

56649

ELECTROMAGNETIC TARGET CLASSIFICATION BY USING
TIME-FREQUENCY ANALYSIS AND NEURAL NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

TÜRKER İNCE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS

ENGINEERING

AUGUST 1996

Approval of the Graduate School of Natural and Applied Sciences.



Prof. Dr. Tayfur Öztürk
for Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of MASTER OF SCIENCE.



Prof. Dr. Fatih Canatan
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of MASTER OF SCIENCE.



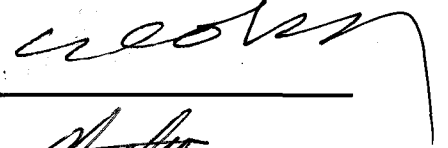
Assoc. Prof. Dr. Kemal
Lelebicioğlu
Co-Supervisor



Assoc. Prof. Dr. Gönül
Turhan Sayan
Supervisor

Examining Committee Members

Prof. Dr. Canan Toker (Chairman)



Assoc. Prof. Dr. Gönül Turhan Sayan



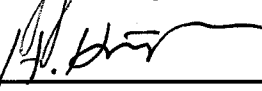
Assoc. Prof. Dr. Kemal Lelebicioğlu



Assist. Prof. Dr. Nevzat Gencer



Ms. E. E. Eng. Hakan Köroğlu



ABSTRACT

ELECTROMAGNETIC TARGET CLASSIFICATION BY USING TIME-FREQUENCY ANALYSIS AND NEURAL NETWORKS

İnce, Türker

M.S., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Gönül Turhan Sayan

Co-Supervisor: Assoc. Prof. Dr. Kemal Leblebicioğlu

August 1996, 115 pages

In this thesis, the design of new electromagnetic target classification systems by using time-frequency analysis and neural networks is presented. For extracting the feature vectors from the time-domain input data, the Wigner distribution is utilized in all proposed classification schemes mainly because of its distributed energy interpretation. The multilayer perceptron, the self-organizing feature map, and the principal component analysis are employed alternatively for classification purpose. The performance of each classification scheme is examined with the available time-domain database of perfectly conducting spheres of different

diameters and perfectly conducting small scale airplanes. Also, by modifying one of the proposed classification schemes, the feasibility of estimating the aspect angle of a classified input data is demonstrated with simulations. The noise performance of the most promising classification scheme is analyzed by using input data with additive Gauss noise in testing phase only.

Keywords: Electromagnetic target classification system, time-frequency analysis, the Wigner distribution, multilayer perceptron, the self-organizing feature map, principal component analysis



ÖZ

ZAMAN-FREKANS ANALİZİ VE SINIR AĞLARI KULLANILARAK ELEKTROMANYETİK HEDEF SINIFLANDIRILMASI

İnce, Türker

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü Bölümü

Tez Yöneticisi: Assoc. Prof. Dr. Gönül Turhan Sayan

Ortak Tez Yöneticisi: Assoc. Prof. Dr. Kemal Leblebicioğlu

Ağustos 1996, 115 sayfa

Bu çalışmada, zaman-frekans analizi ve sinir ağları kullanılarak yeni elektromanyetik hedef sınıflandırma sistemleri tasarımı sunulmuştur. Bütün önerilen sistemlerde, zamandaki giriş bilgisinden özelliği olan vektörlerin elde edilmesi için, doğal enerji yorumuna sahip olması nedeniyle Wigner dağılımı kullanılmıştır. Sınıflandırma amacıyla Çok Katlı Algılayıcı, Kendiliğinden Organize Olan Özellik Dönüştürücü ve Asıl Eleman Analizi metodlarından birisi kullanılmıştır. Önerilen sınıflandırma sistemlerinin performans analizleri değişik çaplara sahip tam iletken küreler ve tam iletken küçük boyutlu uçaklara ait mevcut veritabanı kullanılarak

gerçekleştirilmiştir. Ayrıca, önerilen sistemlerden birisinin değişik versiyonu kullanılarak yapılan simülasyonlarla sınıflandırılmış hedef bilgisine ait görünüş açısının tahmin edilebilirliği gösterilmiştir. Önerilen sistemlerden en iyi olanına ait gürültü performansı, giriş bilgilerine sadece test aşamasında Gauss gürültüsü katılarak, analiz edilmiştir.

Anahtar Kelimeler: Elektromanyetik hedef sınıflandırma sistemi, zaman-frekans analizi, Wigner dağılımı, Çok Katlı Algılayıcı, Kendiliğinden Organize Olan Özellik Dönüştürücü, Asıl Eleman Analizi



ACKNOWLEDGMENTS

I am very grateful to my supervisors, Associate Professor Gönül T. Sayan and Associate Professor Kemal Leblebiciođlu for their invaluable guidance and motivating support during this study.

Also I would like to thank to Prof. Dr. Canan Toker, Assoc. Prof. Dr. Gülbin Dural, Assist. Prof. Dr. Nevzat Gencer and Hakan Körođlu for reading and commenting about the thesis. And special thanks to all my friends for their valuable discussions and helps.

Finally, I want to express my deepest gratitude to my family, for everything they did to bring me to this position. I dedicate this thesis to them.

TABLE OF CONTENTS

ABSTRACT	iii
ÖZ	v
ACKNOWLEDGMENTS	vii
TABLE OF CONTENTS	viii
LIST OF TABLES	x
LIST OF FIGURES	xi
CHAPTER	
1 INTRODUCTION	1
2 TIME-FREQUENCY SIGNAL ANALYSIS	6
2.1 Introduction	6
2.2 Some Important Time-Frequency Analysis Methods	9
2.2.1 Short Time Fourier Spectrum and Spectrogram	9
2.2.2 The Wavelet Transform	11
2.2.3 The Wigner Distribution	15
2.2.4 Generalized Class of TFRs	25
2.3 Application of the WD for Feature Extraction	27
3 NEURAL NETWORKS AND CLASSIFICATION	34
3.1 Introduction	34
3.2 Multilayer Perceptron	35
3.3 Self-Organizing Map	40
3.4 Applications of MLP and SOM for Classification	43
4 PRINCIPAL COMPONENT ANALYSIS AND CLASSIFICATION	50
4.1 Introduction	50
4.2 Principal Component Analysis	51

4.3	Some Useful Properties of Principal Components	54
4.4	Applications of PCA for Classification	56
5	APPLICATIONS OF TARGET CLASSIFICATION	59
5.1	Introduction	59
5.2	Preliminary Information on the Classes and the Database	60
5.3	Simulation Results for the Sphere Class	61
5.3.1	Simulation Problem 1	62
5.3.2	Simulation Problem 2	66
5.3.3	Simulation Problem 3	74
5.3.4	Simulation Problem 4	77
5.3.5	Simulation Problem 5	81
5.4	Simulation Results for the Aircraft Class	83
5.4.1	Simulation Problem 6	83
5.4.2	Simulation Problem 7	90
5.4.3	Simulation Problem 8	92
5.4.4	Simulation Problem 9	96
5.5	Noise Performance of the Proposed Classification System .	97
6	CONCLUSIONS	101
	REFERENCES	103
	APPENDICES	108
A	ALGORITHMS OF THE SIMULATION PROGRAMS	108
B	GENERATION OF NOISY DATA	115

LIST OF TABLES

5.1	Classification results of the sphere class by using only MLP classifier	65
5.2	Classification results of the sphere class by using the WD+MLP classifier	73
5.3	Classification results of the aircraft class by using the WD+MLP classifier	90



LIST OF FIGURES

1.1	Monostatic and bistatic aspect angle definitions (top view)	2
2.1	Backscattered impulse response of a thin wire at 30° aspect angle	30
2.2	The contour plot of WD of the impulse response shown in Figure 1.1	31
2.3	The contour plot of time-averaged WD of the impulse response shown in Figure 1.1	31
2.4	The selected feature vector for 12m wire at 30° aspect angle . . .	33
2.5	The selected feature vector for 12m wire at 60° aspect angle . . .	33
2.6	The selected feature vector for 12m wire at 90° aspect angle . . .	33
3.1	MLP with two hidden layers	36
3.2	Single neuron model	37
3.3	SOM Network Model	41
3.4	SOM+MLP classification system model	48
4.1	PCA output for 7 inch dia. sphere at 120° aspect angle	58
4.2	PCA output for 10 inch dia. sphere at 120° aspect angle	58
4.3	PCA output for 20 inch dia. sphere at 120° aspect angle	58
5.1	Block diagram of the classification scheme discussed in Simulation problem 1	62
5.2	Normalized impulse response data for 7 and 10 inch dia. spheres .	63
5.3	Normalized impulse response data for 15 and 20 inch dia. spheres	63
5.4	Block diagram of the classification scheme discussed in Simulation problem 2	66
5.5	The energy feature matrices for 7 and 10 inch dia. spheres	68
5.6	The energy feature matrices for 15 and 20 inch dia. spheres	68
5.7	The selected feature vector for 10 inch dia. sphere at 30° aspect angle	69
5.8	The selected feature vector for 10 inch dia. sphere at 60° aspect angle	69
5.9	The selected feature vector for 10 inch dia. sphere at 90° aspect angle	70
5.10	The selected feature vector for 10 inch dia. sphere at 120° aspect angle	70
5.11	The instantaneous squared error per output of the MLP classifier	73
5.12	Block diagram of the classification scheme discussed in Simulation problem 3	74

5.13	PCA outputs for 7 inch dia. sphere at 0° and 45° aspect angles . .	75
5.14	PCA outputs for 7 inch dia. sphere at 120° and 150° aspect angles	75
5.15	Block diagram of the classification scheme discussed in Simulation problem 4	77
5.16	The final SOM output (10x10) of sphere class	80
5.17	Block diagram of the classification scheme discussed in Simulation problem 5	81
5.18	The final SOM output (4x5) for aspect estimation of 7 inch dia. sphere data	82
5.19	The final SOM output (4x5) for aspect estimation of 10 inch dia. sphere data	83
5.20	Normalized impulse response data for plane B-707	84
5.21	Normalized impulse response data for plane B-727	84
5.22	Normalized impulse response data for plane B-747	85
5.23	Normalized impulse response data for plane Concorde	85
5.24	Normalized impulse response data for plane DC-10	86
5.25	The selected feature vector for plane B-707 at 45° aspect angle . .	88
5.26	The selected feature data for plane B-727 at 45° aspect angle . . .	88
5.27	The selected feature data for plane B-747 at 45° aspect angle . . .	88
5.28	The selected feature data for plane Concorde at 45° aspect angle .	89
5.29	The selected feature data for plane DC-10 at 45° aspect angle . .	89
5.30	PCA output for plane B-707 at 45° aspect angle	90
5.31	PCA output for plane B-727 at 45° aspect angle	91
5.32	PCA output for plane B-747 at 45° aspect angle	91
5.33	PCA output for plane Concorde at 45° aspect angle	91
5.34	PCA output for plane DC-10 at 45° aspect angle	92
5.35	The final SOM output (15x15) of the aircraft class	95
5.36	The final SOM output (4x5) for aspect estimation of B-707 plane data	96
5.37	The final SOM output (4x5) for aspect estimation of B-727 plane data	96
5.38	The final SOM output (4x5) for aspect estimation of DC-10 plane data	97
5.39	Noise analysis for 10 inch sphere at 0° aspect angle with 15dB, 10dB and 5dB SNR levels	99
5.40	Noise analysis for B-707 aircraft at 60° aspect angle with 15dB and 11dB SNR levels	99

CHAPTER 1

INTRODUCTION

Target classification may simply be defined as the process of deciding which one of the prespecified object classes is the most consistent with the unknown input feature data, or target signature. The problem of identifying or classifying objects using back-scattered signals has been of interest for a long time. Many different target discrimination schemes have been proposed in the past decades [1]-[7]. Most of these schemes use the measured back-scattered response of a target at a set of frequencies. The time-domain target response obtained by the inverse fast Fourier transform (IFFT) of the measured data can also be used as the raw input.

All target classification algorithms primarily extract various features of the unknown object, which uniquely characterize it depending on its physical properties (size, shape and composition), in the classification process. The classification performance is very much affected by the discriminating power of the extracted target features. Therefore, the selection of proper feature vectors, either in time-

domain or in frequency-domain, directly affects the success of target classification system.

An important problem associated with using the back-scattered response of a target is that it highly depends on the aspect angle, frequency and polarization. One well-known measure that represents the target scattering characteristics is the radar cross section (RCS) [8]. However, since the RCS of a target is a function of frequency, aspect and radar polarization, it is not practicle to use it directly in target discrimination schemes. For the back-scattered response of a target, the definitions of monostatic and bistatic aspect angles are shown in Figure 1.1.

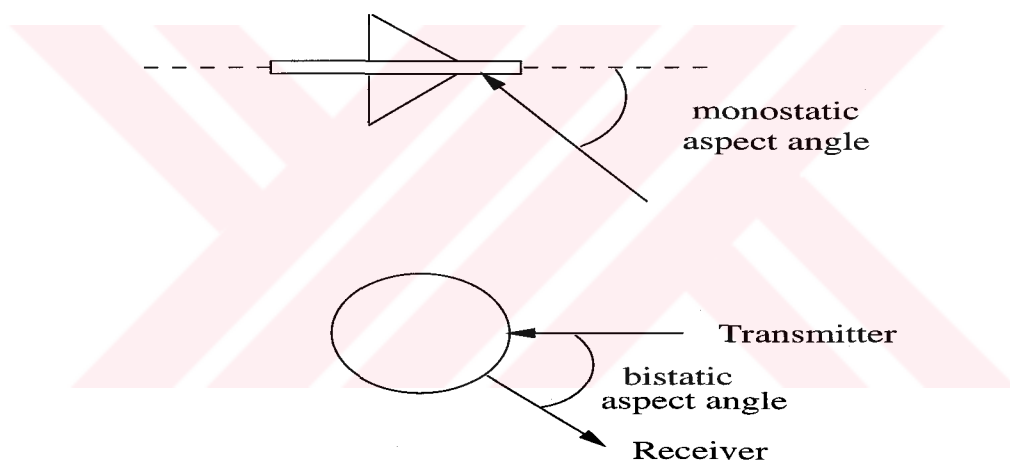


Figure 1.1: Monostatic and bistatic aspect angle definitions (top view)

In 1965, Kennaugh and Moffatt proposed the concept of natural resonance frequencies and showed that radar returns in the resonance region provide the essential information about the overall size, shape, and structure of the object [9]. This paper implies that targets in the resonance range show significant damped oscillations in the RCS due to the zeros of the denominator of the transfer function of the object. These zeros are called the *complex natural resonant* (CNR)

frequencies of the target. This idea then resulted some target discrimination schemes which are based on the late-time natural oscillation behavior of conducting targets. In the early 1970s, Baum established the formulation of the complex natural resonance idea as the singularity expansion method (SEM) [10]. According to SEM, the late-time scattering response from a conducting scatterer can be represented as a series expansion of simple poles in frequency domain. The CNR frequencies (or target poles) are useful characteristics of targets, since they are uniquely determined by physical properties of an object and they are independent of aspect angle and polarization. However, extraction of CNR frequencies from available measurement data is extremely noise sensitive. So instead of using the direct target pole comparison with a database of known targets in a classification scheme, it is possible to develop different classification techniques where the discriminating power of target poles can be utilized indirectly.

The main purpose of this thesis is to design a new electromagnetic target classification system which can be operated at a reasonably high correct decision rate even for a modest size of database. Besides, a short response time in the decision phase, robustness to noise and easy adaptation to practical applications are the other important requirements expected to be met by this classification system. In this thesis, we will propose several target classifier prototypes, which utilize various combinations of neural networks (multilayer perceptrons and self-organizing maps), the Wigner time-frequency distribution and the statistical principal component analysis method. The common property of these proposed classifiers is that they all utilize most of the useful information in the time-frequency content of the signal in feature extraction part by use of the Wigner time-frequency

distribution.

