguish between classes 2 and 3. The number of resulting features obtained with this set-up is $\frac{N!}{2!(N-2)!}$, where *N* is the number of signal classes. Then, the output values may be fed into a classifier, or used as a stand-alone classifier with the decision scheme presented earlier. The main drawback with this set-up is the higher number of neurons needed to implement this approach.

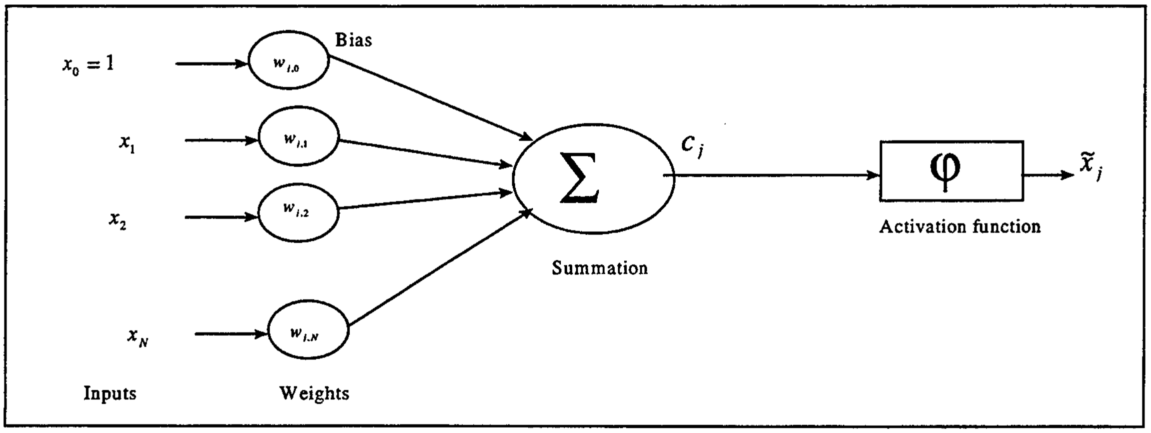

Figure 5.13: A sample PE of the mean separator neural network.

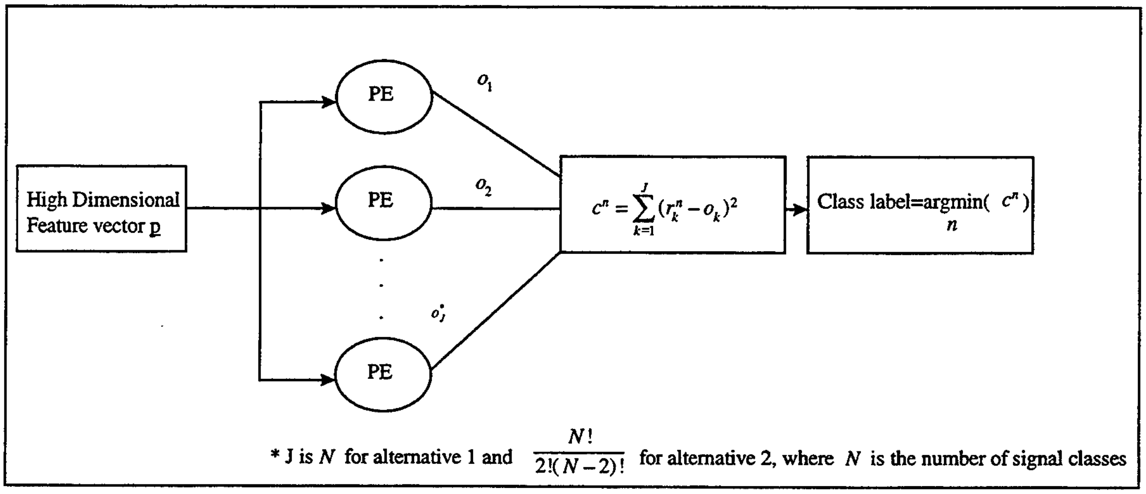
PE
o_1
PE
o_2
PE
o_J^*
High Dimensional Feature vector p
$c^n = \sum_{k=1}^{J} (r_k^n - o_k)^2$
Class label=argmin(c^n) n
* J is N for alternative 1 and $\frac{N!}{2!(N-2)!}$ for alternative 2, where N is the number of signal classes

Figure 5.14: One possible classification configuration with mean separator neural network.

b) Dimension Reduction Examples

The first example deals with the two clusters shown in Figure 5.15. Each feature vector contains two features and 40 signals per class. The weight update equation converged during the training phase after 30 iterations. Figure 5.16 plots the neuron output for the given training data set, where the first 40 values are obtained with signals belonging to the first class while the rest belong to the second class. This figure shows that the trained neuron has positive output values for the first signal class and negative values for the second class. Thus, unlabelled signals may be assigned to one of the two classes based on their output values.

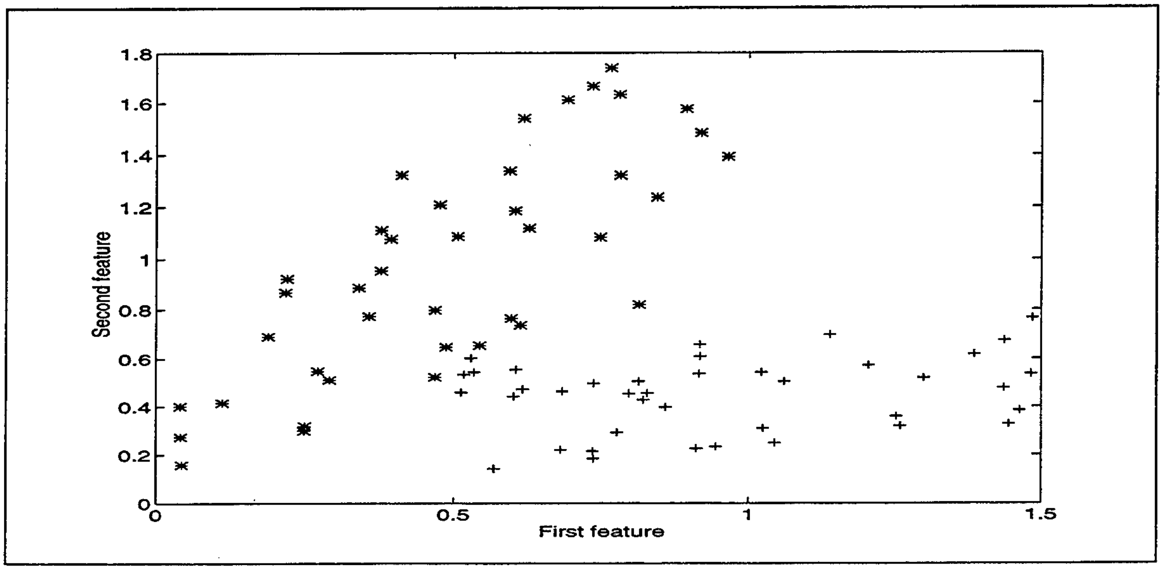

Figure 5.15: Two clusters in 2D.

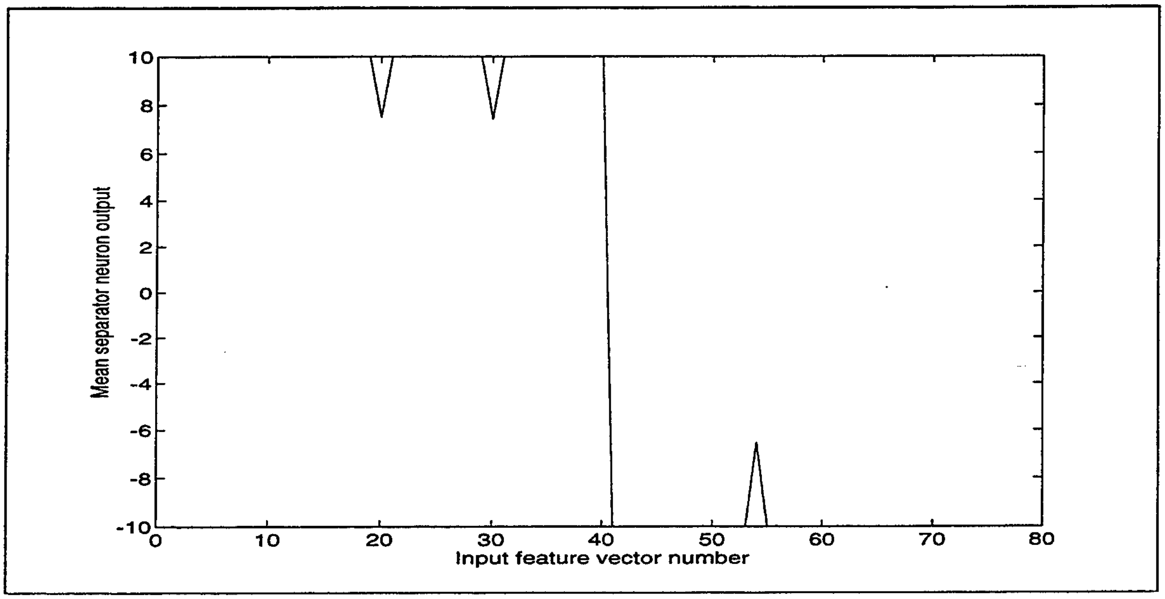

Figure 5.16: *Mean separator* neuron output for the clusters in Figure 5.15.

The second example illustrates the algorithm behavior when dealing with more than two classes and alternative-1 (Class x/Class non-x) is selected for feature reduction. Figure 5.17 plots three separate three-dimensional clusters that belong to three signal classes. Forty signals are used for training. Each of the three neurons is tuned to one single class, and trained using the training data set. (Note that in this example no dimension reduction is done, as there are three features before and after using the *mean separator*. However, this example was selected to get some insight on alternative-1 by visualizing the cluster locations and the resulting neuron output values.) Figure 5.18 plots the neuron output values obtained using the training data set after the training is completed. Note that each neuron gives a specific output when the signal that it was tuned to is presented. For example, the first neuron was tuned to the first signal class. The output values obtained from that signal class are equal to -10 while the output values obtained for the other 2 classes are equal to +10. Thus, this neuron can be used to distinguish class 1 from the rest. Similarly, the second neuron was tuned to distinguish class 2 from the rest. As a result the values obtained when presented class 2 signals are different from those obtained with class non-2 signals, thereby allowing to differentiate class 2 from class non-2. Similar comments hold for the third neuron, which was tuned to distinguish class 3 from the rest. These three neuron outputs can then be fed into a classifier, or can be used with a decision scheme, as illustrated in Figure 5.14.

Finally, notice the training process of the classification scheme is identical to that present in the BCM-based classification scheme when the *mean separator* network is used as a dimension reduction tool; first, the *mean separator* network is trained, next the output values are used in the training of a classifier, or with a decision scheme.

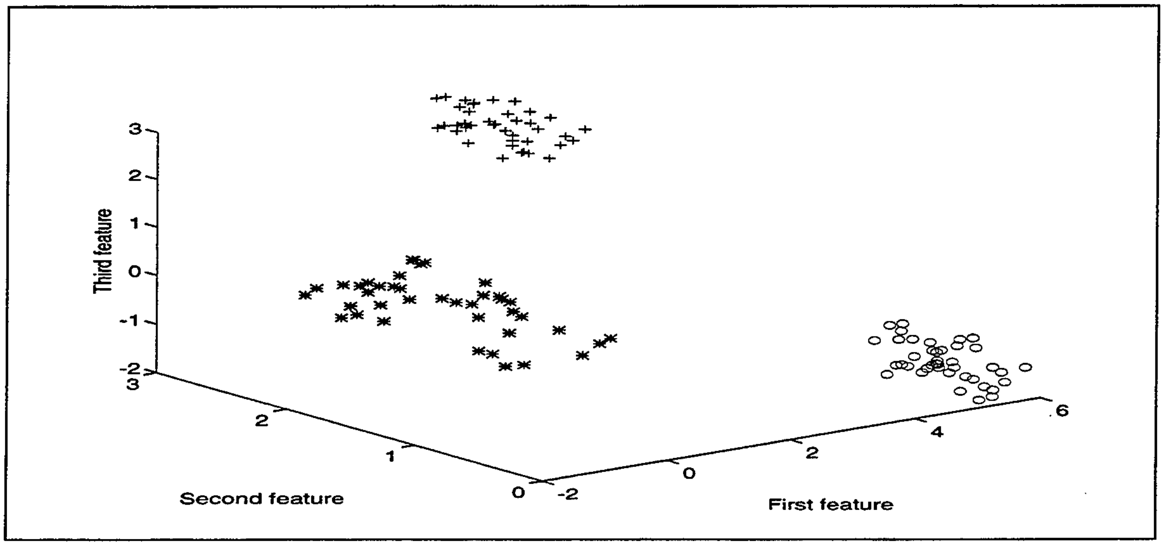

Figure 5.17: Three clusters..

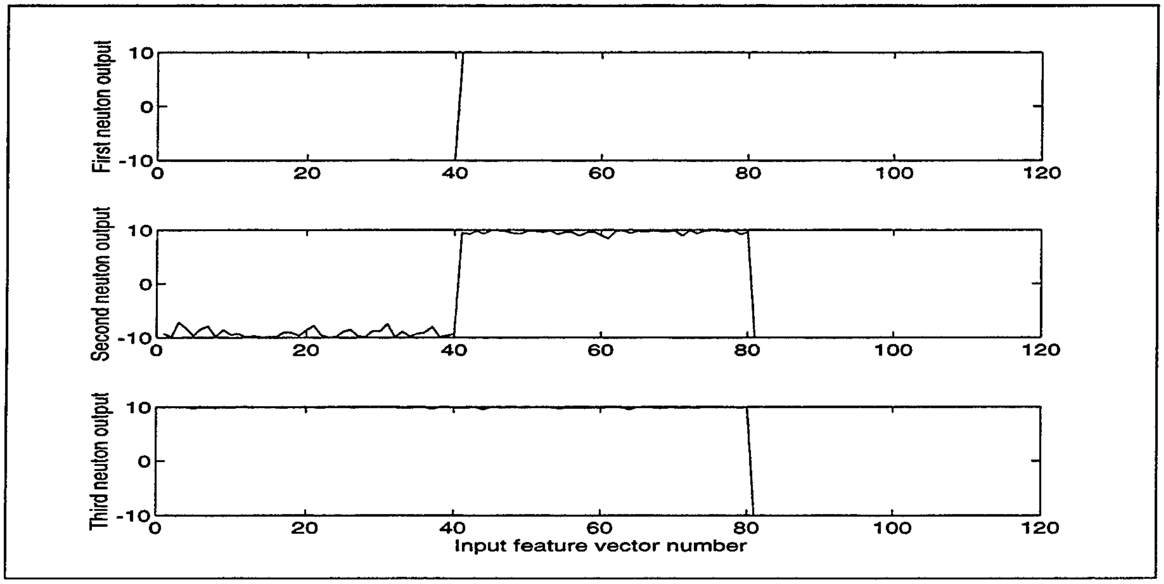

Figure 5.18: Three neuron outputs.

c) *Mean Separator Based Classification Scheme Examples*

(1) Two signal classes. This example considers linear chirp and quadratic chirp signal classes. Signal frequency characteristics were randomly altered 10% and the signal length was taken as 256. Forty signals per class were used for the training phase and 100 testing signals per class were used during the testing phase. White Gaussian noise was added to get a SNR of -5 dB. Initial class feature sets were obtained using the Power method described in Chapter III. The maximum scale selected was 7, resulting in 255 power features per signal. The following three classification schemes were considered (numbers in parenthesis indicate the classification rate):

1-Using the full size feature set and a BP neural network with the configuration 255-50-2 (82.2%),

2-Reducing the feature dimension from 255 to 50 using the *most discriminating* feature reduction scheme described in Chapter III. The resulting features were fed into a BP neural network of configuration 50-10-2 (83.8%),

3- Training a *mean separator* neuron to distinguish between the two signal classes. Class labeling was assigned according to the neuron output values, as illustrated in Figure 5.14. Note that this model doesn't use a BP NN (88%).

(a) Results

Confusion matrices obtained by averaging five trials are presented below for each scheme. Linear and quadratic classes are denoted as class 1 and 2 respectively. Recall that the confusion matrix diagonal elements show correct classification decisions expressed in percentage, while the off-diagonal elements show the incorrect classification decisions.

Scheme 1

Average Class. Rate: 82.2%			
		True Class Label	
		1	2
Declared as	1	88	23.6
Class	2	12	76.4

Scheme 2

Average Class. Rate: 83.8%			
		True Class Label	
		1	2
Declared as	1	81.2	13.6
Class	2	18.8	86.4

Scheme 3

Average Class. Rate: 88%			
		True Class Label	
		1	2
Declared as	1	86	10
Class	2	14	90

Results show that the best overall classification performance is obtained using the feature reduction followed by the decision step, which is actually the least expensive scheme to implement as no BP NN was used in the actual classification step.

(2) Five signal classes. The following five signal classes, previously used in Chapter III, were considered again: linear and quadratic chirps, doppler, high frequency sine and low frequency sine signal classes. The SNR level was set at -5dB. Forty training and 100 testing signals were used. Signal length was kept at 256. Signal frequency characteristics were randomly altered 10%. Again the Power method, with maximum scale equal to 7 was selected to extract the initial features, resulting in 255 power features per signal. Six classification scheme were tested:

1- A BP neural network using the full high-dimensional feature set, with configuration 255-50-5 (79%),

2-A combination of a feature reduction scheme (discussed in Chapter III), followed by a BP neural network. The feature reduction step selected the 50 *most discriminating features.* The resulting 50 features were fed into a BP neural network with configuration 50-10-5 (67%),

3- A combination of a *Mean Separator* neural network with 5 PEs followed by a BP neural network with configuration 5-5-5. Each PE contained in the *Mean Separator* NN was tuned to one signal class, following the Class-x/Class non-x scheme described earlier in Section 2a (81%),

4-A *mean separator* neural network with 5 PEs followed by the decision scheme to set class labels, as described in Section 2a and Figure 5.14 (81%), 5-A combination of a *Mean Separator* neural network with 10 PEs followed by a BP neural network with configuration 10-5-5. Each PE in the *Mean Separator* NN was trained to distinguish two signal classes pairwise, as described in Section 2b (84%),

6- A mean separator neural network with 10 PEs followed by the decision scheme to set class labels, as described earlier in Section 2b and Figure 5.14 (84%).

Confusion matrices obtained with each scheme by averaging five trials are presented below, where linear, quadratic, doppler, high frequency sine and low frequency sine signal classes are denoted as classes 1 to 5 respectively.

Scheme 1:		Average	Classif.	Rate:	79%	
			True	Class	Label	
		1	2	3	4	5
	1	58.8	17.6	1.8	0.2	3.8
Declared	2	17.8	55.4	9	0.6	0.4
as	3	4.4	21.2	88.4	0	0.2
Class	4	7.6	1.8	0.4	99	0
	5	11.4	4	0.4	0.2	95.6

Scheme 2: Average Classif. Rate: 67%						
		True Class Label				
		1	2	3	4	5
Declared as Class	1	24	7.8	0	5.2	8
	2	11.8	46.4	2.2	2	2.8
	3	11.4	35	97.4	1.2	0.4
	4	20.2	2.8	0.4	84.2	7.4
	5	32.4	8	0	7.4	81.4

Scheme 3: Average Classif. Rate: 81%						
		True Class Label				
		1	2	3	4	5
Declared as Class	1	75	21.6	3.4	6	9.8
	2	12.6	58.2	4.4	2.4	2.6
	3	1.8	17	92	0	0
	4	1.4	0	0.2	91.6	0
	5	9.2	3.2	0	0	87.6

Scheme 4:		Average	Classif.	Rate:	81%	
			True	Class	Label	
		1	2	3	4	5
	1	76	26.6	4.2	6.8	8.6
Declared	2	8.2	53	3	1.2	0.4
as	3	1.8	17.2	92.2	0	0
Class	4	1.8	0	0.2	91.4	0
	5	12.2	3.2	0.4	0.6	91

Scheme 5:		Average	Classif.	Rate:	84%	
			True	Class	Label	
		1	2	3	4	5
	1	78.6	12.8	1.6	6.4	13.4
Declared	2	11.6	74.6	7.2	1.2	1.2
as	3	2.6	6.6	90.6	0.4	0.2
Class	4	1.4	0	0.2	92	0
	5	5.8	2.4	0.4	0	85.2

Scheme 6:		Average	Classif.	Rate:	84%	
			True	Class	Label	
		1	2	3	4	5
	1	69.4	15	0.8	4.4	9.6
Declared	2	11	73.6	5.8	0.6	1
as	3	2.8	6.4	93.2	0.2	0
Class	4	8	1.6	0.2	94.8	0
	5	8.8	3.4	0	0	89.4

Results show that the best overall classification performance is obtained using schemes 5 and 6. Note that these schemes reduce the dimension of the input space from 255 to 10.