It is well known that the available time-domain or frequency-domain data for an electromagnetic scatterer are aspect and polarization dependent. Using such data directly, without any signal processing, in a target classification study requires a very large database which takes a very long processing time. Instead, the available data must be pre-processed for feature extraction or feature enhancement for faster and more accurate classification. Our consideration of using the Wigner distribution in a feature extraction part of target classification scheme is mainly motivated by its distributed energy interpretation. The time-frequency localized energy spectrum matrix, which is obtained from the impulse response of a target, forms the feature vector of a target.

The neural network approach to target classification has an important role in different methods mainly due to its powerful learning algorithms [11]. Multilayer perceptrons (MLPs) are the most widely used neural networks especially for classification applications, whereas the natural clustering ability of self-organizing feature maps (SOFMs) is still being studied by researchers and some successful experiment results have been reported in the last years [12, 13]. In our classification system we will use SOFM network, which has originally an unsupervised learning algorithm, in a supervised manner by applying the forced winner training algorithm. On the other hand, principal component analysis (PCA) is a well-known classical statistical method which was proved to be as a powerful feature extraction and selection technique in many applications [14].

The organization of the thesis is as follows: Chapter 2 reviews time-frequency signal analysis, some important time-frequency representations that have been

developed and their properties, and discusses the application of the WD for feature extraction. Chapter 3 presents the MLP and SOM neural networks and their applications to target classification. Chapter 4 outlines the alternative statistical method PCA and its application to target classification. In Chapter 5, the simulation examples with the available sphere class and aircraft class measurement data by using the proposed classification schemes and discussion and comparison of the results are presented. Also at the end of this chapter, some results of noise analysis are given. Chapter 6 includes the concluding remarks and a discussion of possible future works. In Appendix A, the algorithms of the simulation program codes, written in C, are given. The details of generation of noisy data for noise performance analysis are given in Appendix B.

CHAPTER 2

TIME-FREQUENCY SIGNAL ANALYSIS

2.1 Introduction

Most of real-life processes have a nonstationary nature resulting in time-varying signal characteristics. Joint time-frequency analysis is a new signal processing technique in which signals are analyzed in both time domain and frequency domain simultaneously. Time-frequency representations (TFRs), such as the spectrogram, the short-time Fourier transform, the Wavelet transform, and the Wigner distribution, analyzing the energy content of signals simultaneously in time and frequency, have drawn great attention for the analysis and processing of signals in a wide variety of applications related to electromagnetics, signal processing, communication, biomedical and geophysical systems.

In most applications, the classical techniques based on stationary spectral analysis are inadequate in analysing the time-varying nature of nonstationary and transient signals. For many years, Fourier transform has a wide range of

applications for signal processing but mainly for stationary signals. By means of standard Fourier analysis, a signal can be decomposed into individual frequency components with relative intensities. However, especially in case of nonstationary signals whose spectral content is changing in time (time-varying spectrum) it is not possible to determine when those individual frequencies existed from the Fourier analysis output. The well-known Fourier transformation of a continuous-time signal $f(t)$ is defined as,

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \quad (2.1)$$

where the angular frequency $\omega = 2\pi f$ and the associated inversion from frequency domain to time domain is obtained as,

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{j\omega t} d\omega \quad (2.2)$$

For discrete-time signals, Discrete Time Fourier Transformation (DTFT) is defined by the pair of equations given below :

$$X(\omega) = \sum_n x(n)e^{-j\omega n} \quad (2.3)$$

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega)e^{j\omega n} d\omega \quad (2.4)$$

While each signal has a unique Fourier spectrum, a time-frequency analysis of a signal can be done by using many different TFRs. As opposed to Fourier analysis, a TFR maps a one-dimensional signal of time to a two-dimensional

function of time-frequency plane to examine the variation of the frequency content of the signal with respect to time.

Most time-frequency representations, such as the spectrogram, Wigner distribution and pseudo Wigner distributions which belong to Cohen's class [15], are covariant to time and frequency shifts of the signal. Other time-scale distributions, among which the most popular one is the wavelet transform that belongs to affine class [15], are covariant to time shifts and scale changes in the signal. For a large variety of applications where time shifts, frequency shifts and scale changes are the fundamental transformations, the joint time-frequency and time-scale distributions are invaluable tools for data analysis and signal processing purposes.

Main idea behind a joint function of time and frequency is to describe the energy density of a nonstationary signal simultaneously in time and frequency. By the use of joint time-frequency distributions it is possible to determine the amount of energy in a specific range of time and frequency, the distribution of frequency at a specific time, the mean frequency and its local spread by the global and local moments of the distribution. In addition, one can synthesize signals having desirable time-frequency characteristics and also apply certain operations and transformations in time-frequency domain.

From a joint distribution $P(t,w)$, the instantaneous power $|s(t)|^2$ at a specific time and the energy density spectrum $|S(w)|^2$ at a specific frequency can be obtained by the following integral expressions which are called as the marginals,

$$\int P(t, w)dw = |s(t)|^2 \quad (2.5)$$

$$\int P(t, w)dt = |S(w)|^2 \quad (2.6)$$

Provided that the marginals are satisfied, the total energy E of the signal can be found by summing up the distribution at all times and frequencies as,

$$E = \int \int P(t, w)dw dt \quad (2.7)$$

In this chapter, we will first review the linear TFRs such as the short-time Fourier transform and the wavelet transform. Secondly, the quadratic (bilinear) TFRs such as the Wigner distribution and the other distributions in the generalized class of TFRs will be summarized. Finally, in this chapter, applications of the Wigner distribution for feature extraction will be discussed.

2.2 Some Important Time-Frequency Analysis Methods

2.2.1 Short Time Fourier Spectrum and Spectrogram

A time-varying signal can be analysed by use of a sliding time window as in the most widely used standard methods Short Time Fourier Transform (STFT) [16], or spectrogram. STFT concept in Electrical Engineering was first introduced in 1946 by Gabor to analyze the time variation of the frequency contents of signals. STFT is simply the result of multiplying the time series by a short time window and then taking DFT of it, to obtain a local spectrum of the signal around the analysis time.

Mathematically,

$$Y(t, \omega) = \int_{-\infty}^{\infty} y(s) \gamma^*(s - t) e^{-j\omega s} ds \quad (2.8)$$

From equation 2.8, we can interpret the STFT as a linear filtering operation where for a fixed ω_0 , $\gamma^*(-t)e^{j\omega_0 t}$ is the impulse response of the filter.

Possibly with a different window function $\alpha(u)$, the inverse STFT is given as,

$$y(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} Y(u, \omega) \alpha(t - u) e^{j\omega t} du d\omega \quad (2.9)$$

The action of this window is to provide localization in time so that the resulting spectrum is the related local spectrum. This window is then translated along the time axis to produce local spectra for all time which is referred to as a spectrogram [17]. In this case, the assumption of stationarity over the window width may cause some error depending upon the degree of actual nonstationarity of the signal. This error is a result of the inherent tradeoff between time and frequency resolution associated with the famous uncertainty principle which says that the product of the time duration δt and the frequency duration (bandwidth) $\delta \omega$ of a window is bounded below by $1/2$ [18].

The window duration and the bandwidth determine the time and frequency resolution respectively in the short time spectra. This means that if the window duration is too small, then the frequency resolution will be poor and if the window duration is too long, the time resolution will be poor. For a non-stationary signal, since spectral content is changing in time, poor time resolution can also mean poor frequency resolution. So the length of the assumed short-time stationarity

determines the obtainable frequency resolution. This windowing scheme can be described as constant width windowing. The time resolution is the same for all spectral components. The amplitude for any given spectral component of the DFT gives the average over the windowed time series. The choice of window size is completely arbitrary. With the STFT, because there may be different number of periods of each spectral component in the window, the response depends on the frequency. Therefore it is sometimes necessary to perform two STFTs with different window sizes to achieve adequate time and frequency resolution.

For implementation purposes, the form of STFT for discrete-time signals with use of FFT is defined as follows,

$$Y_{m,n} = \sum_{k=-\infty}^{\infty} y_k \gamma_{k-nN}^* e^{-j2\pi k \frac{m}{M}} \quad (2.10)$$

where $0 \leq m \leq M - 1$, $-\infty < n < \infty$.

The energy density spectrum or spectrogram, which is the energy density at a specific time and frequency is defined as,

$$P_s(t, w) = |S_t(w)|^2 \quad (2.11)$$

Actually, the spectrogram is nothing but a time-domain filtering followed by a spectral detection at a given time.

2.2.2 The Wavelet Transform

Wavelet analysis was first introduced by J. Morlet [19] for the seismic data analysis and then I. Daubechies [20] was the first to use wavelets in signal analysis

applications. There is a similarity between the FT and the wavelet transform in the following sense: The FT is the ideal tool for stationary signals whose statistical properties are invariant over time. Actually, the FT is an infinite linear combination of dilated cosine and sine waves. Similarly the non-stationary signals can be represented by linear combinations of special wavelets which allow us to extract the simple constituents of a complicated signal. The basic idea here is to create a set of basis functions and transforms which will give a description of a signal localized in both time and frequency (or scale).

Wavelet transformation is a powerful tool for the analysis and synthesis of signals. Localization of signal characteristics in spatial (or time) and frequency domains can be accomplished very efficiently with wavelets. This allows us to simultaneously determine sharp transitions in the spectrum of the signal and in the position (or time) of their occurrence.

Similar to the STFT, the Wavelet Transform (WT) can be interpreted as a linear filtering operation [21]. We have seen that STFT does not have an optimal localization in both time and frequency domains simultaneously. However the wavelet transform has optimal localization in both domains which is achieved by varying the length of the analysis window. Use of short basis functions at high frequencies and long ones at low frequencies make the isolation of signal discontinuities in both domains possible.

The continuous wavelet transform (CWT) is the decomposition of a function over a set of wavelets which are all the translations and scalings of a prototype function often called a mother wavelet.

The mother wavelet $h_{a,b}(t) = \frac{1}{\sqrt{a}}h(\frac{t-b}{a})$, where a and b are parameters that scale and dilate the mother function $h(t)$ to generate wavelets, is required to satisfy the admissibility condition which is given as,

$$\int_{-\infty}^{\infty} \frac{|H(w)|^2}{|w|} dw < \infty \quad (2.12)$$

where $H(w)$ is the Fourier transform of $h_{a,b}(t)$.

Then the CWT is defined in terms of inner products of the signal with the wavelets as follows,

$$F_w(t, s) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} f(\tau) h^*\left(\frac{\tau - t}{s}\right) d\tau \quad (2.13)$$

For recovery of the original signal from its continuous wavelet coefficients, the inverse CWT is defined as,

$$x(t) = c \int_0^{\infty} \int_{-\infty}^{\infty} F_w(\tau, s) \frac{1}{\sqrt{s}} h\left(\frac{\tau - t}{s}\right) \frac{d\tau ds}{s^2} \quad (2.14)$$

provided that $\int h(\tau) d\tau = 0$, and c is a constant.

As can be seen from its definitions, while the WT preserves time shifts and time scalings, it doesn't preserve frequency shifts. In fact, The WT is very much related to the multiresolution analysis of Mallat [22]. The basic structure of a WT is composed of recursive filtering and decimation, both of which are relatively easy to implement. Multiresolution signal analysis and the pyramidal algorithm are used to obtain wavelet coefficients used in wavelet analysis and synthesis [22].

According to multiresolution analysis [22], it is possible to represent any function $f(t)$ in $L^2(R)$ for a given scale by dilating and translating a single function

$\psi(t)$ as,

$$v^j(t) = \sum_n a_n^j \psi_{2^j}(t - 2^{-j}n) \quad \text{where } \psi_{2^j}(t) = 2^{j/2} \psi(2^j t) \quad (2.15)$$

Here a_n^j are expansion coefficients at successive scales and $v^j(t)$ is an approximation of $f(t)$ in vector space (V_{2^j}) within a scale j , and as $j \rightarrow \infty$, $v^j(t)$ approaches to $f(t)$. At the finest scale J , the signal is represented by 2^J coefficients as follows,

$$v^J(t) = \sum_{n=1}^{N_J} a_n^J \psi_{2^J,n}(t) = \sum_n a_n^J 2^{J/2} \psi(2^J t - n) \quad (2.16)$$

where $N_J = 2^J$.

The computation of the wavelet series coefficients using filter banks was studied by Mallat [22]. Wavelet expansion coefficients, a_n^j 's, at successive scales, called the Discrete Wavelet Transform (DWT), can be determined by a pyramid-like structure, called the pyramid algorithm, consisting of two filtering operations given as,

$$a_n^{j-1} = \sum_k a_k^j h(k - 2n) \quad \text{for all } j \quad (2.17)$$

where $h(n)$ is a discrete-time low-pass filter, and

$$b_n^{j-1} = \sum_k a_k^j g(k - 2n) \quad \text{for all } j \quad (2.18)$$

where $g(n)$ is a discrete-time high-pass filter.

In this algorithm the input signal, whose length is a power of two, is decomposed into a low-pass and a high-pass signal components by filters $h(-n)$ and

$g(-n)$ and then each part is subsampled by 2. In many cases, the low-pass output contains most of the information content of the original signal, whereas the high-pass output (usually called the detail signal) contains the difference between the true input and the reconstructed input which is obtained using only the low-pass outputs. Generally, higher order wavelets provide the low-pass outputs carrying more information than the high-pass outputs. So according to the pyramid algorithm for decomposition, we can represent $f(t)$ at the finest scale J as follows,

$$f(t) = \sum_{j=1}^{J-1} \sum_n b_n^j \phi_{2^j,n}(t) + \sum_n a_n^1 \psi_{2^1,n}(t) \quad (2.19)$$

The wavelet transform is of considerable use especially in signal processing applications such as speech processing [23], image processing [24], pattern recognition, edge detection, fractals etc. The kernel separability property in wavelet transform theory is an interesting feature that renders multidimensional wavelet transforms as a powerful tool for multidimensional signal processing.

2.2.3 The Wigner Distribution

Most of the TFRs from the Cohen class of distributions can be derived from the Wigner distribution, the most popular one and the most widely used [15].

The Wigner distribution (WD), was first introduced in 1932 by Wigner [25] and, it was reintroduced in 1948 by Ville for signal analysis applications [26]. The WD is an important tool for time-frequency analysis with no time-frequency resolution tradeoff. The WD satisfies many desirable mathematical properties including time and frequency marginals, covariance of time-shift, frequency-shift and scaling, energy conservation, realness, and Moyal's formula. The cross-Wigner

distribution of time signals $f(t)$ and $g(t)$ is defined as [27],

$$W_{f,g}(t, w) = \int_{-\infty}^{\infty} e^{-jw\tau} f(t + \tau/2) g^*(t - \tau/2) d\tau \quad (2.20)$$

Similarly the auto-Wigner distribution of a signal $f(t)$ is defined as [27],

$$W_f(t, w) = \int_{-\infty}^{\infty} e^{-jw\tau} f(t + \tau/2) f^*(t - \tau/2) d\tau \quad (2.21)$$

A useful dual expression for the WD in terms of Fourier transforms $F(w)$ and $G(w)$ of the signals $f(t)$ and $g(t)$, respectively can also be defined as ,

$$W_{F,G}(w, t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{j\xi t} F(w + \xi/2) G^*(w - \xi/2) d\xi \quad (2.22)$$

It should be noted that the WD is the only TFR which does not involve any arbitrary window function as other transforms do.