A few comments are in order.

1) The combination of the mean separator network and the BP NN results in a very fast training process, due to the low dimensionality of the input feature space.

2) The dimension reduction method selected in the second scheme didn't work well for this problem. Recall that in Chapter III, section B.2.c, we stated that the selected features may not have enough discriminating information for some of the signal classes, as this feature selection scheme uses averaged pair-wise relative entropy values defined in Equation 3.5. The confusion matrix for this scheme shows that the selected features do not contain discriminating information for the first class. As a result, the

classification rate for this class is only 24%, which degrades the overall classification rate.

3) The combination of the mean separator and the decision step considered in the fourth scheme outperformed the BP NN trained using the full high-dimensional feature set. Note that this scheme is very inexpensive, as class labeling is obtained using a simple decision scheme. The same type of comments holds for the sixth scheme investigated, however, this last scheme requires a higher number of PEs as it is based on pairwise feature discrimination.

d) Problems With the Mean Separator Neural Network

The first problem encountered during the implementations was slow convergence during the training process. Thus, we adopted the variable learning rate with momentum algorithm [31] in the weight update equation to alleviate this problem, as done in the BCM.

The second and maybe the most important problem is the existence of local minima in the optimization scheme. Thus, the algorithm may stop at an undesirable local minimum, depending on the choice of the initial weight values. As a result, users may choose to run the scheme several times with different initial weight values, and select that which leads to the lowest *MD* value. However, this solution is expensive and does not guarantee the global minimum will be reached.

VI. CLASSIFICATION RESULTS

The objective of this chapter is to apply the various classification tools (feature extractors, classifiers, dimension reducers) presented in previous mentioned chapters and to compare their performances when applied to synthetic and real world signals. First, we consider feature extraction schemes when used in connection with classifiers. Next, we consider dimension reduction tools on synthetic signals. Finally, we apply several combined classification schemes to real world underwater signals and compare their performances.

A. PERFORMANCE TESTS ON FEATURE EXTRACTION METHODS AND CLASSIFIERS

The performances of several classification schemes which combine feature extraction steps followed by a classifier are compared next. Five signal classes are considered: linear and quadratic chirp, doppler, high and low frequency sine signals, referred to as class 1 to 5 respectively. As done earlier, the signal frequency characteristics are altered 10% randomly to introduce some variability in the signal classes, and the signal length is kept at 256 samples. White gaussian noise is added to signal samples to get a SNR of -5 dB. Forty training and 100 testing signals per signal class are used in the implementations. The first 7 scales were selected, resulting in 255 power method features, and 256 LDB features. The following four classification schemes are tested:

1- 256 LDB features followed by a BP neural network with configuration 256-50-5 (98%),

2- 255 Power features followed by a BP neural network with configuration 255-50-5 (82%),

3-256 LDB features followed by a CT (57%),

4- 255 Power features followed by a CT (75%).

Trials were performed 5 times and averaged confusion matrices computed, leading to the following results:

Scheme 1:		Average	Classif.	Rate:	98%	
			True	Class	Label	
		1	2	3	4	5
Declared as Class	1	98.4	0.2	0	1	0.8
	2	0.2	99.4	0	0.4	0.8
	3	0.2	0.2	100	0.6	0.2
	4	0.4	0	0	95.6	0.4
	5	0.8	0.2	0	2.4	97.8

Scheme 2:		Average	Classif.	Rate:	82%	
			True	Class	Label	
		1	2	3	4	5
	1	82.4	31.2	6	4.2	5.2
Declared	2	8.2	49.2	5.4	0	0.2
as	3	2	15	88.6	0.2	0.2
Class	4	2.2	0.4	0	95.6	0
	5	5.2	4.2	0	0	94.4

Scheme 3:		Average	Classif.	Rate:	57%	
			True	Class	Label	
		1	2	3	4	5
	1	55.6	10.4	9.2	25.6	21.6
Declared	2	4.2	80.2	3.8	4	5.8
as	3	5	0.8	63.8	4	3.2
Class	4	20.8	5.8	13.2	51.2	33.4
	5	14.4	2.8	10	15.2	36

Scheme 4:		Average	Classif.	Rate:	75%	
			True	Class	Label	
		1	2	3	4	5
Declared as Class	1	6.6	44.8	3	2	3.8
	2	15.2	46	18.8	1.2	2.2
	3	3.8	26	78.2	1.6	0.2
	4	4.4	0.6	0	95.2	0
	5	10.6	2.6	0	0	93.8

A few comments are in order:

1-Results show that the BP neural network outperformed the CT with both feature extraction methods. This result is to be expected as the CT tries to partition signal clusters with perpendicular lines while the NN has no such constraint.

2-The CT has better performance when using Power method features than LDB features, while the opposite is true for the BP NN. Sample CTs obtained for the third and fourth schemes are shown in Figures 6.1 and 6.2 respectively. Note that the CT in Figure 6.1 is far more complex than that of Figure 6.2, which may indicate that the class clusters obtained using the LDB are not as well matched as those obtained using the Power method for tree partitioning. In general, we noted that classification rates tended to be lower when the associated CT were complex.

3- The best performance was obtained when using the LDB method followed by a BP neural network. However, note that one may hardly expect ideal behavior with real world signals, where one would have to deal with time synchronization issues, etc. When random time shifts between 0 to 100 sample points are introduced in the signals to be classified, the results change drastically, leading to:

Scheme 1:		Average	Classif.	Rate:	47%	
			True	Class	Label	
		1	2	3	4	5
	1	40	30.2	11.2	15.8	12.2
Declared	2	23	37.8	16.6	8.4	14.4
as	3	13.2	16.4	51.2	9.4	20.2
Class	4	11.4	6.2	10.4	60.8	8.8
	5	12.4	9.4	10.6	5.6	44.4

Scheme 2:		Average	Classif.	Rate:	60%	
			True	Class	Label	
		1	2	3	4	5
	1	35.6	15	14.4	4.4	7.6
Declared	2	19.6	38	9.4	2.6	7.4
as	3	28.4	38.2	68.8	7.6	14.6
Class	4	6	2	3.2	84.8	1.8
	5	10.4	6.8	4.2	0.6	68.6

Scheme 3:		Average	Classif.	Rate:	24%	
			True	Class	Label	
		1	2	3	4	5
	1	17	17.8	16.6	18.6	18
Declared	2	19.8	19.8	18	25.2	23.4
as	3	21.8	26	32.6	15.6	19.2
Class	4	17.8	19.2	14.6	22.8	14.4
	5	23.6	17.2	18.2	17.8	25

Scheme 4:		Average	Classif.	Rate:	43%	
			True	Class	Label	
		1	2	3	4	5
	1	19.6	15.8	20.8	14.2	14.2
Declared	2	40.2	40.6	25.4	18	14.4
as	3	26	34.6	42.4	6	15.6
Class	4	2.8	1.2	1.6	60.2	1.2
	5	11.4	7.8	9.8	1.6	54.6

Such degradations are to be expected. However, results also show that the Power method is more robust to time shifts than the LDB method is. Thus, one may expect the Power method followed by a BP NN to outperform schemes involving LDB and CTs in real world problems where time shifts may occur. We will further test this classification scheme along with the others at the end of this chapter on real world underwater signals.

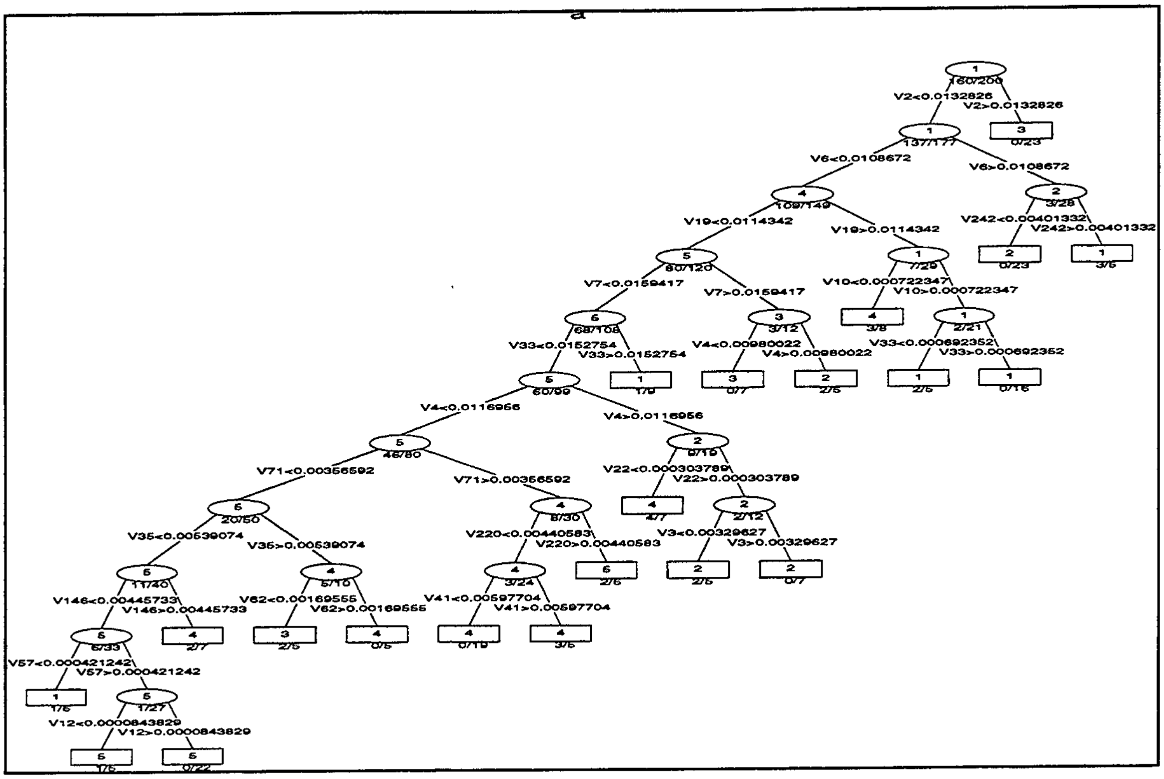

Figure 6.1: One sample CT for the LDB + CT classification scheme.

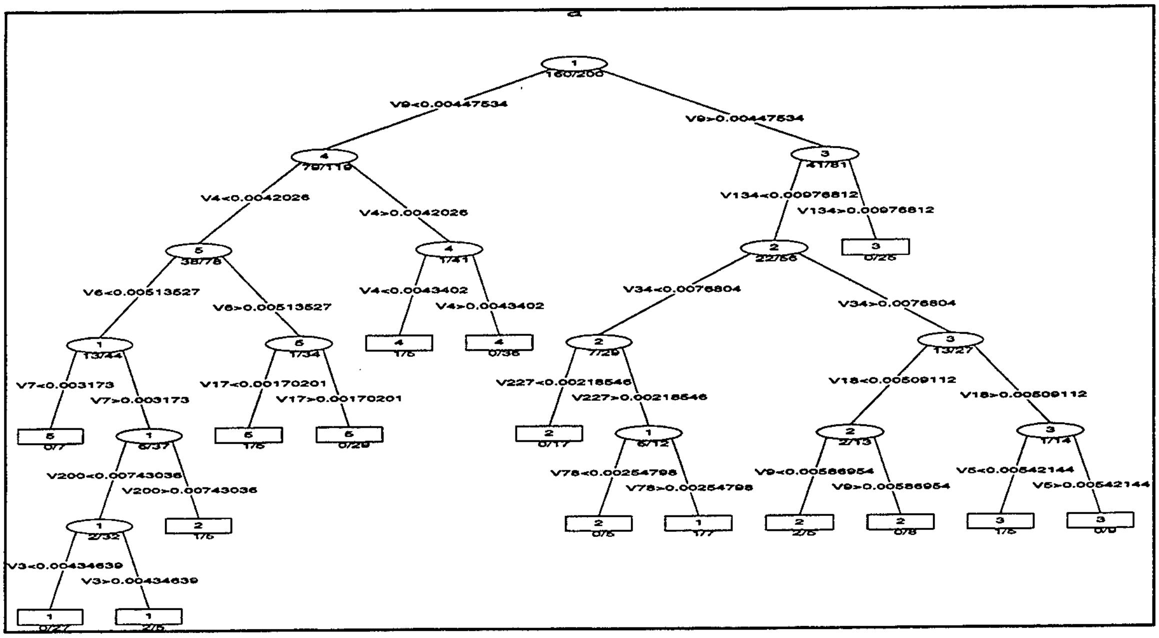

Figure 6.2: One sample CT for the Power + CT classification scheme.

B. DIMENSION REDUCTION EXPERIMENTS

This section considers dimension reduction issues and investigates whether they are useful when using CTs. Next, we test dimension reduction schemes on synthetic signals.

1. Dimension Reduction Issues with CT

As mentioned in Chapter IV, the CT growing process involves the selection of "best questions" to partition the data, and as a result, to extract a small number of features which are used in the classification process while the rest is simply disregarded. Actually, this partitioning process itself can also be viewed as some type of dimension reduction scheme. Thus, there is no need to reduce the dimension of the features prior to using the CT as long as the CT growing process preserves all the class information.

This comment was illustrated on the five signal class example used earlier (no time shifts are added to the data). The signal length is kept at 512 samples, and the maximum scale selected is 8, leading to 512 LDB and 511 Power method features. We first considered using Power method features followed a CT in the following four classification schemes:

1- 511 (all) Power features followed by a CT (91%),

2- 250 *most discriminating* Power features followed by a CT (86%),

3- 100 *most discriminating* Power features followed by a CT (74%),

4- 50 *most discriminating* Power features followed by a CT (63%).

Five trials were performed, and average confusion matrices computed, leading to the following results:

Scheme 1:		Average	Classif.	Rate:	91%	
			True	Class	Label	
		1	2	3	4	5
	1	87	18.2	0	0.4	3.8
Declared	2	7.8	79.4	3.8	0	0.4
as	3	0	1.6	96.2	0	0
Class	4	1.6	0	0	99.6	0
	5	3.6	0.8	0	0	95.8

Scheme 2:		Average	Classif.	Rate:	86%	
			True	Class	Label	
		1	2	3	4	5
	1	75.2	17	0	6.4	7.8
Declared	2	7.8	79.4	3.8	0	0.4
as	3	0	1.6	96.2	0	0
Class	4	14.4	2	0	93.6	4.6
	5	2.6	0	0	0	87.2

Scheme 3:		Average	Classif.	Rate:	74%	
			True	Class	Label	
		1	2	3	4	5
	1	62.4	18.8	0	10.2	12.4
Declared	2	7.8	79.4	3.8	0	0.4
as	3	0	1.6	96.2	0	0
Class	4	9.2	0	0	64.6	21.8
	5	20.6	0.2	0	25.2	65.4

Scheme 4:		Average	Classif.	Rate:	63%	
			True	Class	Label	
		1	2	3	4	5
	1	59	19.2	0	25	33.4
Declared	2	15.2	77.8	5.8	3.8	3
as	3	0	1.8	94.2	0	0
Class	4	10	1	0	42.2	22.4
	5	15.8	0.2	0	29	41.2

A few comments are in order:

1) Results show that the best performance was obtained when no dimension reduction was performed prior to the CT step, which indicates that the CT natural dimension reduction process is effective. The poor performance obtained with schemes 2, 3 and 4 may be due to the specific *most discriminating* dimension reduction method considered in these schemes. Better results might have been obtained with another dimension reduction scheme, but the classification rate obtained without doing any dimension reduction (using all features with CT) is high and relying on CT natural dimension reduction process seems sufficient at this point.

Second, the same experiment was performed using the LDB feature extraction method followed by a CT. The following four classification schemes were considered :

1- 512 LDB features followed by a CT (61%),

2- 250 *most discriminating* LDB features followed by a CT (61%),

3- 100 *most discriminating* LDB features followed by a CT (61%),

4- 50 *most discriminating* LDB features followed by a CT (61%).

Average confusion matrices were computed using five trials, leading to:

Scheme 1: Average Classif. Rate: 61%						
		True Class Label				
		1	2	3	4	5
Declared as Class	1	63	20.6	17.2	20	19.8
	2	3.4	76.2	2	1.6	2.2
	3	1.8	0.2	75.8	2.2	2.2
	4	15	1.2	1.4	38.4	25
	5	16.8	1.8	3.6	37.8	50.8

Scheme 2: Average Classif. Rate: 61%						
		True Class Label				
		1	2	3	4	5
Declared as Class	1	63.8	20.6	18.2	21	21
	2	3.4	76.2	2	1.6	2.2
	3	2.2	0.2	75	2.6	2.6
	4	15.2	1	1.4	37.6	24.6
	5	15.4	2	3.4	37.2	49.6

Scheme 3:		Average Classif. Rate: 61%				
		True Class Label				
		1	2	3	4	5
	1	72	20.4	15.8	27.8	25
Declared	2	3.4	76.2	2	1.6	2.2
as	3	1.8	0.2	78.2	2.8	2
Class	4	12.8	2	1.6	32.8	27.2
	5	10	1.2	2.4	35	43.6

Scheme 4:		Average Classif. Rate: 61%				
		True Class Label				
		1	2	3	4	5
	1	72	20.4	15.8	27.8	25
Declared	2	3.4	76.2	2	1.6	2.2
as	3	1.8	0.2	78.2	2.8	2
Class	4	12.8	2	1.6	32.8	27.2
	5	10	1.2	2.4	35	43.6

Results show lower classification rates than those obtained using the Power method. This is to be expected as the averaging operation present in the Power method makes it more robust to in-class signal variations. Results also show that same performances are obtained with the full set of LDB features or a smaller number selected using the *most discriminating* LDB scheme.

Thus, these results illustrate the fact that the CT has its own powerful "dimension reduction" process which makes using additional dimension reduction schemes unnecessary at this point. As a result, we select the BP neural network as classifier type when testing the performances of dimension reduction methods on LDB and Power methods.

2. Performance Tests on Dimension Reduction Tools

First, we will investigate the performances of the dimension reduction schemes on the LDB feature extraction method. The same five signal class example as used earlier is considered again. Signal length is set at 256 samples, and the SNR was kept at -5 dB. The first 7 scales were selected resulting in 256 LDB features per signal sample. Forty training and 100 testing signals were used. The following eight classification schemes were implemented:

1- 256 LDB features followed by a BP neural network with configuration 256-50-5 (97%),

2-100 *most discriminating* LDB features followed by a BP neural network with configuration 100-20-5 (96%),

3-50 *most discriminating* LDB features followed by a BP neural network with configuration 50-10-5 (88%),

4-20 *most discriminating* LDB features followed by a BP neural network with configuration 20-5-5 (73%),

5-A combination of a *Mean Separator* neural network with 5 PEs followed by a BP neural network with configuration 5-5-5. Each PE contained in the *Mean Separator* NN was tuned to one signal class, following the Class-x/Class non-x scheme described earlier in Chapter V, Section 2a (97%),

6-A *mean separator* neural network with 5 PEs followed by the decision scheme to set class labels, as described in Chapter V, Section 2a and Figure 5.14 (98%),

7-A combination of a *Mean Separator* neural network with 10 PEs followed by a BP neural network with configuration 10-5-5. Each PE in the *Mean Separator* NN was trained to distinguish two signal classes pairwise, as described in Chapter V, Section 2b (95%),

8-A mean separator neural network with 10 PEs followed by the decision scheme to set class labels, as described earlier in Chapter V, Section 2b and Figure 5.14 (96%).