From equations 2.20 and 2.22, a useful symmetry relation between time and frequency domain definitions can be obtained as,

$$W_{F,G}(w, t) = W_{f,g}(t, w) \quad (2.23)$$

The cross-Wigner distribution for discrete-time signals $f(n)$ and $g(n)$ is defined as [28],

$$W_{f,g}(n, \theta) = 2 \sum_{k=-\infty}^{\infty} e^{-j2k\theta} f(n+k) g^*(n-k) \quad (2.24)$$

Similarly the auto-Wigner distribution of a discrete-time signal is defined as ,

$$W_f(n, \theta) = 2 \sum_{k=-\infty}^{\infty} e^{-j2k\theta} f(n+k) f^*(n-k) \quad (2.25)$$

Also, the WD in terms of Fourier spectra of discrete-time signals can be defined as follows,

$$W_{F,G}(\theta, n) = \frac{1}{\pi} \int_{-\pi}^{\pi} e^{j2n\xi} F(\theta + \xi) G^*(\theta - \xi) d\xi \quad (2.26)$$

1.2.3.1 Properties of the Wigner Distribution

The properties of WD for continuous time [27], (also for discrete time case [28]) can be summarized as follows:

•

$$W_{f,g}(t, w) = W_{g,f}^*(t, w) \quad (2.27)$$

Then it can easily be concluded that the WD of any function, whether real or complex, is real and the WD of a real signal is an even function of frequency as mathematically expressed below:

$$W_f(t, w) = W_f^*(t, w) \quad (2.28)$$

$$W_f(t, w) = W_{f^*}(t, -w) \quad (2.29)$$

These conclusions can be easily extended to the discrete-time case as follows:

$$W_{f,g}(n, \theta) = W_{g,f}^*(n, \theta) \quad (2.30)$$

$$W_f(n, \theta) = W_f^*(n, \theta) \quad (2.31)$$

$$W_f(n, \theta) = W_{f^*}(n, -\theta) \quad (2.32)$$

- The WD of time-shifted signals is also time-shifted as shown below,

$$W_{\Gamma_\tau f, \Gamma_\tau g}(t, w) = W_{f, g}(t - \tau, w) \quad (2.33)$$

where $\Gamma_\tau f = f(t - \tau)$.

This property applies to the discrete-time case as

$$W_{\Gamma_m f, \Gamma_m g}(n, \theta) = W_{f, g}(n - m, \theta) \quad (2.34)$$

- The WD of signals modulated with $e^{j\Omega t}$ is a frequency-shifted WD as shown below,

$$W_{M_\Omega f, M_\Omega g}(t, w) = W_{f, g}(t, w - \Omega) \quad (2.35)$$

where $(M_\Omega f)(t) = e^{j\Omega t} f(t)$.

This property applies to the discrete-time case as

$$W_{M_\beta f, M_\beta g}(n, \theta) = W_{f, g}(n, \theta - \beta) \quad (2.36)$$

- The WD of the signals that are both time-shifted and modulated by $e^{j\Omega t}$ is,

$$W_{M_\Omega \Gamma_\tau f, M_\Omega \Gamma_\tau g}(t, w) = W_{\Gamma_\tau M_\Omega f, \Gamma_\tau M_\Omega g}(t, w) = W_{f, g}(t - \tau, w - \Omega) \quad (2.37)$$

After some manipulations the equality given in equation 2.38 below for the WD at a specific time t and frequency w can be obtained,

$$W_{f,g}(t, w) = 2(\Gamma_{-t}M_{-w}f, R\Gamma_{-t}M_{-w}g) \quad (2.38)$$

where $(Rf)(t) = f(-t)$.

These properties can be expressed in the discrete-time case as follows:

$$W_{f,g}(n, \theta) = 2(\Gamma_{-n}M_{-\theta}f, R\Gamma_{-n}M_{-\theta}g) \quad (2.39)$$

- The WD for the sum of two signals is given as follows,

$$W_{f_1+f_2, g_1+g_2}(t, w) = W_{f_1, g_1}(t, w) + W_{f_1, g_2}(t, w) + W_{f_2, g_1}(t, w) + W_{f_2, g_2}(t, w) \quad (2.40)$$

and for auto-WD case,

$$W_{f+g}(t, w) = W_f(t, w) + W_g(t, w) + 2\text{Re}W_{f,g}(t, w) \quad (2.41)$$

These conclusions can be easily extended to the discrete-time case as follows:

$$W_{f_1+f_2, g_1+g_2}(n, \theta) = W_{f_1, g_1}(n, \theta) + W_{f_1, g_2}(n, \theta) + W_{f_2, g_1}(n, \theta) + W_{f_2, g_2}(n, \theta) \quad (2.42)$$

- The multiplication of WD by time variable t or frequency variable w can be expressed as ,

$$2tW_{f,g}(t, w) = W_{\Lambda f, g}(t, w) + W_{f, \Lambda g}(t, w) \quad (2.43)$$

$$2wW_{f,g}(t, w) = W_{Df, g}(t, w) + W_{f, Dg}(t, w) \quad (2.44)$$

where

$$(\Lambda.f)(t) = tf(t) \quad (2.45)$$

$$(D.f)(t) = \frac{1}{j}f'(t) \quad (2.46)$$

and for the discrete-time case, we have

$$2nW_{f,g}(n, \theta) = W_{\Lambda f, g}(n, \theta) + W_{f, \Lambda g}(n, \theta) \quad (2.47)$$

$$e^{j2\theta}W_{f,g}(n, \theta) = W_{\Gamma_{-1}f, \Gamma_1g}(n, \theta) \quad (2.48)$$

- Since the standard cross-WD definition is the FT of $f(t+\tau/2)g^*(t-\tau/2)$, the inverse Fourier transform of $W_{f,g}(t, w)$ will produce,

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} e^{jw\tau} W_{f,g}(t, w) dw = f(t + \tau/2)g^*(t - \tau/2) \quad (2.49)$$

which also leads to,

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} e^{jw(t_1-t_2)} W_{f,g}\left(\frac{t_1+t_2}{2}, w\right) dw = f(t_1)g^*(t_2) \quad (2.50)$$

For the special case of $t_1 = t_2 = t$ and $f = g$,

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} W_f(t, w) dw = |f(t)|^2 \quad (2.51)$$

which is the instantaneous signal power at time t .

Also, for the special case $t_1 = t, t_2 = 0$,

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} e^{j\omega t} W_{f,g}(t/2, \omega) d\omega = f(t)g^*(0) \quad (2.52)$$

which means that $f(t)$ can be obtained from the inverse Fourier transform of $W_{f,g}$ at time $t/2$.

- By taking the integral of both sides of equation 2.51, the energy of $f(t)$ in the time interval $t_a < t < t_b$ can be expressed in terms of the WD of f ,

$$\frac{1}{2\pi} \int_{t_a}^{t_b} \left[\int_{-\infty}^{\infty} W_f(t, \omega) d\omega \right] dt = \int_{t_a}^{t_b} |f(t)|^2 dt \quad (2.53)$$

The discussions in the last two properties can be applied to frequency domain definition of WD (in terms of spectra $F(\omega)$ and $G(\omega)$) producing similar results.

- By using Moyal's formula,

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_{f_1, g_1}(t, \omega) W_{f_2, g_2}^*(t, \omega) dt d\omega = (f_1, f_2)(g_1, g_2)^* \quad (2.54)$$

where the inner product of f and g is defined as,

$$(f, g) = \int_{-\infty}^{\infty} f(t)g(t)^* dt \quad (2.55)$$

for a special case $f_1 = f_2 = g_1 = g_2 = f$,

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_f^2(t, w) dt dw = \|f\|^4 \quad (2.56)$$

- If $f(t)$ and $g(t)$ are time-limited then the WD is also time-limited in the same interval i.e., if

$$f(t) = g(t) = 0, \quad t < t_a \text{ or } t > t_b \quad (2.57)$$

then

$$W_{f,g}(t, w) = 0, \quad t < t_a \text{ or } t > t_b, \quad \forall w. \quad (2.58)$$

For discrete-time case, if

$$f(n) = g(n) = 0, \quad n < n_a \text{ or } n > n_b \quad (2.59)$$

then

$$W_{f,g}(n, \theta) = 0, \quad n < n_a \text{ or } n > n_b, \quad \forall \theta. \quad (2.60)$$

- If $f(t)$ and $g(t)$ are band-limited, then the WD is also limited to the same frequency interval i.e, if

$$F(w) = G(w) = 0, \quad w < w_a \text{ or } w > w_b \quad (2.61)$$

then

$$W_{f,g}(t, w) = 0, \quad w < w_a \text{ or } w > w_b, \quad \forall t. \quad (2.62)$$

Since the WD for discrete-time signals is periodic with period π , If

$$F(\theta) = G(\theta) = 0, \quad \theta < \theta_a \text{ or } \theta > \theta_b \quad (2.63)$$

and $\theta_b - \theta_a > \pi$ then,

$$W_{f,g}(n, \theta) = 0, \quad \text{for } \theta_a < \theta < \theta_b - \pi \quad (2.64)$$

The WD is certainly an important tool for time-frequency analysis due to its long list of desirable properties presented so far. However, the method has also some drawbacks such as the cross-components that can interfere with the desired auto-components, the negative values and the noise propagation tendency.

Because of its quadratic nature, the WD includes interference terms (ITs) which complicates the interpretation of results. Consider two signals which are individually localized around (t_1, w_1) and (t_2, w_2) respectively and reaching to almost zero levels at the point $(\frac{t_1+t_2}{2}, \frac{w_1+w_2}{2})$. However, the cross-WD of these two signals will include the oscillatory cross terms (ITs) lying midway between the two signal terms, specifically at $t = \frac{t_1+t_2}{2}$ and $w = \frac{w_1+w_2}{2}$. Generally these nonzero WD values are called cross-components which are usually undesirable. When these cross-components interfere with the true signal components, undesired results occur. In this case, smoothing in time or frequency (or both) domains can be used to eliminate the undesirable cross-components at the expense of destroying the desirable properties of the WD to some extent.

Since the cross or interference terms of the WD are oscillatory, it is possible to attenuate them by means of smoothing in time or (and) frequency. The pseudo-Wigner distribution (PWD) is a short-time version of the WD which does smoothing only in the frequency direction by using a sliding analysis window [29]. The definition of the PWD for the discrete case is given as,

$$PWD_f(n, \theta) = 2 \sum_{k=-L}^L f(n+k) f^*(n-k) h^2(k) e^{-j2k\theta} \quad (2.65)$$

where $h(k)$ is a real-valued, even window of length $2L + 1$.

In general, the smoothed pseudo-Wigner distribution (SPWD) does additional smoothing in time, also to attenuate the ITs in that direction [30]. However, the SPWD does not satisfy most of the useful properties that the WD satisfies, especially the time and frequency marginals and the Moyal's formula. The discrete-time definition of the SPWD can be given in as,

$$SPWD_f(n, \theta) = 2 \sum_{l=-N+1}^{N-1} |h_{\theta, N}(l)|^2 \left[\sum_{k=-M+1}^{M-1} w_{n, M}(k) f_A(n+k+l) f_A^*(n+k-l) \right] e^{-j2l\theta} \quad (2.66)$$

where f_A is the analytic signal of f , $w_{n, M}$ is the time window and $h_{\theta, N}$ is the frequency window.

Since the WD of a signal satisfies the marginal properties, it can be, in some sense, interpreted as the distribution of the signal energy over time and frequency. However, the WD may assume negative values as well, leading to a contradiction with the energy interpretation. De Bruijn relates this behaviour of the WD to Heisenberg's uncertainty principle and states that appropriately taken averages

of the WD will always be positive [31].

As it can be easily deduced from the definition of the WD, even a small amount of localized noise in the original signal tends to spread to a much wider range in the resulting WD. Therefore, the noise performance of WD is poor if no care is taken for smoothing.

2.2.4 Generalized Class of TFRs

Quadratic time-frequency distributions (TFDs) were developed to describe the distribution of the signal energy in the time-frequency plane [15, 32]. However, it is known by uncertainty principle that no time-frequency distribution can exactly describe the localized time-frequency energy density, instead good approximations can be achieved. There are many TFDs with different forms and properties, and each TFD does satisfy the marginals and can be very useful tools for time-varying signal analysis. Furthermore it is shown to be possible to derive an infinite number of TFD from the general formula [15],

$$P(t, w) = \frac{1}{4\pi^2} \int \int \int e^{-j\theta t - j\tau w + j\theta\nu} \phi(\theta, \tau) f(\nu + \tau/2) f^*(\nu - \tau/2) d\nu d\tau d\theta \quad (2.67)$$

where $\phi(\theta, \tau)$ represents the arbitrary kernel function resulting in a different distribution for each specific choice. It is also possible to express the general form in terms of the input spectrum as follows ,

$$P(t, w) = \frac{1}{4\pi^2} \int \int \int e^{-j\theta t - j\tau w + j\tau\nu} \phi(\theta, \tau) F^*(\nu + \theta/2) F(\nu - \theta/2) d\nu d\tau d\theta \quad (2.68)$$

The properties of each distribution depends upon the corresponding kernel. If the kernel is independent of time and frequency, then the distribution is time and shift invariant. If the kernel is independent of the signal, then the distributions are bilinear.

Some of other known quadratic distributions and their kernels are given below for completeness,

- Rihaczek distribution

This distribution, which was first derived by Rihaczek [33], gives the complex energy spectrum of a signal as given below,

$$E_c(t, w) = \frac{1}{\sqrt{2\pi}} f(t) F^*(w) e^{-jw t} \quad (2.69)$$

- Page distribution

This distribution, which was derived by Page in 1952 [34], uses the running spectrum idea to indicate that the intensity at a specific frequency will be larger if that frequency is observed more frequently up to that time. The definition is given as,

$$P^-(t, w) = \frac{\partial}{\partial t} \left| \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t f(\tau) e^{-jw\tau} d\tau \right|^2 \quad (2.70)$$

- Choi-Williams distribution

This distribution, which was developed by Choi and Williams [35], is a smoothed WD that achieves partial attenuation of ITs while retaining many

desirable mathematical properties. Especially, it satisfies the marginals so can be interpreted as an energy distribution. The definition is as follows,

$$P(t, w) = \frac{1}{4\pi^{3/2}} \int \int \sqrt{\frac{\sigma}{\tau^2}} e^{-\sigma(\nu-t)^2/4\tau^2 - j\tau w} f(\nu+\tau/2) f^*(\nu-\tau/2) d\nu d\tau \quad (2.71)$$

where the parameter σ controls the amount of smoothing.

- The cone-kernel distribution

This distribution has some characteristic properties related to IT attenuation and time-frequency concentration. Since it contains significant amount of negative terms, as compared to the standard WD, mainly due to its different kernel function, it is not an energy distribution. The definition is given below as [32],

$$P_x(t, f) = \int_{\tau} \left[\int_{t'} \psi(t-t', \tau) x(t'+\tau/2) x^*(t'-\tau/2) dt' \right] e^{-j2\pi f\tau} d\tau \quad (2.72)$$

where $\psi(t, \tau)$ is the specific weighting function.

It should be noted that for all these TFRs, given a distribution, the kernel can be determined by Fourier inversion as [15],

$$\phi(\theta, \tau) = \frac{\int \int e^{j\theta t + j\tau w} P(t, w) dt dw}{\int e^{j\theta\nu} f(\nu + \tau/2) f^*(\nu - \tau/2) d\nu} \quad (2.73)$$

2.3 Application of the WD for Feature Extraction

So far, in this chapter, many important time-frequency methods for time-varying signal analysis have been reviewed. Among them, the WD is considered to be

more convenient to be used as a part of an electromagnetic target classification scheme mainly because of its distributed energy interpretation. It was also possible to use the WT, as an alternative, where the scale parameter s does not directly correspond to frequency. The WD's energy interpretation is however more natural and easier to apply especially for realistic systems.

We employ the standard WD for feature extraction instead of other alternatives such as PWD, SPWD or Choi-Williams distribution, which are obtained by some smoothing operations on the WD to reduce the cross-term problems. This is because all these smoothed distributions do not satisfy the marginals and so they do not have the distributed energy interpretation of standard WD.