After 5 trials confusion matrices were computed. The results are:

Scheme 1:		Average	Classif.	Rate:	97%	
			True	Class	Label	
		1	2	3	4	5
	1	97.6	0	0.2	2.2	0.6
Declared	2	0.2	98.8	0	2	0.4
as	3	0.6	0.2	98.6	0.6	0
Class	4	1	0	0.2	94	1.2
	5	0.6	1	1	1.2	97.8

Scheme 2:		Average	Classif.	Rate:	96%	
			True	Class	Label	
		1	2	3	4	5
	1	97.8	0	0	3.6	4.2
Declared	2	0	99.6	0	1	2.8
as	3	0.2	0	100	0.8	0.6
Class	4	0.6	0.2	0	92.8	1.6
	5	1.4	0.2	0	1.8	90.8

Scheme 3:		Average	Classif.	Rate:	88%	
			True	Class	Label	
		1	2	3	4	5
	1	90.8	0	0.4	3.8	12.4
Declared	2	0.2	98.4	0	1.2	5.2
as	3	0.8	0.6	99.4	0.4	1.8
Class	4	5	0.4	0.2	89.8	17
	5	3.2	0.6	0	4.8	63.6

Scheme 4:		Average	Classif.	Rate:	73%	
			True	Class	Label	
		1	2	3	4	5
	1	77.8	0.2	0	12.6	33.2
Declared	2	1.8	97.4	0.8	3	14.4
as	3	2	0.6	99.8	0.8	1.2
Class	4	16.4	1	0.2	82.8	43.2
	5	2	0.8	0	0.8	8

Scheme 5:		Average	Classif.	Rate:	97%	
			True	Class	Label	
		1	2	3	4	5
	1	95.8	0.2	0	1	2.4
Declared	2	2.4	98.6	1	3.6	3
as	3	0.4	0.6	99	0	0
Class	4	0.8	0	0	95.4	0.2
	5	0.6	0.6	0	0	94.4

Scheme 6:		Average	Classif.	Rate:	98%	
			True	Class	Label	
		1	2	3	4	5
	1	97	0.8	0.4	2.2	1
Declared	2	0.8	98	0	0	0.8
as	3	0.4	0.4	99.2	0	0.2
Class	4	0.6	0	0	97.6	0.2
	5	1.2	0.8	0.4	0.2	97.8

Scheme 7:		Average	Classif.	Rate:	95%	
			True	Class	Label	
		1	2	3	4	5
	1	96.6	2.2	0.2	0.6	3
Declared	2	0.6	93.4	0	0.2	3.6
as	3	1.8	0.4	96.8	0.6	1.8
Class	4	0.8	3	2.8	98.4	2.6
	5	0.2	1	0.2	0.2	89

Scheme 8:		Average	Classif.	Rate:	96%	
			True	Class	Label	
		1	2	3	4	5
	1	94.8	0.6	0	0.4	2.6
Declared	2	0.6	97	0	0.6	3.6
as	3	1.8	0.4	99	0.6	1.6
Class	4	2.6	2	0.8	98.2	0.6
	5	0.2	0	0.2	0.2	91.6

A few comments are in order;

1-Results show that the *mean separator* based classification schemes have good performances. In addition, the mean separator feature reduction scheme significantly decreased the BP NN training time required in schemes 5 and 7.

2-Schemes 6 and 8 are the computationally cheapest ones, as they use the decision scheme.

3-Schemes 1 to 4 show that classification performances decrease as the number of LDB features kept decreases. This is to be expected as such feature reduction steps may result in information loss, causing degradations in the classifier performances.

4-Finally, we also implemented the BCM followed by a BP neural network classification scheme. However, a suitable lateral inhibition factor couldn't be isolated and results were much worse than those shown here (62%).

Next, we consider the Power method dimension reduction schemes mentioned in Chapter III; *Learned and Willsky's, most consistent, most discriminating nodes,* and the LDB based dimension reduction schemes. The maximum scale selected was 7 resulting in 255 power features per signal. The SNR was chosen as -5 dB. The following five classification schemes were implemented for comparison:

1- 255 power features followed by a BP neural network with configuration 255-50-5 (83%),

2- 16 power features selected by the *Learned and Willsky's* dimension reduction scheme followed by a BP neural network with configuration 16-5-5 (56%), 3-50 power features selected by the *most discriminating nodes* dimension reduction scheme followed by a BP neural network with configuration 50-10-5 (70%),

4- 50 power features selected by the *most consistent nodes* dimension reduction scheme followed by a BP neural network with configuration 50-10-5 (76%),

5- 31 power features selected by the LDB based dimension reduction scheme followed by a BP neural network with configuration 31-6-5 (82%).

Five trials were performed and the resulting average confusion matrices computed, leading to:

Scheme 1: Average Classif. Rate: 83%						
		True Class Label				
		1	2	3	4	5
	1	77.6	25.8	4.8	2	3.4
Declared	2	11	55.6	5.8	0.4	0.8
as	3	4	15.2	89	0.2	0.6
Class	4	2.8	0.6	0.2	97.4	0.2
	5	4.6	2.8	0.2	0	95

Scheme 2:		Average	Classif.	Rate:	56%	
			True	Class	Label	
		1	2	3	4	5
	1	17.4	7.4	0.4	5	8.2
Declared	2	14.6	38.2	6.6	4	3.8
as	3	14.6	31.4	91.4	2.4	3.6
Class	4	26.6	13	1.4	76.4	30
	5	26.8	10	0.2	12.2	54.4

Scheme 3:		Average	Classif.	Rate:	70%	
			True	Class	Label	
		1	2	3	4	5
	1	27.4	4	0.2	9	10
Declared	2	22	77.8	10	3.4	5.6
as	3	10.6	12.2	89.2	0.8	1.6
Class	4	14.6	1.4	0.2	78.6	3.4
	5	25.4	4.6	0.4	8.2	79.4

Scheme 4: Average Classif. Rate: 76%						
		True Class Label				
		1	2	3	4	5
Declared as Class	1	83.6	39.8	4.8	1.2	9.2
	2	4.4	17.8	4.4	0	0.4
	3	1.4	36	90.2	0	0
	4	2	1.2	0.4	98.8	0.2
	5	8.6	6.8	0.2	0	90.2

Scheme 5: Average Classif. Rate: 82%						
		True Class Label				
		1	2	3	4	5
Declared as Class	1	77.4	29	2.8	1.4	0.2
	2	10	43.8	6.4	0.6	0.4
	3	3.4	24.6	90.4	0.4	0.2
	4	3.4	0.4	0.2	97.2	0
	5	5.8	2.2	0.2	0.4	99.2

Summary

Results showed that the LDB based dimension reduction scheme used in connection with the Power feature extraction method gives good classification results and that the results are similar to those obtained using the full set of power features.

C. APPLICATIONS OF CLASSIFICATION SCHEMES TO UNDERWATER SIGNALS

This section investigates the application of the various classification schemes described earlier to five real-world underwater signals: gray whale, humpback whale, killer whale, sperm whale and underwater earthquake. These experiments use 2 or 3 recordings of average length 40000 per signal class. Figure 6.3 plots a section of one recording for each signal class. Figures 6.4 through 6.8 plot the associated spectrograms. One hundred training and 129 testing sets were obtained by segmenting the data in successive nonoverlapping segments of length 512. No attempt at time synchronization was made. We compared several classification performances obtained using the Power feature extraction and LDB schemes followed by various classification schemes.

1. Power Feature Extraction Scheme

We consider the following 7 classification methods based on the Power feature extraction scheme:

1-255 power features followed by a CT (74%),

2- 255 power features followed by a BP neural network with configuration 255-50-5 (95%), 3-A combination of a *Mean Separator* neural network with 5 PEs followed by a BP neural network with configuration 5-5-5. Each PE contained in the *Mean Separator* NN was tuned to one signal class, following the Class-x/Class non-x scheme described earlier in Chapter V, Section 2a (90%),

4-A *mean separator* neural network with 5 PEs followed by the decision scheme to set class labels, as described in Chapter V, Section 2a and Figure 5.14 (87%),

5-A combination of a *Mean Separator* neural network with 10 PEs followed by a BP neural network with configuration 10-5-5. Each PE in the *Mean Separator* NN was trained to distinguish two signal classes pairwise, as described in Chapter V, Section 2b (92%),

6-A mean separator neural network with 10 PEs followed by the decision scheme to set class labels, as described earlier in Chapter V, Section 2b and Figure 5.14 (92%),

7- 44 power features selected by the LDB based dimension reduction scheme followed by a BP neural network with configuration 44-6-5 (94%).