It is well known that the available time-domain or frequency-domain data for an electromagnetic scatterer are aspect (i.e. the orientation of the scatterer with respect to transmitting and receiving antennas) and polarization dependent. Using such data directly, without any signal processing, in a target classification study requires a very large database which takes a very long processing time. Instead, the available data must be pre-processed for feature extraction or feature enhancement for faster and more accurate classification. In the resonance region, the most fundamental feature of an electromagnetic target is its set of complex natural resonance (CNR) frequencies (i.e. target's system poles) which are independent of both aspect and polarization of the available data [9]. However, extraction of CNR frequencies from measurement data is extremely noise sensitive, therefore a target classification scheme based on direct target pole comparison does not have much chance to work properly. It is yet possible to develop some other classification techniques where the discriminating power of target poles can

be utilized indirectly. An example of this indirect utilization can be carried out by using a selected subregion of the Wigner distribution output matrix. To be more specific, in the relatively late time portion of WD output matrix, the effect of damped natural oscillations due to target's CNR frequencies are much more pronounced. The effect of the forced part of the impulse response is dominant in the early time portion on the other hand.

For the WD computations in this thesis, for the specified frequency interval, and for all the sampled time points, the WD matrix of input impulse response is found. Then the $M \times N$ feature matrix (M frequency points, N time bands) is formed for each input by taking the time average of the WD data in specified time bands. By this time averaging, we reduce the dimension of feature matrix, which causes reduction of the complexity of neural network classifier to be used, and also we perform some smoothing in time resulting in the reduction of the oscillatory cross-components. However, this time averaging negatively affects the discriminating power of resultant feature vectors by decreasing the time resolution for the energy spectrum content of signal.

For example, for a perfectly conducting thin wire (length $L = 12\text{m}$, length to diameter ratio $L/d = 2000$), the backscattered impulse response data at 30° aspect angle are plotted in Figure 2.1. This time-domain data are obtained by taking the Inverse Fast Fourier Transformation (IFFT) of the windowed frequency domain data generated by a computer code based on the Moment Method computations [36]. This data were generated over the frequency range $[0, 256 \text{ MHz}]$ with 2 MHz frequency steps. Therefore, the corresponding time-domain signal (uniformly sampled at 2048 points) has a span of 500 nanoseconds with time steps (Δt)

being about 0.24426 nanoseconds. The magnitude of the signal is normalized such that its total energy in $[0, 500\text{nsec}]$ equals to unity.

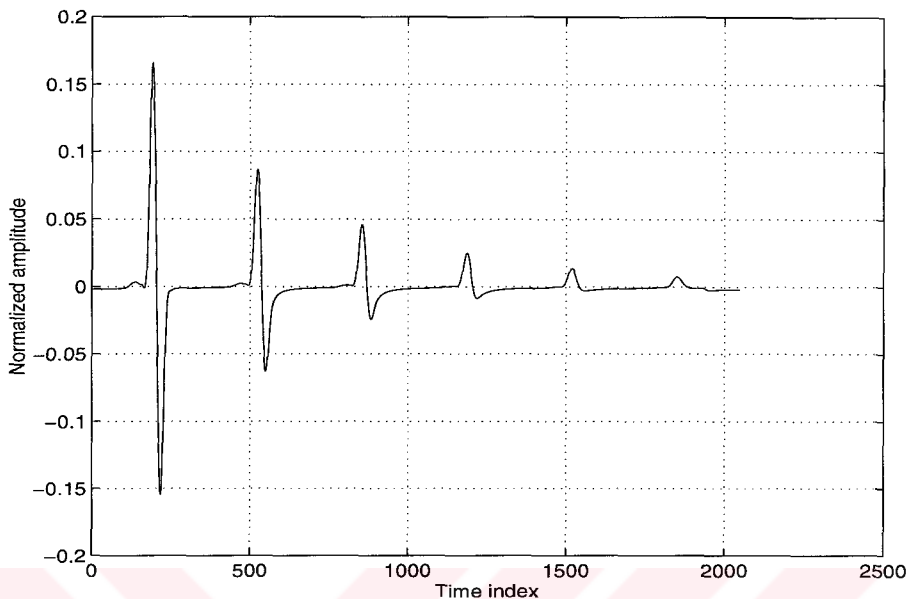


Figure 2.1: Backscattered impulse response of a thin wire at 30° aspect angle

The contour plot of the WD output matrix for this time-domain signal, taking only the positive WD outputs, is shown in Figure 2.2 where the time axis is sampled at 2048 points in the range $[0, 500\text{nsec}]$ and the frequency axis is sampled at 50 points in the range $[-64\text{MHz}, 64\text{MHz}]$.

In Figure 2.3 on the other hand, a partially time-averaged WD output is plotted where the time span is divided into ten bands. Each band corresponds to 204 time samples, i.e. to 50 nanoseconds. At each sample frequency, then, the WD values over each band are averaged. Since a conducting thin wire is a very high-Q target, its CNR frequencies are very strongly excited and are easily identified in Figures 2.2 and 2.3. In these figures, the 26th index corresponds to zero frequency. The first pair of CNR frequencies (approximately at 12MHz and -12MHz) appear around the frequency indices 22 and 30. It is possible to identify

the first five CNR frequency pair locations on Figures 2.2 and 2.3 just by looking at the peaks of the WD output.

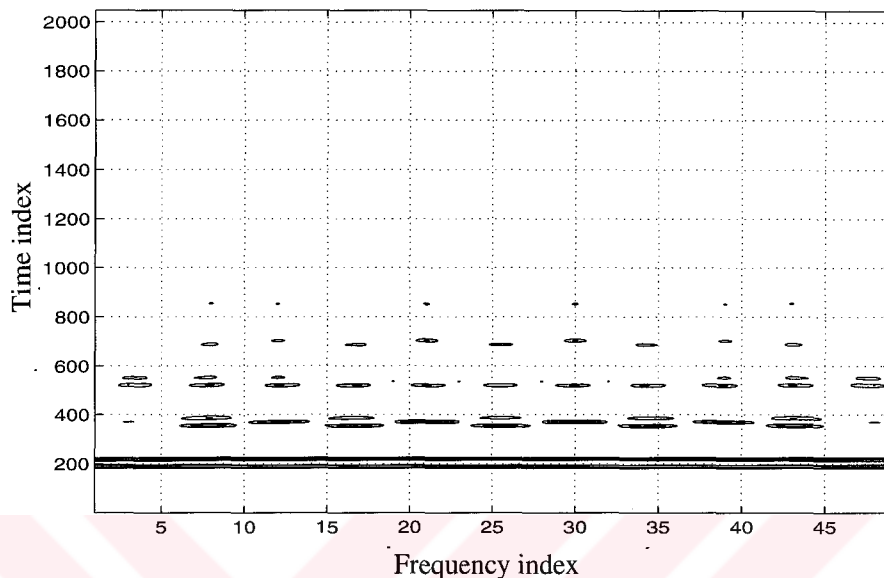


Figure 2.2: The contour plot of WD of the impulse response shown in Figure 1.1

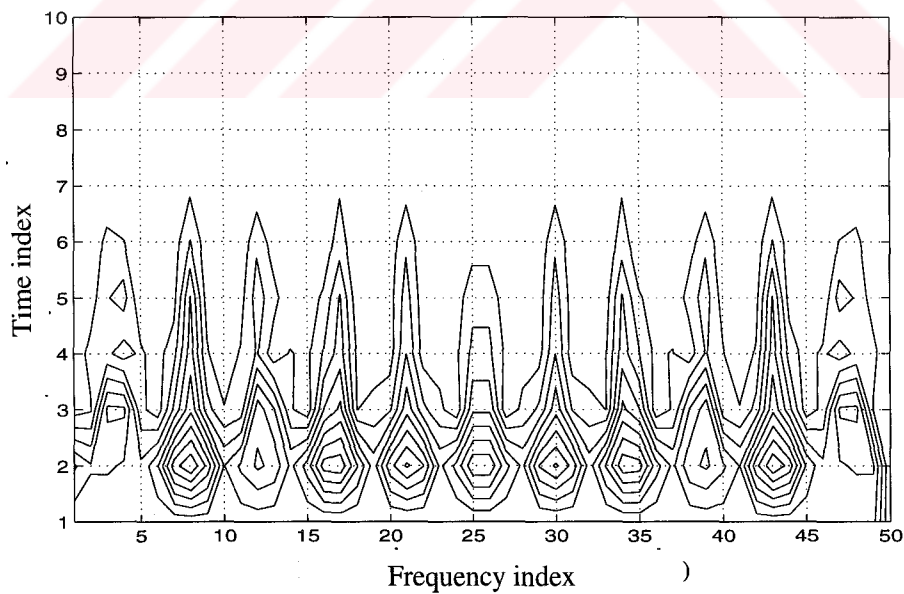


Figure 2.3: The contour plot of time-averaged WD of the impulse response shown in Figure 1.1

It is also possible to concentrate on a selected part of the WD outputs for additional feature enhancement purposes. Late time bands of the time-averaged WD output matrix, for instance, can be used in the further steps of a target classification system to emphasize the information carried by natural resonances as explained earlier. The selected feature vectors for 12m wire at 30° , 60° and 90° aspect angles, where specifically the 6th, 7th and 8th time bands are selected, are shown in Figures 2.4, 2.5 and 2.6.

As a result, the Wigner based time-localized energy spectrum data is used to extract a feature matrix from the impulse response signal of each target, which then will form the input of our target classification system.

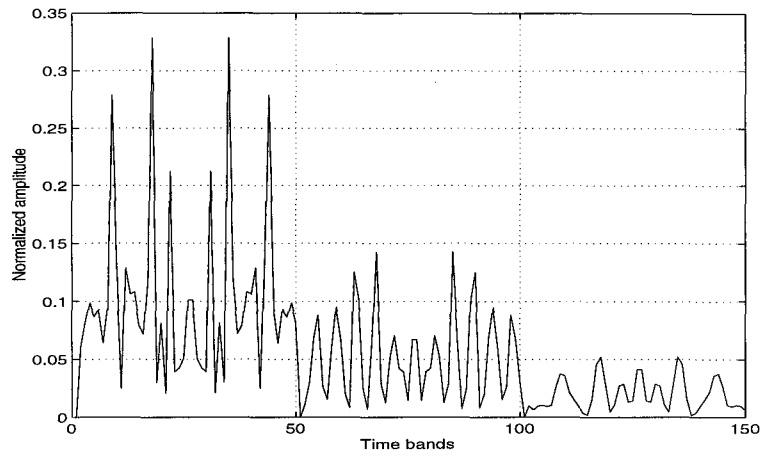


Figure 2.4: The selected feature vector for 12m wire at 30° aspect angle

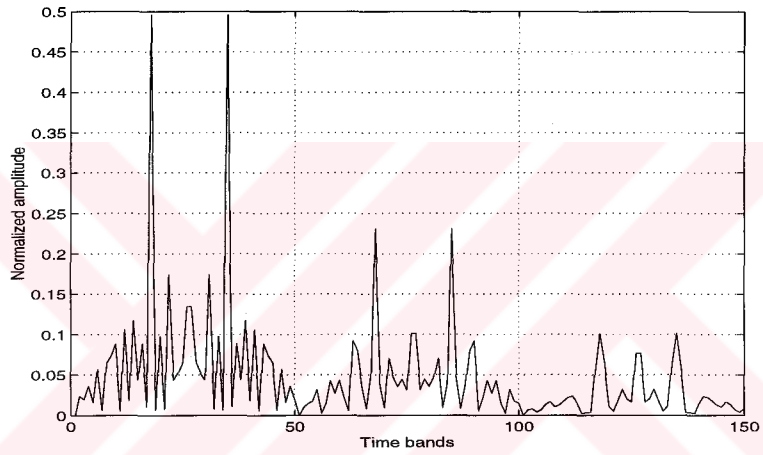


Figure 2.5: The selected feature vector for 12m wire at 60° aspect angle

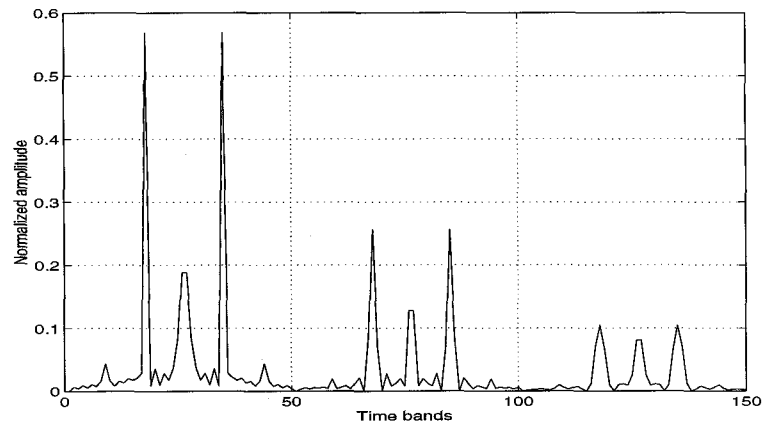


Figure 2.6: The selected feature vector for 12m wire at 90° aspect angle

CHAPTER 3

NEURAL NETWORKS AND CLASSIFICATION

3.1 Introduction

A neural network (NN) is a signal processing device in which the design and the functioning structure are motivated by those of a human brain. These computing devices are designed to realize some of the learning processes that are believed to take place in the brain. The basic idea behind a NN is to provide a high computational power by a large number of highly inter-connected parallel processing elements which are called neurons.

Two important information processing capabilities of a neural network (NN) [37], its massively parallel processing ability and its ability to learn and generalize, make it very attractive for many different practical applications such as pattern recognition, image recognition, radar target detection and classification, control, speech processing, biomedical instrumentation and financial forecasting.

The learning process through which a NN adapts itself by adjusting some parameters to its environment is one of the most important features of NNs. Usually learning is accomplished by exposing the NN to the training inputs that are randomly drawn from the environment. After learning, the representation of the environment is encoded by the NN into the weights and thresholds.

The flexibility of NN makes them able to discover more general relationships in data than traditional statistical models can. Even in the case of incompletely defined problems, the NNs are expected to produce better results as compared to many conventional approaches, due to the fact that NNs do not need much information about the specific problem modelling.

Generally, the network topology, neuron characteristics and learning or adaptation rules specify the NN type. It is possible to categorize NN classifiers by their learning type [38]: supervised, where the classes of available training inputs are known; or unsupervised, where no class information is provided during training. The most popular supervised NN classifiers are multilayer perceptrons (MLP), Hopfield networks and Hamming networks. The Carpenter-Grossberg classifiers (ART) and the Kohonen self-organizing feature maps (SOFM) are, on the other hand, well known examples of unsupervised NNs. In this chapter, the principles of only MLP and SOFM techniques will be reviewed, which are utilized in this thesis for target classification purposes.