The resulting confusion matrices obtained are:

Scheme 1:		Average	Classif.	Rate:	74%	
			True	Class	Label	
		1	2	3	4	5
	1	81.4	0	96.9	0	0
Declared	2	0	86.05	0	0	0
as	3	18.6	0	1.55	0	0
Class	4	0	0	1.55	100	0
	5	0	13.95	0	0	100

Scheme 2:		Average	Classif.	Rate:	95%	
			True	Class	Label	
		1	2	3	4	5
	1	92.25	0	12.4	0	0
Declared	2	0.78	98.45	0	0	0
as	3	6.98	0	86.82	1.55	0
Class	4	0	0	0.78	98.45	0
	5	0	1.55	0	0	100

Scheme 3:		Average Classif. Rate: 90%				
		True Class Label				
		1	2	3	4	5
Declared as Class	1	93.02	0	14.73	1.55	0
	2	0.78	83.72	0	0	0
	3	5.43	0	75.19	1.55	0
	4	0.78	0	10.08	96.9	0
	5	0	16.28	0	0	100

Scheme 4:		Average Classif. Rate: 87%				
		True Class Label				
		1	2	3	4	5
Declared as Class	1	82.17	0	8.5	0.77	0
	2	0.77	88.37	0	0	0
	3	15.5	0	65.91	0.77	0
	4	0.77	9.3	25.58	98.45	0
	5	0.77	2.32	0	0	100

Scheme 5		Average	Classif.	Rate:	92%	
			True	Class	Label	
		1	2	3	4	5
	1	91.47	0	14.73	0.78	0
Declared	2	0.78	93.8	0	0	0
as	3	6.2	0	77.52	1.55	0
Class	4	1.55	0	7.75	97.67	0
	5	0	6.2	0	0	100

Scheme 6		Average	Classif.	Rate:	92%	
			True	Class	Label	
		1	2	3	4	5
	1	87.59	0	13.95	0.77	0
Declared	2	1.55	97.67	0	0	0
as	3	8.52	0	76.74	1.55	0
Class	4	2.32	0	9.3	97.67	0
	5	0	2.32	0	0	100

Scheme 7		Average	Classif.	Rate:	94%	
			True	Class	Label	
		1	2	3	4	5
Declared as Class	1	90.7	0	13.95	0	0
	2	0.78	98.45	1.55	0	0
	3	7.75	0.78	82.95	0.78	0
	4	0.78	0	1.55	99.22	0
	5	0	0.78	0	0	100

The best classification performance was obtained using all power features with BP NN (95%) and the next best performance was obtained using 44 power features selected by LDB based dimension reduction scheme with BP NN (94%). As a result, the LDB based dimension reduction scheme should be considered to reduce the number of features as it significantly reduces the BP NN computational load.

2. LDB Feature Extraction Scheme

We consider the following 6 classification methods based on the LDB feature extraction scheme:

1-512 LDB features followed by a CT (64%),

2-256 *most discriminating* LDB features followed by a BP neural network with configuration 256-50-5 (84%),

3-A combination of a *Mean Separator* neural network with 5 PEs followed by a BP neural network with configuration 5-5-5. Each PE contained in the *Mean Separator* NN was tuned to one signal class, following the Class-x/Class non-x scheme described earlier in Chapter V, Section 2a (82%),

4-A *mean separator* neural network with 5 PEs followed by the decision scheme to set class labels, as described in Chapter V, Section 2a and Figure 5.14 (81%),

5-A combination of a *Mean Separator* neural network with 10 PEs followed by a BP neural network with configuration 10-5-5. Each PE in the *Mean Separator* NN was trained to distinguish two signal classes pairwise, as described in Chapter V, Section 2b (86%),

6-A mean separator neural network with 10 PEs followed by the decision scheme to set class labels, as described earlier in Chapter V, Section 2b and Figure 5.14 (86%).

The confusion matrices are:

Scheme 1:		Average	Classif.	Rate:	64%	
			True	Class	Label	
		1	2	3	4	5
	1	86.82	3.87	58.91	17.05	1.55
Declared	2	0	65.89	6.97	4.65	10.1
as	3	3.1	0	0.77	1.55	0
Class	4	10.07	3.1	33.3	76.74	0
	5	0	27.13	0	0	88.4

Scheme 2: Average Classif. Rate: 84%						
		True Class Label				
		1	2	3	4	5
	1	86.82	0	15.5	9.3	0
Declared	2	0.78	99.22	0.78	0	0
as	3	6.2	0	55.04	8.53	0
Class	4	6.2	0	28.68	82.17	0
	5	0	0.78	0	0	100

Scheme 3: Average Classif. Rate: 82%						
		True Class Label				
		1	2	3	4	5
	1	78.29	0	14.73	0.78	0
Declared	2	0.78	83.72	1.55	0	0
as	3	19.38	0	48.84	1.55	0
Class	4	1.55	0	34.88	97.67	0
	5	0	16.28	0	0	100

Scheme 4:		Average	Classif.	Rate:	81%	
			True	Class	Label	
		1	2	3	4	5
	1	77.52	0	14.72	0.77	0
Declared	2	0.77	81.39	0	0	0
as	3	20.93	0	48.06	1.55	0
Class	4	0.77	0	37.2	97.67	0
	5	0	18.6	0	0	100

Scheme 5		Average	Classif.	Rate:	86%	
			True	Class	Label	
		1	2	3	4	5
	1	86.05	0	17.83	0.78	0
Declared	2	0.78	95.35	0.78	0	0
as	3	12.4	0	51.94	0.78	0
Class	4	0.78	0	29.46	98.45	0
	5	0	4.65	0	0	100

Scheme 6		Average	Classif.	Rate:	86%	
			True	Class	Label	
		1	2	3	4	5
	1	81.39	0	17.05	0	0
Declared	2	1.55	97.67	0	0	0
as	3	14.72	0	50.38	1.55	0
Class	4	2.32	0	32.55	98.45	0
	5	0	2.32	0	0	100

The best classification performance was obtained using a mean separator neural network with 10 PEs followed by the decision scheme (86%) or the BP NN (86%). The next best performance was obtained using 256 *most discriminating* LDB features followed by a BP neural network (84%).

A few comments are in order.

1- Classification rates show that the Power method performs well as a feature extraction method for underwater signals. In addition, results show that the overall classification rates obtained with the Power method is significantly higher than those obtained using LDB features. This is to be expected as we showed earlier that the Power method is more robust to time shifts than the LDB is, 2- The BP neural network gives better performance than the CT, as observed earlier with the synthetic data experiments,

3- The *mean separator* dimension reduction schemes perform quite well. Classification schemes combining the *mean separator* and the decision scheme gave same performance as those combining the *mean separator* and the BP neural network at a fraction of the computational cost. Thus, there is no need to use a BP neural network when the *mean separator* NN is selected for feature reduction step,

4- The LDB based dimension reduction scheme associated with the Power feature extraction method may also be considered as a good dimension reduction tool.

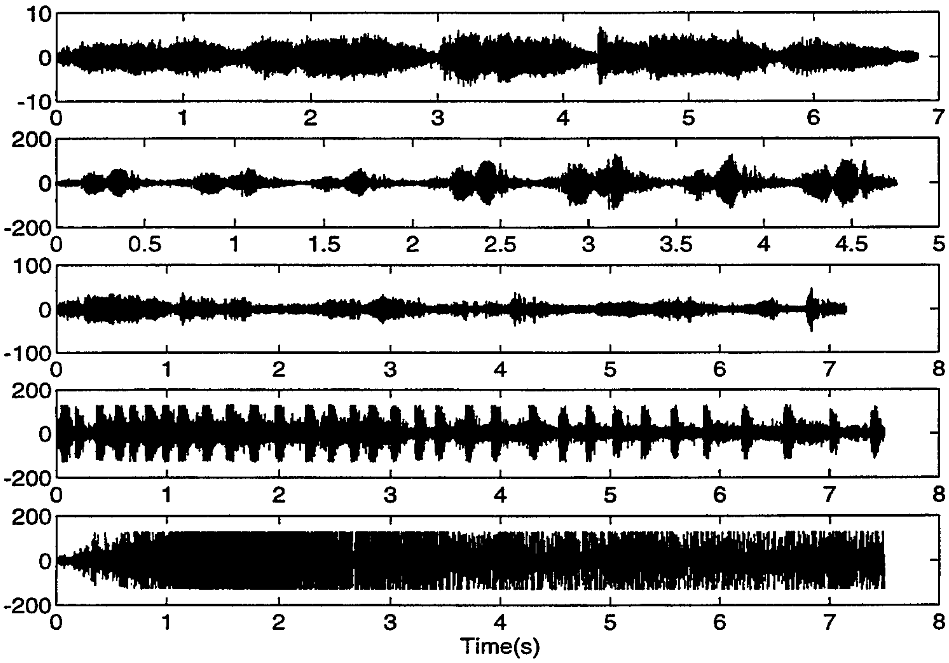

Figure 6.3: Time domain representations of sample recordings for (from the top) gray, humpback, killer, sperm whales and underwater earthquake.

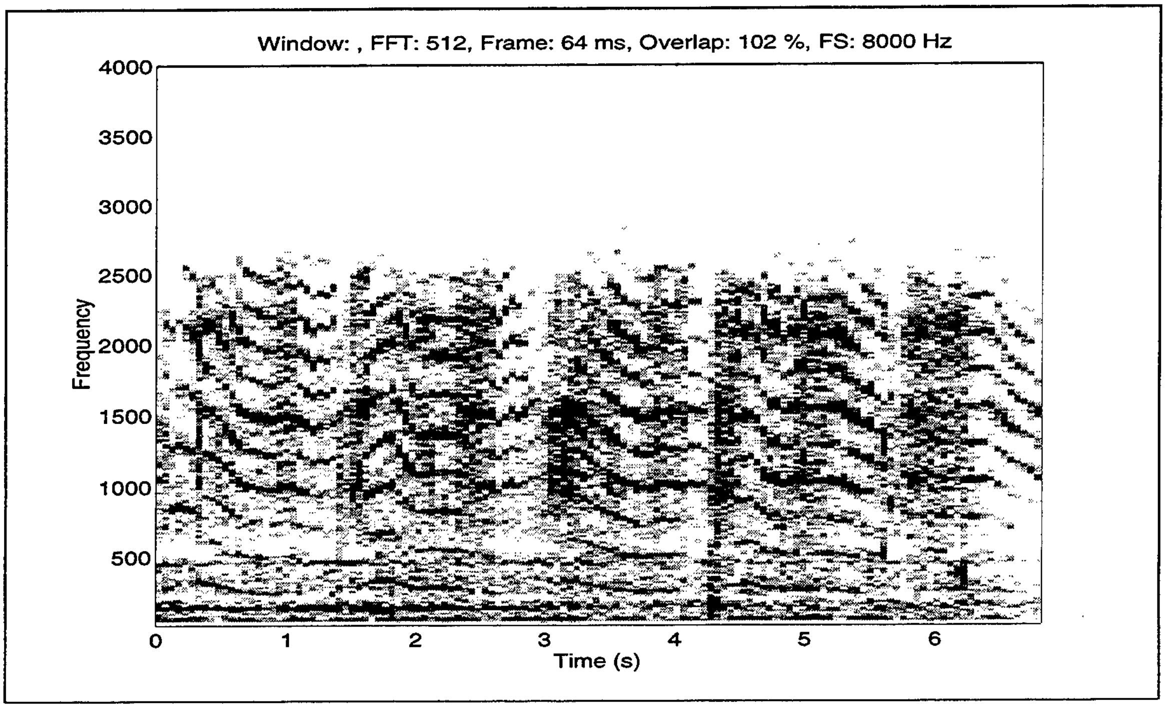

Figure 6.4: Spectrogram of gray whale recording.