3.2 Multilayer Perceptron

Multilayer perceptrons (MLPs), or multilayer feedforward networks, represent the generalization of the single-layer perceptron which is the simplest form of a NN

consisting of a single neuron with adjustable synaptic weights and a threshold [37]. MLP consists of a set of input units (input layer), one or more intermediate layers of processing elements (hidden layers) and an output layer of processing elements. The graph of a MLP with two hidden layers is shown in Figure 3.1,

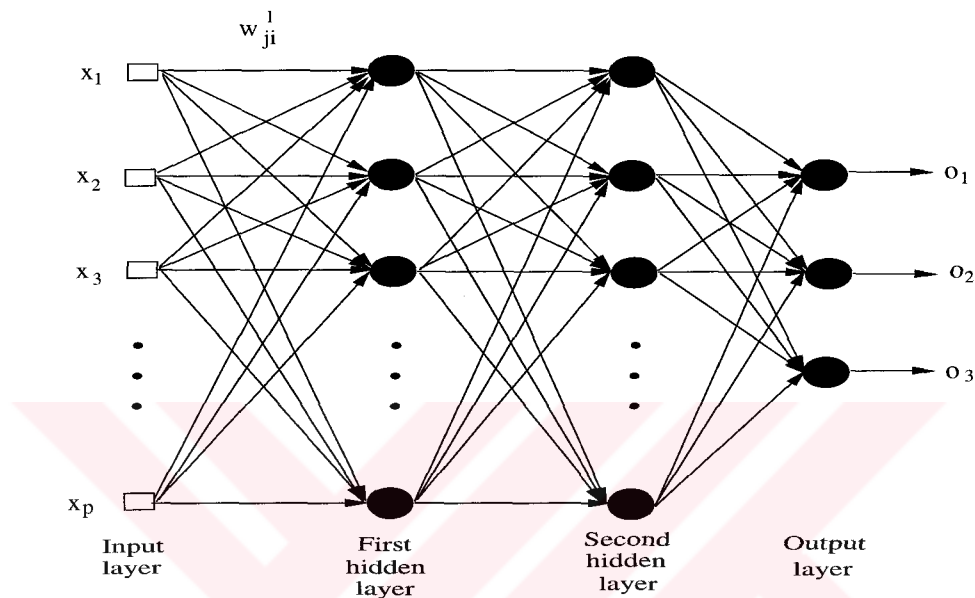


Figure 3.1: MLP with two hidden layers

The operation of a neuron which is the fundamental information processing unit of a NN can be described as follows: Each possible input of a neuron is multiplied by the connection weights and then linear combination of the weighted inputs is compared to a certain threshold value. The result is then applied as an input to a nonlinear activation function (generally sigmoidal function), where its nonlinear model is shown in Figure 3.2,

Mathematically, the output of a neuron can be expressed as,

$$y_j = \psi\left(\sum_{i=1}^n w_{ji}x_i - \theta_j\right) \quad (3.1)$$

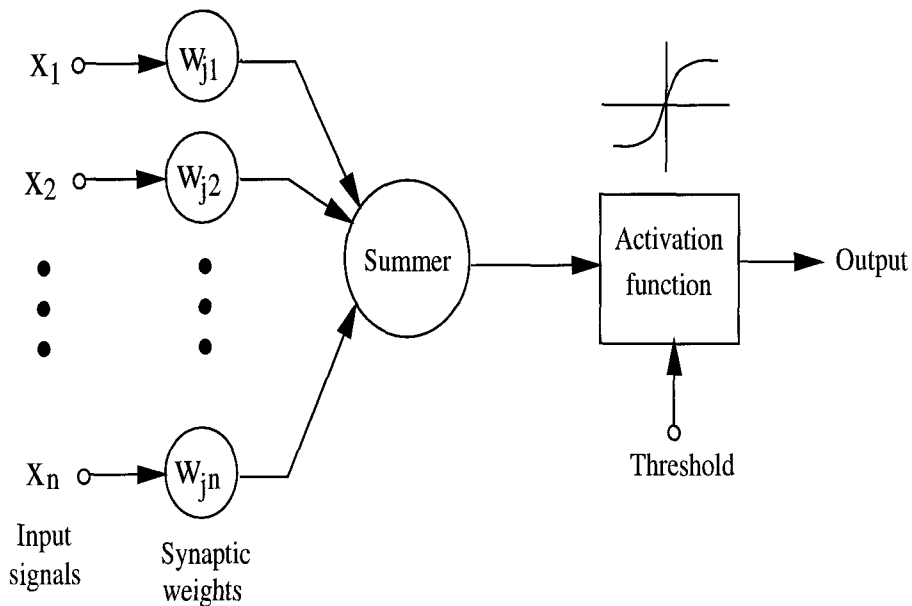


Figure 3.2: Single neuron model

where x_i s are inputs, w_{ji} s are connection weights, θ_j is the threshold and ψ is the nonlinear activation function.

An important result was proved by Kolmogorov [39] that an MLP with a single hidden layer with total number of $N(2N + 1)$ neurons is able to approximate any continuous function of N variables, i.e. it is able to form arbitrarily complex decision regions.

The operation of MLP consists of two distinct passes through layers. In forward pass, first the input signal is presented, then each neuron output is computed and finally outputs at the output layer of the network are obtained. In backward pass, the error at the output layer, which is the difference between the actual output and the desired output, is computed and propagated back to the input layer. During the propagation of errors from the output layer to input layer, the weights at each layer are adjusted to minimize the mean squared error between the actual output and the desired output.

The back-propagation training algorithm [37, 40] is an iterative gradient descent technique to minimize a cost function of the difference between the desired and the actual network outputs. The least mean square (LMS) algorithm is the special case of the back-propagation algorithm. Mathematically, the steps of the back-propagation algorithm can be summarized as follows :

- All weights and threshold levels of the network are set to uniformly distributed small random numbers. In case of available prior information, it's better to use that information for guessing initial values of free parameters. Otherwise natural choice of small random numbers prevents saturation of neurons and may help the network to converge faster.
- At each learning iteration, a randomly selected epoch of training examples from a training set, including the input and the desired output values, is presented to the network
- The operation of each neuron during forward propagation is defined as follows :

$$v_j^{(l)}(n) = \sum_{i=0}^p w_{ji}^{(l)}(n) o_i^{(l-1)}(n) \quad (3.2)$$

where $o_i^{(l)}(n)$ is the neuron output, $v_j^{(l)}(n)$ is the summer output in a neuron and $w_{ji}^{(l)}(n)$ is the connection weights for the lth layer at nth iteration.

For any differentiable nonlinear activation function f ,

$$o_j^{(l)}(n) = f_j(v_j^{(l)}(n)) \quad (3.3)$$

Error at the output neuron j is computed as,

$$e_j(n) = d_j(n) - o_j(n) \quad (3.4)$$

where d_j is the corresponding desired signal.

The average of sum of squared errors at the output of the network is,

$$E_{av} = (1/2N) \sum_{n=1}^N \sum_{j=1}^q e_j^2(n) \quad (3.5)$$

where q is equal to total number of output neurons.

- During the back-propagation of errors, the local gradients are first computed using equations 3.6 and 3.7, which result from the minimization of network error with respect to weights using $(\frac{-\partial E}{\partial v_j})$,

if o_j is an output unit,

$$\delta_j^{(l)} = (d_j - o_j) f_j'(v_j^{(l)}) \quad (3.6)$$

if o_j is not an output unit,

$$\delta_j^{(l)} = f_j'(v_j^{(l)}) \sum_k \delta_k^{(l+1)} w_{kj}^{(l+1)} \quad (3.7)$$

From the generalized delta rule, for the learning rate η and momentum constant α , the weights in layer l of the network are adjusted as,

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha[w_{ji}^{(l)}(n) - w_{ji}^{(l)}(n-1)] + \eta \delta_j^{(l)}(n) o_i^{(l-1)}(n) \quad (3.8)$$

- Continue with the second step, until the weight values stabilize and the E_{av} has an acceptably small value. The learning rate parameter can be dynamically adjusted during training by using the error gradient search algorithms (such as 1-D gradient search algorithm). Generally, as the number of iterations increases, the learning rate and the momentum constant need to be decreased either linearly or being inversely proportional to time.

Generally for MLPs the error surface over weight space may include some local minima points, and the weights are adapted in the direction of the gradient of this error surface [37]. The use of momentum constant emphasizes the past weights while adjusting the weights at the current step. Use of the adaptive learning rate together with the momentum constant can prevent the network to get stuck into a local minima.

MLPs trained with the back-propagation algorithm have been proven to have advantages over the classical methods especially in detection and classification applications [11, 41].

3.3 Self-Organizing Map

The self-organizing maps (SOM), which are introduced by T. Kohonen [42, 43, 44], are one of the most important models of unsupervised learning NN class and they can be used in applications such as vector quantization, speech processing, robotics and control, statistical pattern recognition, radar classification and image compression [45].

The SOM algorithm creates a mapping from a high dimensional vector space

of real numbers \mathbb{R}^n onto a two-dimensional lattice of points, where in addition to dimensionality reduction, metric ordering relations of input samples can be visualized [46]. It is also possible to define SOM as a nonlinear projection of the probability density function of the input data onto the two-dimensional plane. As shown in Figure 3.3, a SOM generally has one input layer and one output layer.

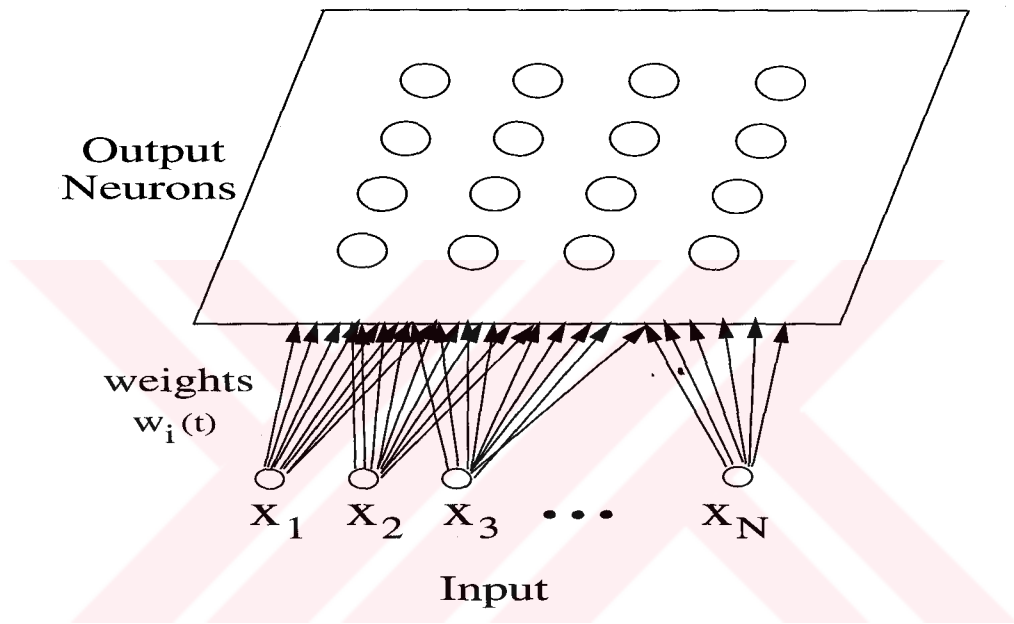


Figure 3.3: SOM Network Model

During the operation of the SOM, the similarity of the input vector and the weight vector of a SOM unit is measured in terms of the Euclidian distance. Then the weight vector of the winning unit, which is the closest one to the input vector, and that of each unit in the neighborhood of the winning unit is updated in the direction of the input. This procedure is applied iteratively for all randomly drawn input vectors and each time the weight vectors better approximate the input distribution. Finally all weights are stabilized and global ordering results.

Mathematically the steps of the general SOM training algorithm is given as follows [37]:

- First, the assignment of weight vectors to small random numbers is accomplished.
- Then, randomly drawn input vectors from an input distribution are iteratively presented to the network.
- The best matching output unit (winner) is determined by the following equation,

$$\alpha = \operatorname{argmin}_{(i)} \|x(t) - w_i(t)\| = \operatorname{argmin}_{(i)} \sqrt{\sum_j (x_j - w_{ij})^2} \quad (3.9)$$

where α is the index of the winner and $\|\cdot\|$ denotes the Euclidian norm.

- Next, the weight vectors are adapted according to

$$w_i(t+1) = w_i(t) + N_i(t)\eta(t)[x(t) - w_i(t)] \quad (3.10)$$

where $N_i(t)$ is the neighborhood function around the winning unit and $\eta(t)$ is the learning rate as a function of time.

- Continue with the second step, until the weights are stabilized.

The size of the neighborhood function N is a function of time, initially the neighborhood includes all the output neurons, but it shrinks monotonically with time. Similarly, the value of the learning rate η is a function of time and decreases, usually linearly or inversely, with time as shown below,

$$\alpha(t) = \alpha(0)(1 - t/\maxrep) \quad (3.11)$$

$$\alpha(t) = \alpha(0)\frac{k}{k+t} \quad (3.12)$$

where k is a constant that depends on \maxrep , which is the maximum number of iterations.

As stated by Kohonen [47], the aim of the SOM algorithm is to find the weight vectors which simultaneously approximate the distribution of the input data and preserve the topological ordering of input data.

Originally SOM is an unsupervised learning algorithm, however, we add some sort of supervision to its natural clustering ability in order to use it effectively in one of our classification schemes. This modification will be discussed in detail in Chapter 5 while presenting the related applications.

3.4 Applications of MLP and SOM for Classification

MLPs using the back-propagation training algorithm are the most widely applied type of NNs in the area of classification. So we first use an MLP to achieve classification using a set of synthetic data for perfectly conducting spheres of various diameters and a set of measurement data for five different models of electroplated small scale airplanes. To compare the classification results, in the first step only the decimated time-domain impulse response signals (without feature extraction) are used as the input of an MLP, in the second step the time-localized energy spectrum feature matrices extracted from the input data are used at the input of an MLP.

In the first case, without using feature extraction, where the decimated (512 sample points) impulse response data of spheres make inputs of an MLP, the parameters of the MLP trained with the backpropagation algorithm can be listed as follows:

- MLP layer dimensions :

Number of neurons in the input layer = 512

Number of neurons in the first hidden layer = 250

Number of neurons in the second hidden layer = 75

Number of neurons in the output layer = 4

- Training phase parameters:

Learning rate ($\eta(t)$) : $\eta_0 = 0.2$ at $t=0$ (for $t > 0$, $\eta(t)$ decreases inversely with time)

Momentum constant (α) = 0.1

Maximum number of repetitions in training = 30000

- Training and test data size :

Number of input data sets used for training = 13 (62%)

Number of input data sets used for testing = 8 (38%)

In the second case, using extracted feature matrices, the characteristics of the MLP used in simulations, both for sphere class and plane class, are as follows,

For sphere class:

- MLP layer dimensions :

Number of neurons in the input layer = 150

Number of neurons in the first hidden layer = 75

Number of neurons in the second hidden layer = 25

Number of neurons in the output layer = 2 (binary output)

- Training phase parameters:

Learning rate ($\eta(t)$) : $\eta_0 = 0.3$ at $t=0$ (for $t > 0$, $\eta(t)$ decreases inversely with time)

Momentum constant (α) = 0.1

Maximum number of repetitions in training = 10000

- Training and test data size:

Number of input data sets used for training = 13 (62%)

Number of input data sets used for testing = 8 (38%)

For plane class:

- MLP layer dimensions :

Number of neurons in the input layer = 150

Number of neurons in the first hidden layer = 100

Number of neurons in the second hidden layer = 30

Number of neurons in the output layer = 5

- Training phase parameters:

Learning rate ($\eta(t)$) : $\eta_0 = 0.35$ at $t=0$ (for $t \geq 0$, $\eta(t)$ decreases inversely with time)

Momentum constant (α) = 0.1

Maximum number of repetitions in training = 20000

- Training and test data size :

Number of input data sets used for training = 17 (55%)

Number of input data sets used for testing = 14 (45%)

The results of simulations and discussions for classification by MLP are presented in Chapter 5.

We propose a different classification scheme using a modified SOM to classify the time-averaged WD feature matrices of two groups of input data.

In practice, there are two phases of SOM training: In the first phase, the weight vectors of the SOM units are ordered and in the second phase the weight vectors find their correct values. In our simulations, initial neighborhood in the first phase is usually chosen to include all output neurons and initial learning rate is generally taken near 0.1, while in the second phase initial neighborhood is chosen to include neurons up to a distance of 2 or 3 and initial learning rate is taken smaller than 0.05 for fine tuning of SOM output values.

During training, different classes of training data are forced to different regions of the output map by initially setting their winning neuron positions. So by using forced winner training algorithm, the approximate locations of clusters

corresponding to each class are purposely set on the map. What is meant by *purposely* is that training data of classes which are more difficult to classify are set farther positions on SOM output map in order to reduce false classification rate. In our simulations, about 60 percent of the available data are used for training the NN and the remaining 40 percent are used for testing. The approximate class boundaries on the SOM output map can be determined after the training stage is completed by closely examining the clusters. Accuracy of these boundaries depends on the absolute size of the training data set.

After determining the class to which the test data belong, we have shown that it is also feasible to determine an aspect angle range for the data by means of additional SOMs for each class. These new SOMs can be similarly trained by forced winner algorithm for aspect angles between 0° (back-scattered response) and 180° this time within each specific class.

The results of simulations and discussions for the classification scheme using SOMs are presented in Chapter 5.

In addition to MLP-only and SOM-only classifiers, in the course of our research, we have also tried using two different combinations of MLP and SOM algorithms as follows :

In the first one, the classification system consists of the cascade of an unsupervised SOM and a supervised MLP networks which is shown in Figure 3.4. In this system, the output states of SOM after each training are sent to the MLP block to be used as input states of the MLP together with the class information. Then MLP is trained with the backpropagation algorithm to adjust its weights. In testing stage, only the last sets of weights of SOM and MLP are needed.

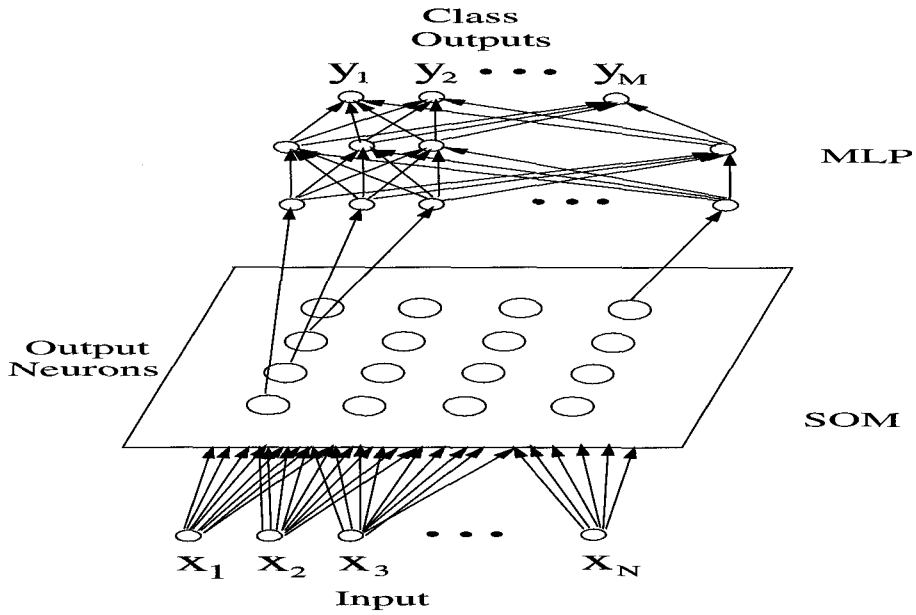


Figure 3.4: SOM+MLP classification system model

The second one is a slightly modified version of the first one, where a feedback from MLP outputs to SOM inputs is provided for every input pattern excluding the first one during the training phase. The existence of this feedback enables the system to contribute the effect of each pattern of input to the next one.

Other than these MLP/SOM combinations, we also designed a SOM-only classifier with internal feedback between the input data patterns during the testing stage. In this case, the SOM is trained as originally by using the training input data consisting of a fixed number of patterns. But in the testing stage, the SOM outputs of a current pattern are increased by the amount which is equal to a weighted sum of output values from the previous pattern if these values come from the neurons which fall in the current neighborhood of the SOM.

The results of this SOM classifier with internal feedback as well as the results of SOM/MLP combinations are not found satisfactory as compared to the results of the forced winner SOM classifier for the specific electromagnetic target clas-

sification problem that we have concentrated on in this thesis. Therefore, these last three approaches are left as alternative ideas for some other NN applications beyond the scope of our research.



CHAPTER 4

PRINCIPAL COMPONENT ANALYSIS AND CLASSIFICATION

4.1 Introduction

Principal Component Analysis (PCA) is a well-known classical statistical method which was first introduced by Pearson in 1901 [37]. PCA, or the Karhunen-Loeve expansion in stochastic theory [48], is an important feature extraction technique which is based on the eigenvectors of the correlation or covariance matrices. PCA computes the optimal description of the data sets by finding the best subspace that maximizes the projection of the input patterns onto principal axis. This linear transform has many applications in different scientific areas especially for data analysis, data compression, dimensionality reduction, and feature extraction since it is the optimum method regarding the information loss in the input signal.

The PCA method is primarily a data analysis technique which obtains linear transformations of a group of correlated variables such that certain optimal

conditions are achieved. In addition, PCA is an effective method for feature extraction especially when the input data consist of few clusters, since generally the major principal axes found by PCA can separate these clusters well. After feature extraction by using PCA, there are usually fewer features extracted than the number of inputs, so in this way, data reduction can be achieved.

4.2 Principal Component Analysis

The PCA is based on the statistical representation of a random variable. Let \mathbf{x} be an input matrix whose rows $x_i, i=1, \dots, n$ are randomly selected data vectors

$$\mathbf{x} = (x_1, \dots, x_n)^T \quad (4.1)$$

where the mean of \mathbf{x} is $\mu_{\mathbf{x}} = [\mu_{x_{ij}}]$ where $\mu_{x_{ij}} = E\{x_{ij}\}$.

Then the covariance matrix of \mathbf{x} is defined as the product of $\mathbf{x} - \mu_{\mathbf{x}}$ with its transposed form as follows [37],

$$\mathbf{C}_{\mathbf{x}} = E\{(\mathbf{x} - \mu_{\mathbf{x}})(\mathbf{x} - \mu_{\mathbf{x}})^T\} \quad (4.2)$$

The elements c_{ij} of the matrix $\mathbf{C}_{\mathbf{x}}$ represent the covariances between the random variable components x_i and x_j , which indicates that a linear relationship exists between these two variables and the strength of that relationship is represented by the correlation coefficient $p_{ij} = c_{ij} / \sqrt{c_{ii}c_{jj}}$. It is easily seen from its definition that $\mathbf{C}_{\mathbf{x}}$ is a symmetric matrix. The diagonal element c_{ii} of the matrix gives the variance of the component x_i , which shows the spread of the component values around its mean. If $c_{ij} = 0$, then the data components x_i and x_j are

uncorrelated.

For the covariance matrix \mathbf{C}_x , an orthogonal basis can be calculated from its eigenvalues and eigenvectors. The solutions of the equation

$$C_x u_i = \lambda_i u_i, \quad i = 1, \dots, n \quad (4.3)$$

gives the eigenvectors (u_i 's) and the corresponding eigenvalues (λ_i 's).

Generally, distinct eigenvalues can be found from the solutions of the characteristic equation given below,

$$|\mathbf{C}_x - \lambda \mathbf{I}| = 0 \quad (4.4)$$

where \mathbf{I} is the identity matrix of the same order as \mathbf{C}_x and $|\cdot|$ represents the determinant of the matrix ($\mathbf{C}_x - \lambda \mathbf{I}$).

When the data vector has a large number of components, solving eigenvalues and corresponding eigenvectors from Equation 4.4 becomes difficult. In this case, one good solution is to use a PCA neural network to compute eigenvectors.

After determining the eigenvalues and the corresponding eigenvectors, it is possible to find directions in which the input data set has the most significant amount of energy by putting the eigenvalues in a decreasing order and constructing an ordered orthogonal basis with the first eigenvector having the direction of largest variance of the data. It is possible to know approximately how much of the energy is concentrated along the specific eigenvector by comparing the values of individual eigenvalues to the sum of the all eigenvalues.

By using the matrix \mathbf{K} , whose row vectors are the eigenvectors of the covariance matrix \mathbf{C}_x , a data matrix \mathbf{x} can be transformed to points in the orthogonal coordinate system of eigenvectors as [49],

$$\mathbf{y} = \mathbf{K}(\mathbf{x} - \mu_x) \quad (4.5)$$

where the data matrix \mathbf{x} is projected onto a subspace defined by the orthogonal basis of computed eigenvectors.

The reconstruction of the original data matrix \mathbf{x} can be done by a linear combination of the orthogonal basis vectors, using the property of an orthogonal matrix $\mathbf{K}^{-1} = \mathbf{K}^T$ as below,

$$\mathbf{x} = \mathbf{K}^T \mathbf{y} + \mu_x \quad (4.6)$$

The data matrix \mathbf{x} can also be represented approximately by only some of the eigenvectors of the orthogonal basis instead of using all the eigenvectors of the covariance matrix. Then using the matrix \mathbf{K}_M , which has the first M eigenvectors as its rows, a similar transformation is defined as [49],

$$\mathbf{y} = \mathbf{K}_M(\mathbf{x} - \mu_x) \quad (4.7)$$

with a similar reconstruction rule,

$$\mathbf{x} = \mathbf{K}_M^T \mathbf{y} + \mu_x \quad (4.8)$$

Using equations 4.7 and 4.8, the original data matrix \mathbf{x} can be projected onto the coordinate axes of dimension M and then can be reconstructed with

a minimum mean squared error between the transformed data and the original data.

4.3 Some Useful Properties of Principal Components

Some properties of principal components, which are especially useful when using PCA in practical applications, are explained as follows [14]:

Let \mathbf{x} be a vector, then

- After transforming a set of variables \mathbf{x} by a linear transformation $\mathbf{y} = \mathbf{K}(\mathbf{x} - \mu_{\mathbf{x}})$, whether \mathbf{K} is orthonormal or not, it is possible to determine the covariance matrix of the new variables, let \mathbf{C}_y , directly from the covariance matrix of the original data set \mathbf{C}_x as follows,

$$\mathbf{C}_y = \mathbf{K}^T \mathbf{C}_x \mathbf{K} \quad (4.9)$$

Note that the orthonormality of \mathbf{K} is not a sufficient condition for the transformed variables to be uncorrelated. Only the characteristic vector solution in equation 4.9 will produce a diagonal matrix \mathbf{C}_y resulting uncorrelated new variables.

- Especially in practice, there are two generalized measures giving the overall variability of the data set.

The first measure is the determinant of the covariance matrix, $|\mathbf{C}_x|$, which is called the generalized variance. The quantity $\sqrt{|\mathbf{C}_x|}$ is proportional to the area or volume generated by the data set.

The second measure is the sum of the variances of the variables, which is called the trace of \mathbf{C}_x , as

$$T(\mathbf{C}_x) = x_1^2 + x_2^2 + \dots + x_n^2 \quad (4.10)$$

An important property of PCA is that the variability determined by either measure is preserved. For the first measure this property means that the determinant of the original covariance matrix \mathbf{C}_x is equal to the product of the characteristic roots, which are the variances of the principal components, as below,

$$|\mathbf{C}_x| = l_1 l_2 \dots l_n \quad (4.11)$$

Also for the second measure, this property means that the sum of the original variances is equal to the sum of the characteristic roots as follows,

$$x_1^2 + x_2^2 + \dots + x_n^2 = l_1 + l_2 + \dots + l_n \quad (4.12)$$

- Another useful measure is the correlation of each principal component with each of the original variables. For example, the correlation of the i 'th principal component y_i with the j 'th original variable x_j is given as,

$$p_{yx} = \frac{k_{ji} \sqrt{l_i}}{\sqrt{c_{jj}}} \quad (4.13)$$

where k_{ji} is constant. So, the principal component corresponding to higher eigenvalue is more highly correlated with the original variables.

We have seen that by choosing a fixed number of eigenvectors and the corresponding eigenvalues a consistent representation of the input data, which preserves varying amount of energy of the original input data depending on the chosen number of eigenvectors, can be obtained. Alternatively, it is possible to fix the amount of energy and vary the number of eigenvectors used in transformation. So there is a trade-off between the reduction of input data dimension and preserving much more of the information content of the original data. By using PCA this trade-off can be adaptively controlled.

In practical cases, generally two methods are applied. The first one employs nonlinear iterative partial least squares algorithm and the second one employs successive average orthogonalization algorithm. Both of these methods sequentially finds the individual principal components from the largest to the smallest.

The PCA approach has important applications in discriminant analysis, classification and clustering [50]. Since the PCA can reduce a system of correlated variables to a system of smaller number of new variables, the desired features of the available row data can be emphasized by means of this process.

4.4 Applications of PCA for Classification

In Chapter 2 and Chapter 3, we shortly discussed the applications of TFRs as well as two important neural network methods to the problem of feature selection and classification of radar data. Our purpose here is to apply also the well-known classical statistical method PCA to this same problem and then we will be able to compare the results obtained by three different methods using neural networks and the theory of statistics.

After obtaining the time-frequency localized energy spectrum matrix of each input data to be classified in a preprocessing stage as explained in Chapter 2, it's possible to apply PCA to this feature matrix to find its principal components in the eigenspace.

From the results of PCA analysis, we decided to extract only the major eigenvector corresponding to the most dominant eigenvalue since all the other eigenvalues are found negligible compared with the leading one. Here, the choice of eigenvectors from the result of PCA analysis depends on the construction of a time-averaged WD matrix whose independent features specify the number of dominant principal components . In using PCA, our aim is to decompose a feature matrix data set into a simpler data set by using the largest principal component which explains most of the variance in the feature matrix data set.

As an example, the PCA approach is applied to feature matrix data of three conducting spheres of diameters 7 inches, 10 inches and 20 inches. (These feature matrix data were obtained from the Wigner transformation of the impulse responses of these spheres at 120° aspect angle.) As indicated above, only the most dominant eigenvalues are taken into account in the PCA method to obtain the resulting dominant eigenvectors plotted in Figures 4.1 through 4.3 for these three spheres.

Then for classification, we simply compute the Euclidian distance between the new test vector and the known vectors of database.

The results of simulations and discussions for the classification scheme using PCA and these two different classification measures are presented in Chapter 5.

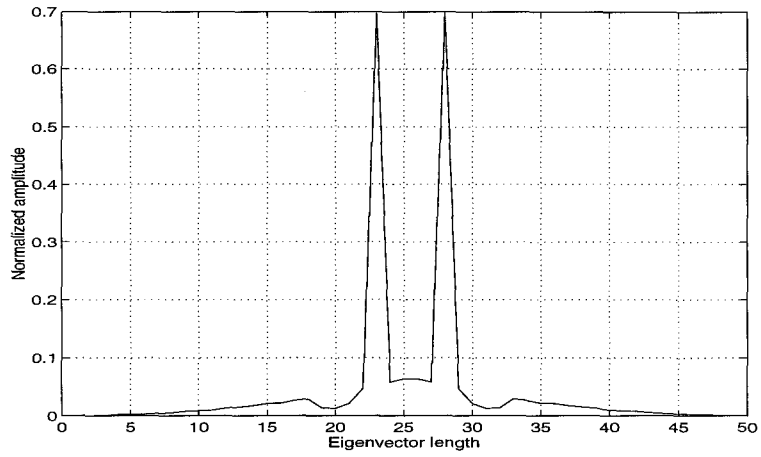


Figure 4.1: PCA output for 7 inch dia. sphere at 120° aspect angle

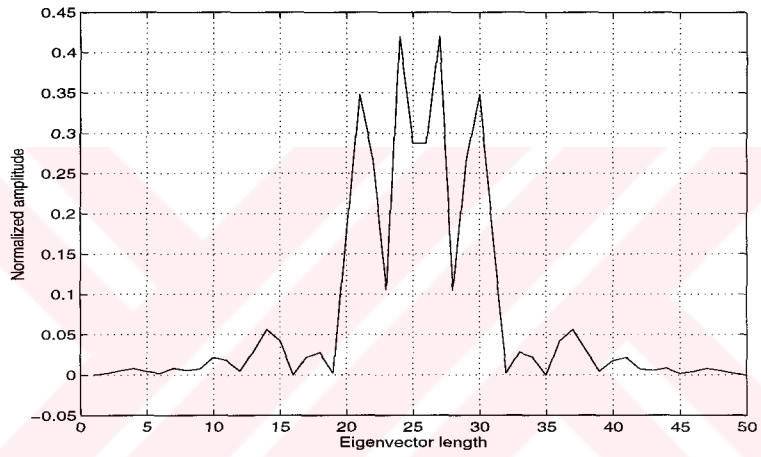


Figure 4.2: PCA output for 10 inch dia. sphere at 120° aspect angle

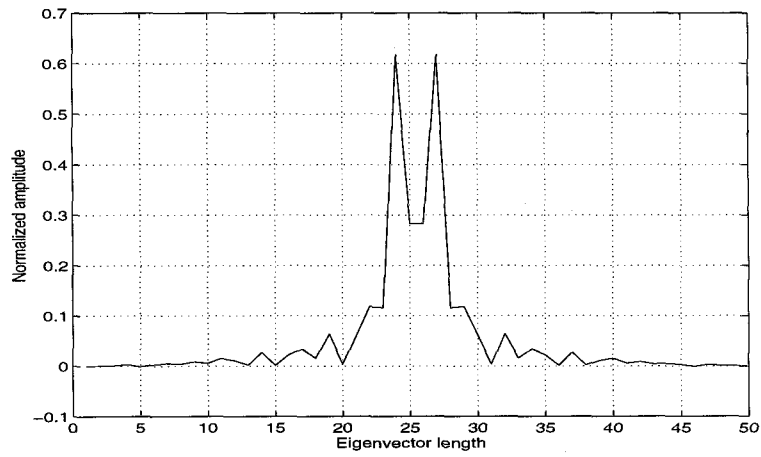


Figure 4.3: PCA output for 20 inch dia. sphere at 120° aspect angle

CHAPTER 5

APPLICATIONS OF TARGET CLASSIFICATION

5.1 Introduction

The main objective of this thesis is to design a new electromagnetic target classification system which can be operated at a reasonably high correct decision rate even for a modest size of database. Besides, a short response time in the decision phase and robustness to noise are the other important requirements expected to be met by this classification system.