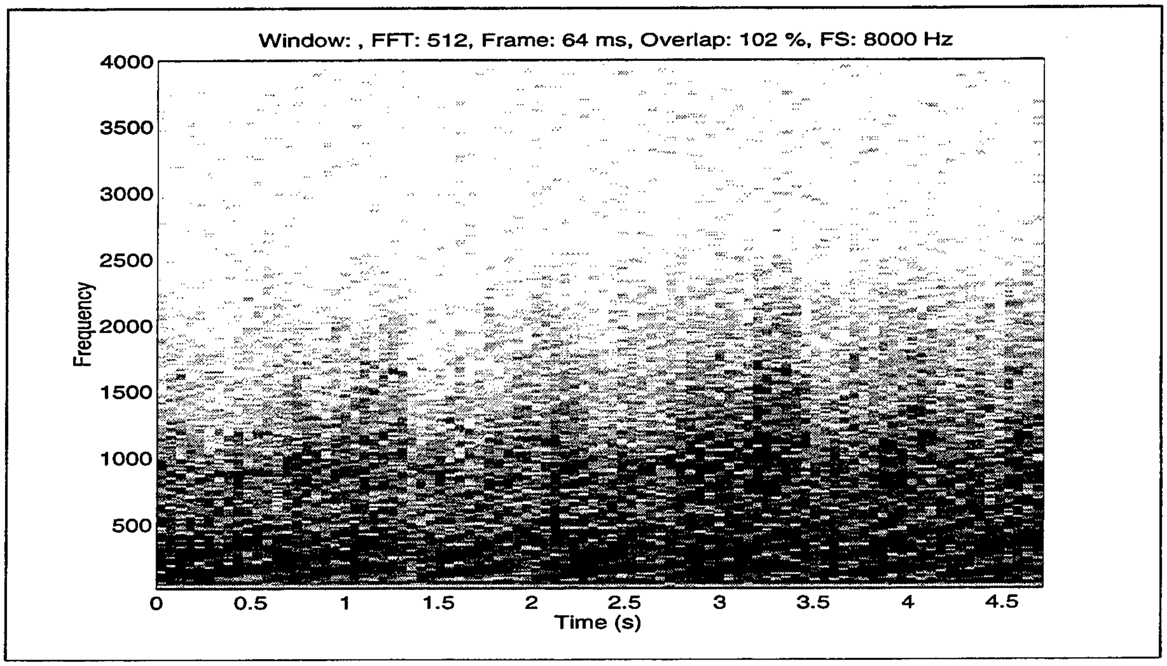

Figure 6.5: Spectrogram of humpback whale recording.

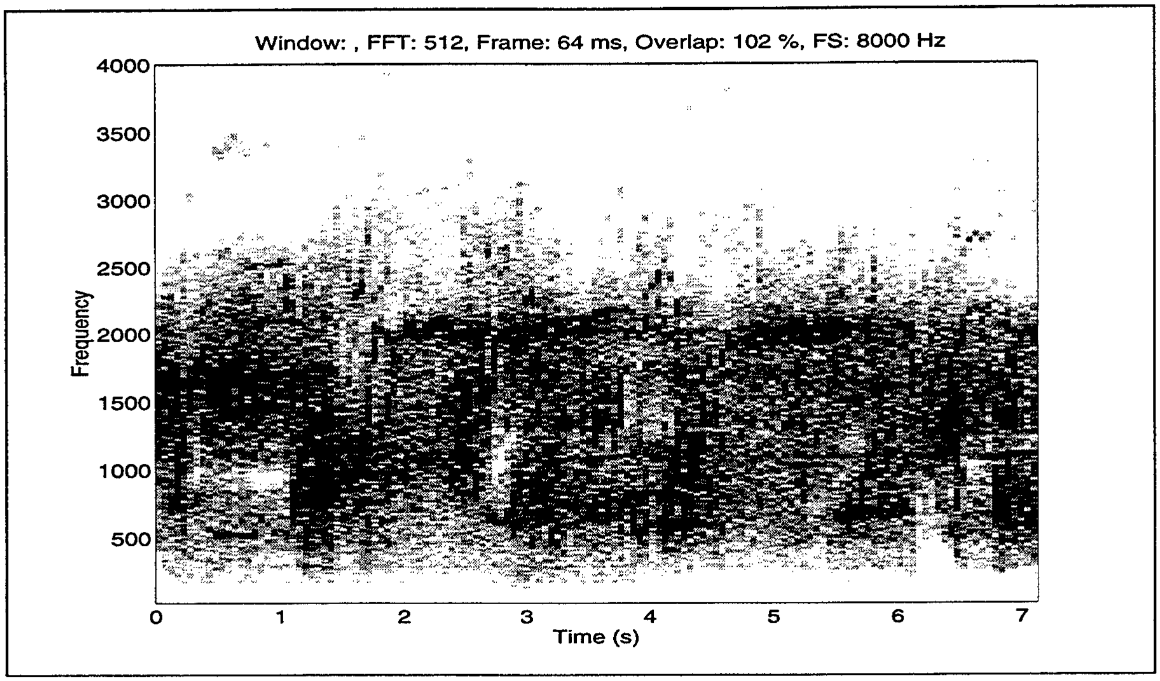

Figure 6.6: Spectrogram of killer whale recording.

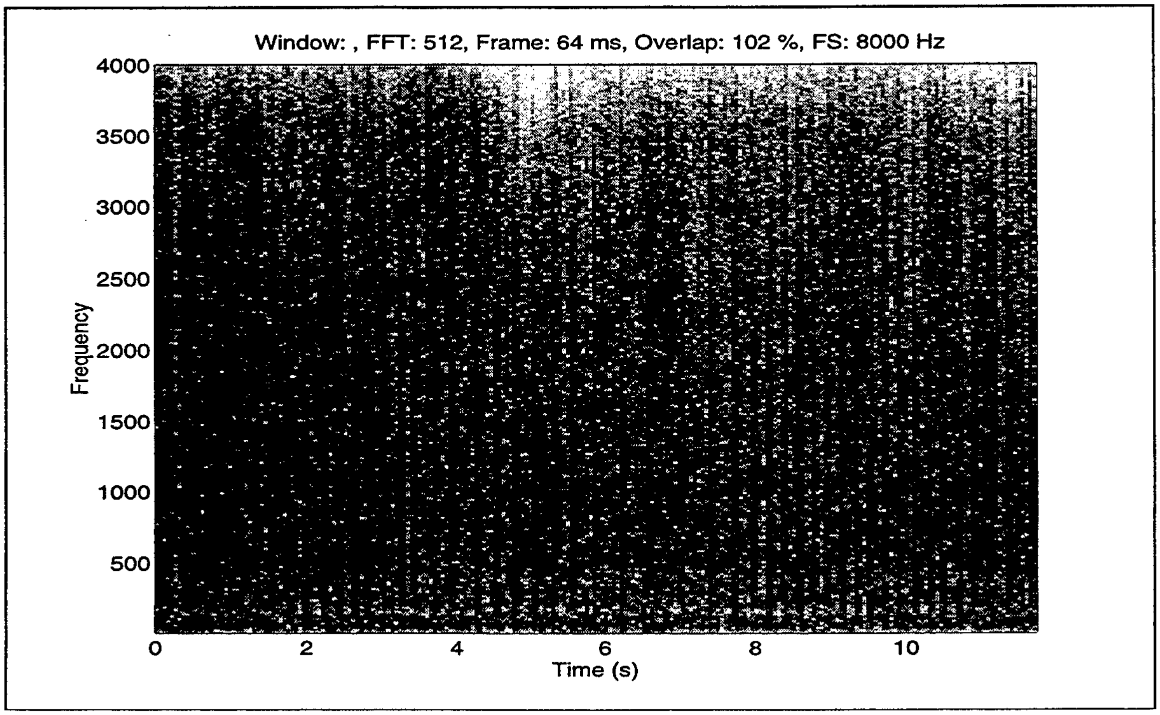

Figure 6.7: Spectrogram of sperm whale recording.