As a result of our studies, several target classifier prototypes are shown to be feasible as demonstrated in the simulation examples of this chapter. In these alternative classifiers, various combinations of neural networks (MLPs and SOMs), the Wigner TFD and the PCA method are utilized. The advantages and/or disadvantages of these candidate classification systems are going to be discussed in the simulation examples.

5.2 Preliminary Information on the Classes and the Database

Basically two classes of electromagnetic scatterers are used in the applications: Perfectly conducting spheres of different diameters and perfectly conducting small scale airplanes.

The sphere class consists of four perfectly conducting spheres with diameters of 7 inches, 10 inches, 15 inches and 20 inches. Within this class, all the targets have obviously the same geometrical features but different sizes. Therefore, classification within such a class may become quite tough especially for those spheres whose diameter values are close to each other.

The time-domain database used for the sphere class simulations is composed of impulse response waveforms at various bistatic angles. The original frequency domain data were generated by a computer program using the Mie series over the frequency band [0.05 GHz - 10 GHz] with 0.05 GHz frequency steps. Therefore, the period of the signal after the inverse fast Fourier transformation (IFFT) is 20 nanoseconds over which the time signal is sampled at 2048 equally spaced points [36].

The model airplane class, on the other hand, consists of five small scale electroplated commercial aircraft models: Boeing 707, Boeing 727, Boeing 747, Concorde and DC-10. The original geometrical details of these aircrafts are preserved in the small scale models. The vertically polarized frequency domain data for all five targets were measured at the Ohio State University compact RCS measurement range over the frequency band [1 GHz - 8 GHz] with 0.05 GHz steps. The impulse response waveforms obtained by taking the IFFT of the windowed fre-

quency domain data are sampled at 4096 points over a period of 20 nanoseconds in time domain [36]. The resulting time domain database is utilized in the airplane class simulation examples.

5.3 Simulation Results for the Sphere Class

In this section, mainly four different classification schemes will be demonstrated for a class of four conducting spheres using the previously explained time domain database. These schemes can be outlined as

- An MLP network using the database directly without any preprocessing or feature extraction stage. (see Simulation Problem 1)
- An MLP network using the selected feature matrix information which is equivalent to time-localized energy matrix of the database obtained by computing the WD of the database. (see Simulation Problem 2)
- The statistical PCA method with a minimum distance (or correlation) measure using the feature matrix information obtained by computing the WD of the database. (see Simulation Problem 3)
- A SOM network using the feature matrix information obtained by computing the WD of the database. (see Simulation Problem 4)

Now, the use of these schemes will be demonstrated by the following simulation examples:

5.3.1 Simulation Problem 1

In this classification scheme, the input layer of the MLP accepts the sample values (decimated to 512 discrete samples) of the impulse response data from the sphere database. The class labels are provided at the output layer of the MLP, which is composed of four neurons corresponding to each sphere in the class. The block diagram of this classification scheme is shown in Figure 5.1.

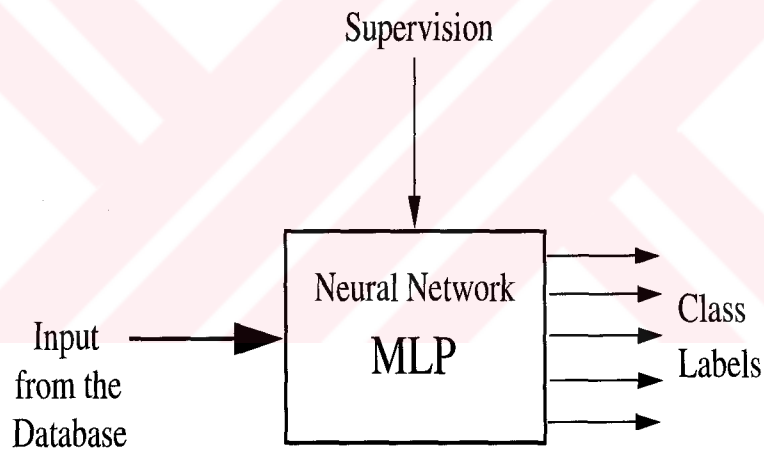


Figure 5.1: Block diagram of the classification scheme discussed in Simulation problem 1

The impulse response data, which is normalized to unit total energy, for four different spheres at 120° aspect angle are shown in Figures 5.2 and 5.3.

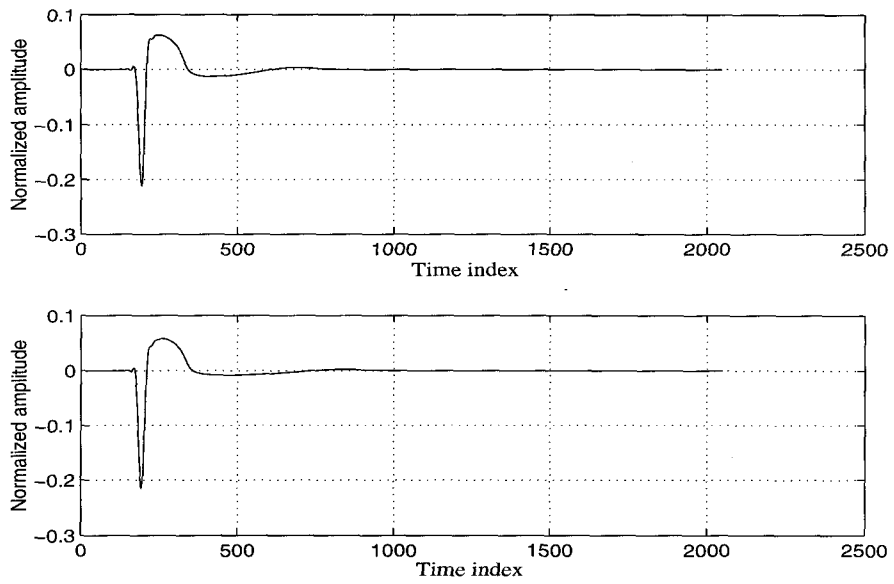


Figure 5.2: Normalized impulse response data for 7 and 10 inch dia. spheres

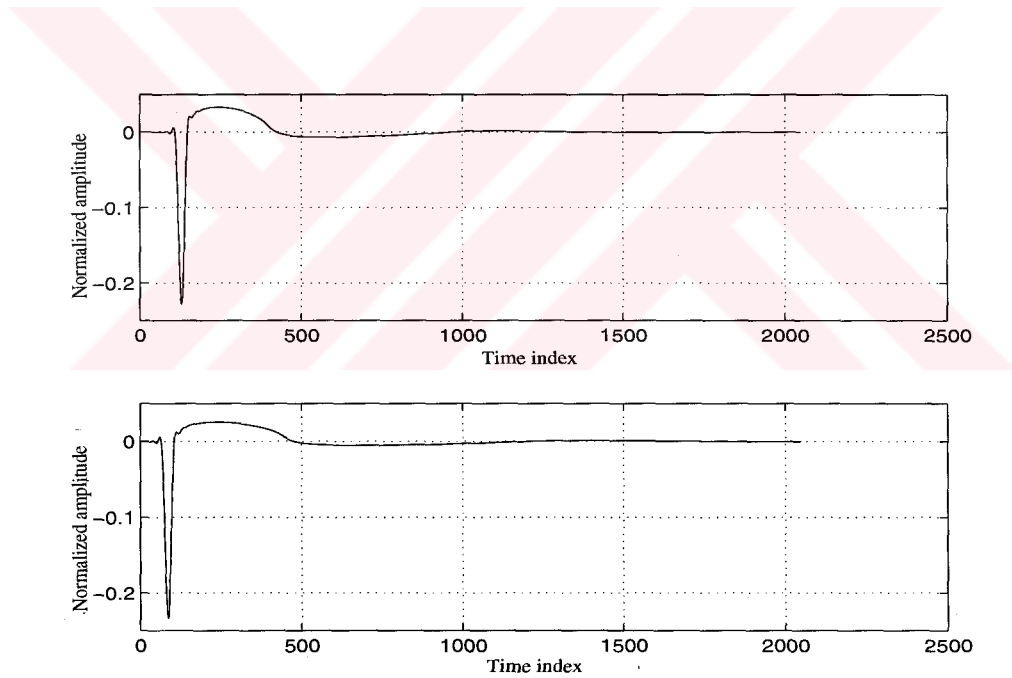


Figure 5.3: Normalized impulse response data for 15 and 20 inch dia. spheres

The important parameters of the MLP trained with the backpropagation algorithm can be listed as follows:

- MLP layer dimensions :

Number of neurons in the input layer = 512

Number of neurons in the first hidden layer = 250

Number of neurons in the second hidden layer = 75

Number of neurons in the output layer = 4

- Training phase parameters:

Learning rate ($\eta(t)$) : $\eta_0 = 0.2$ at $t=0$ (for $t>0$, $\eta(t)$ decreases inversely with time)

Momentum constant (α) = 0.1

Maximum number of repetitions in training = 30000

- Training and test data size :

Number of input data sets used for training = 13 (62%)

{7 inch dia. sphere at 45° , 100° and 150° aspect angles,

10 inch dia. sphere at 0° , 60° , 90° , 135° and 180° aspect angles,

15 inch dia. sphere at 0° and 120° aspect angles,

20 inch dia. sphere at 0° , 120° and 180° aspect angles}

Number of input data sets used for testing = 8 (38%)

{7 inch dia. sphere at 0° and 120° aspect angles,

10 inch dia. sphere at 30°, 45° and 120° aspect angles,
 15 inch dia. sphere at 30° and 150° aspect angles,
 20 inch dia. sphere at 30° aspect angle}

After training the MLP by 62 percent of the available data, the system is tested by using the whole sphere database. As shown in Table 5.1, the overall correct classification rate is 66 percent. On the other hand, the correct classification rate for the completely new test data is found to be very low as only 38 percent.

Data set	Size ratio	Correct decision rate
Test only	38% (8/21)	37% (3/8)
Train only	62% (13/21)	85% (11/13)
Train+Test	100% (21/21)	67% (14/21)

Table 5.1: Classification results of the sphere class by using only MLP classifier

Therefore, these results imply that to achieve acceptable accuracy, further signal processing for feature enhancement must be provided as to be demonstrated in the next simulation problem.

For the classification scheme shown in Figure 5.1, the total cpu processing time spent for a typical test cycle is measured approximately as 120 msec on a Sun Sparc 4 machine.

5.3.2 Simulation Problem 2

The block diagram of this classification scheme is shown in Figure 5.4.

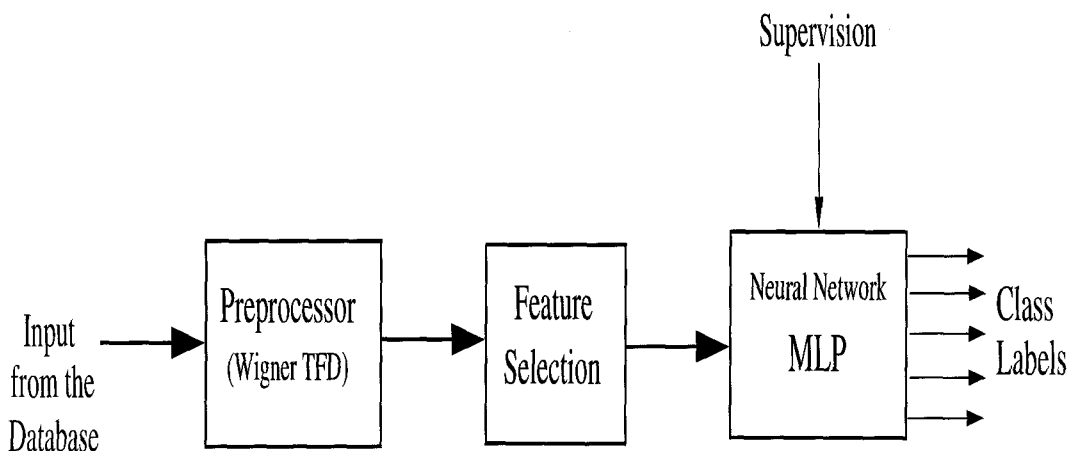


Figure 5.4: Block diagram of the classification scheme discussed in Simulation problem 2

In the first stage the time-frequency localized energy feature matrix is extracted from the impulse response type input data by use of the WD. Then, in the feature selection stage, first a feature vector, suitable to serve the MLP, is formed from the time-frequency energy matrix by placing the consecutive time bands one after another to make a one-dimensional vector. Then, after careful analysis, the dimension of the resulting feature vector can be reduced by considering only some of the time bands which carry more useful information for classification purposes. Finally, the selected feature vectors are applied to the MLP classifier with their class labels.

The WD output matrix for the time-domain signal of sphere class is computed with 2048 sampling points in the time range $[0, 20\text{nsec}]$ and 50 sampling points in the frequency range $[-2.5\text{GHz}, 2.5\text{GHz}]$. A partially time-averaged WD output is obtained by dividing the time span into ten bands and then averaging the WD

values over each band at each sample frequency. Each band corresponds to 204 samples, i.e. to 2 nanoseconds.

The time-frequency localized energy matrices for 7, 10, 15 and 20 inch diameter spheres at 120° aspect angle are shown in Figures 5.5 and 5.6.