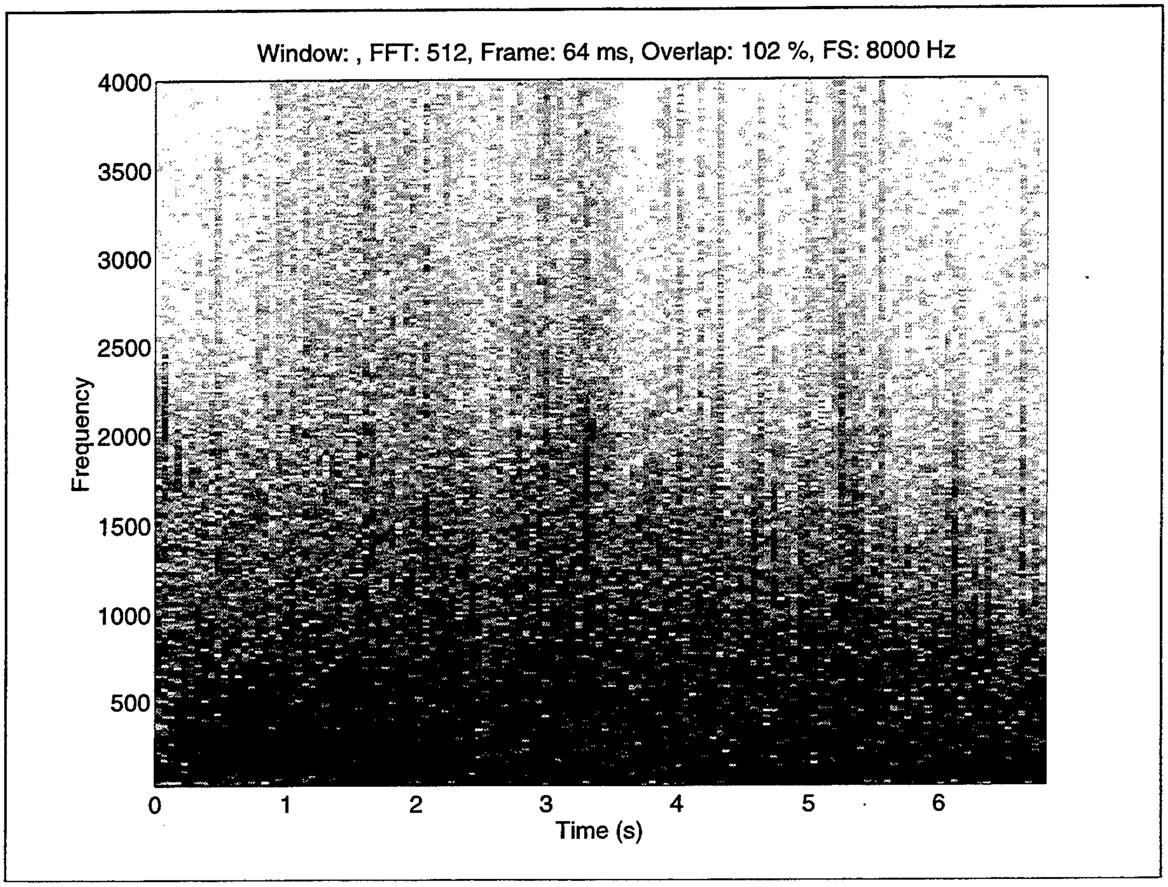

Figure 6.8: Spectrogram of underwater earthquake recording.

VII. CONCLUSIONS

In recent years, wavelet-based decompositions have been used in numerous areas such as engineering, finances, and medical applications. This popularity is due in part to their multi-resolution capabilities, which make them better matched to various signals of interest. In signal processing, wavelet decompositions have been applied to such areas as signal compression, noise removal and signal classification [12, 17, 23, 26].

This work considered the application of wavelet decompositions to classification applications. First, we investigated the application of the wavelet packet decomposition to the LDB scheme originally proposed by Saito, and showed that it is sensitive to time synchronization problems. Thus, we investigated an alternative, based on frequency band specific power quantities, which are more robust to time synchronization issues without worsening the classification performances.

Next, we presented and compared two main types of classifiers: back-propagation neural networks (BP NN) and classification trees (CT). Results showed that better performance was obtained with back-propagation neural networks. This is to be expected as BP NN have fewer constraints than CTs in partitioning the input spaces.

Next, we considered several feature extraction and dimension reduction methods. Such steps are key to obtaining good classification performance when the amount of data available to build the classification tools is limited, or when subject to computer capability constraints. We considered the BCM neural network implementation, which can be used as a feature reduction scheme, and showed that it is computationally slow. As a result, we proposed an alternative, called the mean separator neural network (MS NN), initially designed to distinguish between two classes, and extended it to the more-than two-classes case. We also showed that the MS NN can be followed by a decision step to create a stand alone classification scheme which has performances comparable to those obtained with more sophisticated classifiers, as a fraction of the computational cost.

We investigated the behavior of the various schemes considered both on synthetic and real-world underwater signals. Results also showed that the proposed MS NN is a successful dimension reduction scheme that may be used with both LDB and Power feature extraction methods.

For the underwater data considered, the following classification schemes can be ordered from best to worse in terms of overall classification performances:

1- Power method + MS NN + decision scheme,

2- Power method + MS NN + BP NN,

3- Power method + LDB based dimension reduction scheme + BP NN,

4- LDB + MS NN + decision scheme,

5- LDB + MS NN + BP NN.

REFERENCES

[1] A. V. Oppenheim, A. S. Willsky, and I. T. Young, *Signals and Systems*, Prentice-Hall, Inc., New Jersey, 1992

[2] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*, Prentice-Hall, Inc., New Jersey, 1992

[3] H. P. Hsu, *Applied Fourier Analysis*, Harcourt Brace College Outline Series, San Diego, 1991

[4] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley, 1991

[5] S. Haykin, *Communication Systems*, John Wiley &Sons, Inc., New York, 1994

[6] C. Sidney Burrus, R.A. Gopinath and H. Guo, *Introduction to Wavelets and Wavelet Transforms*, Prentice-Hall, Inc., Houston, Texas, 1996

[7] J. Buckheit, S. Chen, D. Donoho, and J. Scargle, "Wavelab.700", http://www.wavelab/playfair.stanford.edu, 1996

[8] J. Sadowsky, "The Continuous Wavelet Transform: A Tool for Signal Investigation and Understanding, " Johns Hopkins APL, Technical Digest, Vol. 15, No. 4, 1994

[9] G. Strang, T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, 1996

[10] O. Rioul and M. Vetterli, "Wavelets and Signal Processing," *IEEE Signal Processing Magazine*, October, 1991

[11] V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*, A. K. Peters, Ltd., Massachusetts, 1994

[12] R. Loe, K. Anderson, and K. Jung, "Comparative Analysis Results for Underwater Transient Classification," *SPIE*, Vol.2242, pp. 815-823, Wavelet Applications, 1994

[13] S. Del Marco, J. Weiss, and K. Jaggler, "Wavepacket-Based Transient Signal Detector Using a Translation Invariant Wavelet Transform," *SPIE*, Vol. 2242, pp. 792-802, Wavelet Applications, 1994

[14] R. Coifman and D. Donoho, "Translation-Invariant Denoising," *Internal Report*, Department of Statistics, Stanford University, 1995

[15] I. Cohen, S. Raz and D. Malah, "Orthonormal Shift-Invariant Wavelet Packet Decomposition and Representation," *Israel Institute of Technology Technical Report*, August, 1996

[16] M. Cody, "The Wavelet Packet Transform," *Dr. Dobbs Journal*, April 1994

[17] N. Saito, *Local Feature Extraction and Its Applications Using a Library of Bases*, Ph.D. Dissertation, Yale University, 1994

[18] R. J. Barsanti, Jr., *Denoising of Ocean Acoustic Signals Using Wavelet-Based Techniques*, MSEE Thesis, Naval Postgraduate School, Monterey, California, December, 1996

[19] C. W. Therrien, *Discrete Random Signals and Statistical Signal Processing*, Prentice- Hall, Inc., New Jersey, 1992

[20] R. E. Bellman, *Adaptive Control Process,* Princeton University Press, Princeton, New Jersey, 1961

[21] R. E. Learned and A. S. Willsky, " A Wavelet Packet Approach to Transient Signal Classification," Internal Report, Department of Electrical and Computer Science, Massachusetts Institute of Technology, 1996

[22] G. Strang, *Linear Algebra and Its Applications,* Harcourt Brace Jovanovich, Inc., Florida, 1988

[23] M. P. Fargues and R. C. Bennett, "Comparing Wavelet Transform and AR Modeling as Feature Extraction Tools for Underwater Signal Classification," *Proceedings of the 29th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, California, October 1995

[24] R.C. Bennett, *Classification of Underwater Signals Using a Backpropagation Neural Network*, MSEE Thesis, Naval Postgraduate School, June 1997

[25] N. Saito and R. R. Coifman, "Improved Local Discriminant Bases Using Empirical Probability Density Estimation," *Amer. Statist. Assoc. Proc. Statistical Computing*, 1996

[26] R. Coifman and M. Wickerhauser, "Entropy Based Algorithms for Best Basis Selection," *IEEE Trans. On Information Theory,* Vol. 3b, No. 2, March 1992

[27] J. Jang, C. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, 1997

[28] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Chapman & Hall, Inc., New York, 1984

[29] M. T. Hagan, H. B. Demuth and M. Beale, *Neural Network Design,* PWS Publishing Company, Boston, Massachusetts, 1996

[30] W. N. Venables and B. D. Springer, *Modern Applied Statistics with S-Plus,* Prentice-Hall, Inc., March, 1997

[31] NeuralWorks Professional II/PLUS, Version 5.23, NeuralWorks, Inc., 1996

[32] N. Intrator and L. N. Cooper, "Objective Function Formulation of the BCM Theory of Visual Cortical Plasticity: Statistical Connections, Stability Conditions," *Neural Networks,* Vol. 5, pp., 3-17, 1992

[33] J. H. Friedman, "Exploratory Projection Pursuit," *Journal of the American Statistical Association,* Vol. 82, No. 397, Theory and Methods, March 1987

[34] N. Intrator, "Feature Extraction Using an Unsupervised Neural Network," *Neural Computation,* Vol. 4, pp. 98-107, 1992

[35] Neural Network Toolbox, Version 2.0b, The MathWorks, Inc., 1994

[36] T. W. Anderson, *Introduction to Multivariate Analysis,* John Wiley, Inc., New York, 1958

[37] R. A. Becker and J. M. Chambers, *S: An Interactive Environment for Data Analysis and Graphics,* Wadsworth, Inc., Belmont, 1984

[38] S-Plus For Windows, Version 3.3, MathSoft, Inc., 1995

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Chairman, Code EC 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

4. Prof. Monique P. Fargues, Code EC/Fa 2
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

5. Prof. Ralph Hippenstiel, Code EC/Hi 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

6. Mr. Steve Greneider, Code 2121 1
   Naval Undersea Warfare Center-Newport Division
   Building 1320, Rm 381
   Newport, RI 02841

7. Dr. Paul M. Baggenstoss, Code 2121 1
   Naval Undersea Warfare Center-Newport Division
   Building 1320, Rm 381
   Newport, RI 02841

8. Deniz Kuvvetleri Komutanligi..2
Personel Daire Bsk.ligi
Bakanliklar, Ankara, Turkiye

9. Deniz Harp Okulu Komutanligi 1
Kutuphane
81704 Tuzla, Istanbul, Turkiye

10.Asim Duzenli, Yeni Emniyet 4 2
sitesi 231. Sokak kent-kop
mahallesi 2. blok no. 14
Batikent, Ankara, Turkiye