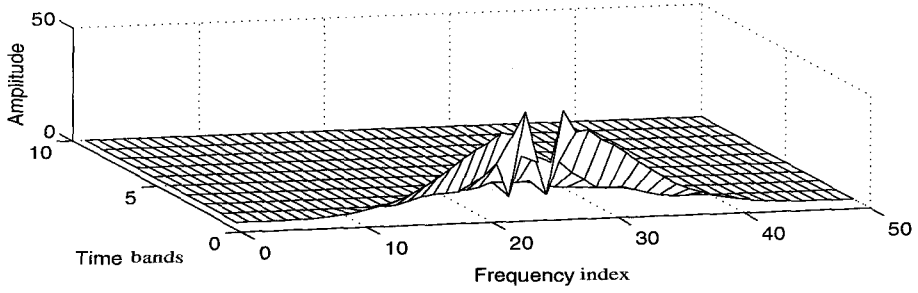
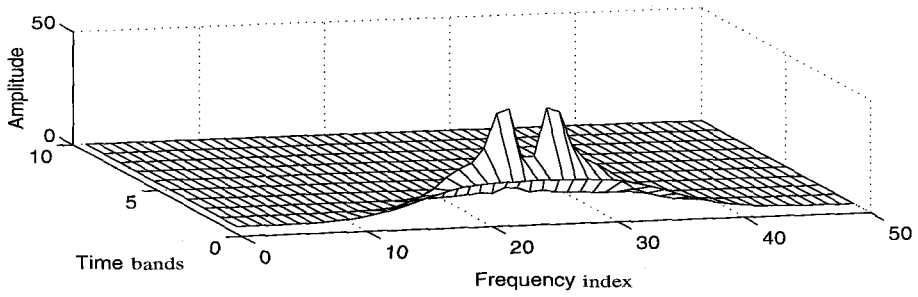


Figure 5.5: The energy feature matrices for 7 and 10 inch dia. spheres

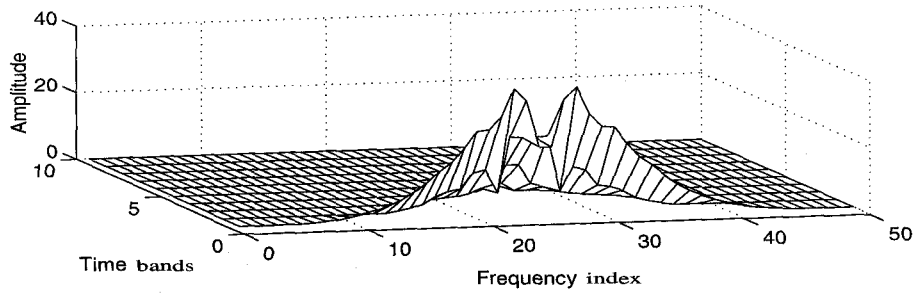
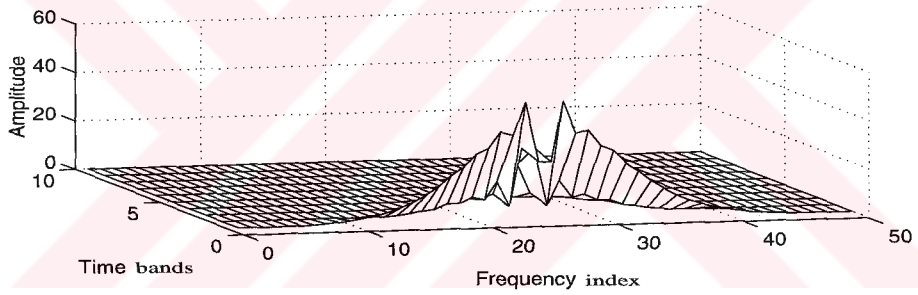


Figure 5.6: The energy feature matrices for 15 and 20 inch dia. spheres

To gain additional insight about this classification system, the selected feature vectors of the 10 inch diameter sphere at aspect angles 30° , 60° , 90° and 120° are presented in Figures 5.7, 5.8, 5.9 and 5.10, respectively. These vectors are formed by selecting the data in the time bands 5, 6 and 7 from the related energy feature matrices as discussed in Section 2.3 previously.

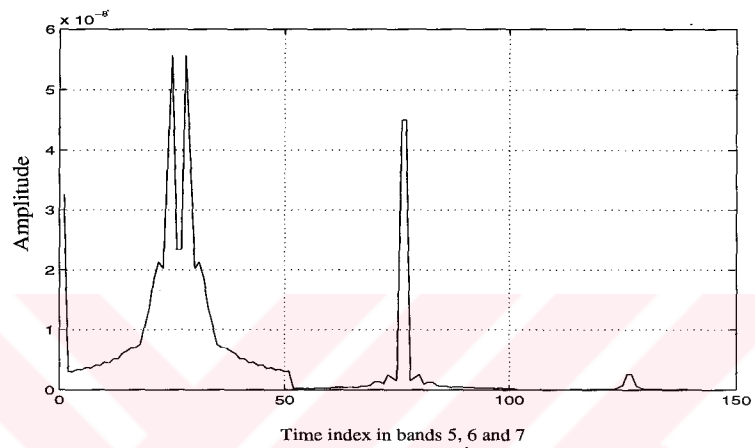


Figure 5.7: The selected feature vector for 10 inch dia. sphere at 30° aspect angle

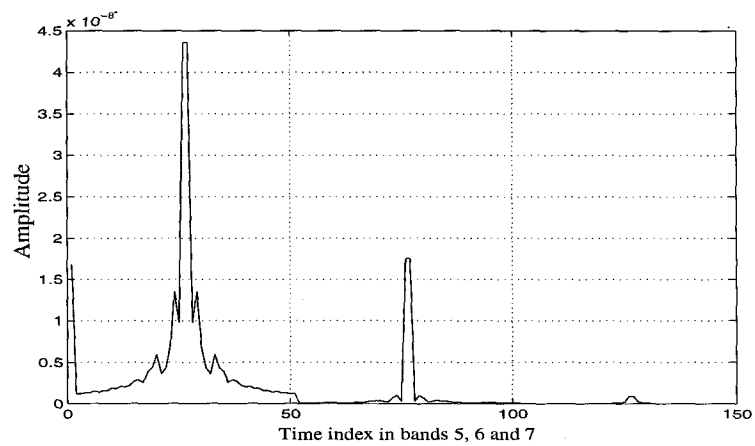


Figure 5.8: The selected feature vector for 10 inch dia. sphere at 60° aspect angle

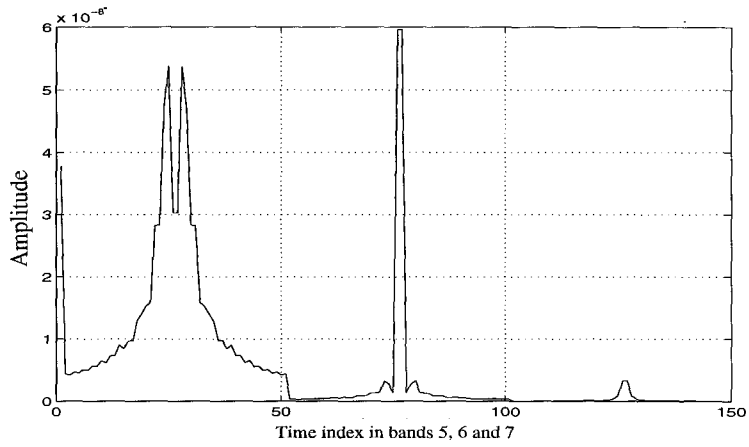


Figure 5.9: The selected feature vector for 10 inch dia. sphere at 90° aspect angle

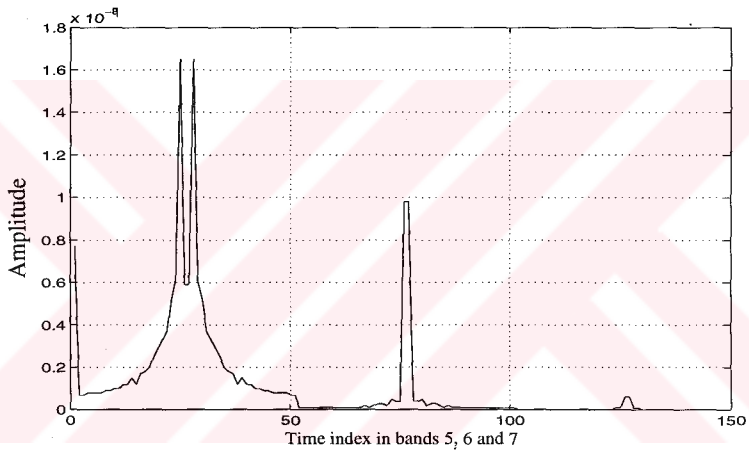


Figure 5.10: The selected feature vector for 10 inch dia. sphere at 120° aspect angle

This feature selection process enables the neural networks with less input neurons to be used and so reduces the overall computational complexity.

The algorithm of the Wigner TFD simulation program, written in C, is given in Appendix A for further applications.

The important parameters of the MLP trained with the backpropagation algorithm can be listed as follows:

- MLP layer dimensions :

Number of neurons in the input layer = 150

Number of neurons in the first hidden layer = 75

Number of neurons in the second hidden layer = 25

Number of neurons in the output layer = 2 (binary output)

- Training phase parameters:

Learning rate ($\eta(t)$) : $\eta_0 = 0.3$ at $t=0$ (for $t > 0$, $\eta(t)$ decreases inversely with time)

Momentum constant (α) = 0.1

Maximum number of repetitions in training = 10000

- Training and test data size :

Number of input data sets used for training = 13 (62%)

{7 inch dia. sphere at 45° , 100° and 150° aspect angles,

10 inch dia. sphere at 0° , 60° , 90° , 135° and 180° aspect angles,

15 inch dia. sphere at 0° and 120° aspect angles,

20 inch dia. sphere at 0° , 120° and 180° aspect angles}

Number of input data sets used for testing = 8 (38%)

{7 inch dia. sphere at 0° and 120° aspect angles,

10 inch dia. sphere at 30° , 45° and 120° aspect angles,

15 inch dia. sphere at 30° and 150° aspect angles,

20 inch dia. sphere at 30° aspect angle}

A sample run output of the MLP test simulation program is given below :

```
>> mlptst  
  
>> Input the filename to load for testing > os1060mn.mex  
  
>> Input first hidden layer weight filename > wh1last.mex  
  
>> Input second hidden layer weight filename > wh2last.mex  
  
>> Input output weight filename > woplast.mex  
  
>> Net.class→outs[1] = 0.216113 Net.class→outs[2] = 0.829882  
  
>> *** Classification Result → Target is 10 inch dia. sphere
```

The algorithms of the MLP simulation programs for training and testing, written in C, are given in Appendix A for further applications.

The instantaneous squared error per output during the training phase of MLP is shown in Figure 5.11. Note that the error drops to negligible levels after about 10000 iterations in the training phase.

After the MLP is trained by 62 percent of the available data, the system is tested by using the whole sphere database. As shown in Table 5.2, the overall correct classification rate is 86 percent. On the other hand, the correct classification rate for the completely new test data is found to be 63 percent, which is much better than as in Simulation problem 1.

It is confirmed by comparing the simulation results in Table 5.1 and Table 5.2 that the preprocessing stage in the classification systems, which benefits from time-frequency analysis for feature extraction, plays a very important role to increase the correct classification rate.

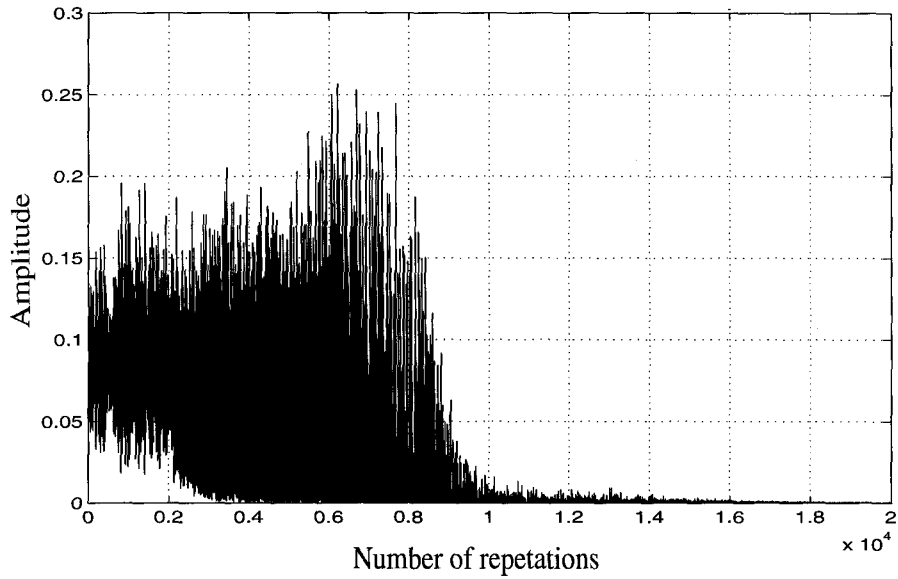


Figure 5.11: The instantaneous squared error per output of the MLP classifier

Data set	Size ratio	Correct decision rate
Test only	38% (8/21)	62% (5/8)
Train only	62% (13/21)	100% (13/13)
Train+Test	100% (21/21)	86% (18/21)

Table 5.2: Classification results of the sphere class by using the WD+MLP classifier

For the classification scheme shown in Figure 5.4, the total cpu processing time for the testing phase is measured approximately as 270 msec (190 msec for feature extraction stage and 80 msec for MLP classification stage) on a Sun Sparc 4 machine.

5.3.3 Simulation Problem 3

In this classification scheme, the popular statistical method PCA with its simple postprocessing classification measures based on minimum distance or cross-correlation techniques is applied to a time-frequency localized energy matrix extracted by use of the WD.

The block diagram of this classification scheme is shown in Figure 5.12

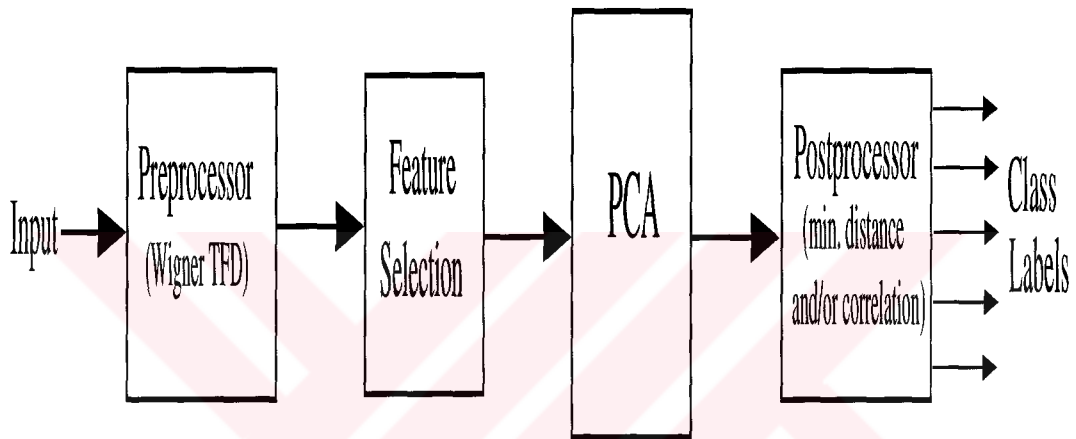


Figure 5.12: Block diagram of the classification scheme discussed in Simulation problem 3

The PCA outputs (eigenvectors) corresponding to the major eigenvalue for 7 inch diameter sphere data at 0° , 45° , 120° and 150° aspect angles are shown in Figures 5.13 and 5.14.

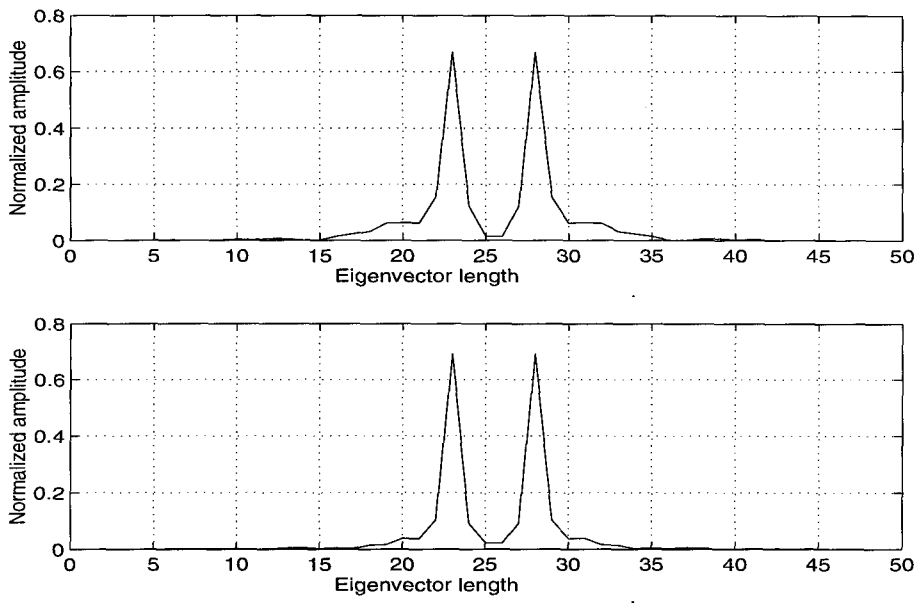


Figure 5.13: PCA outputs for 7 inch dia. sphere at 0° and 45° aspect angles

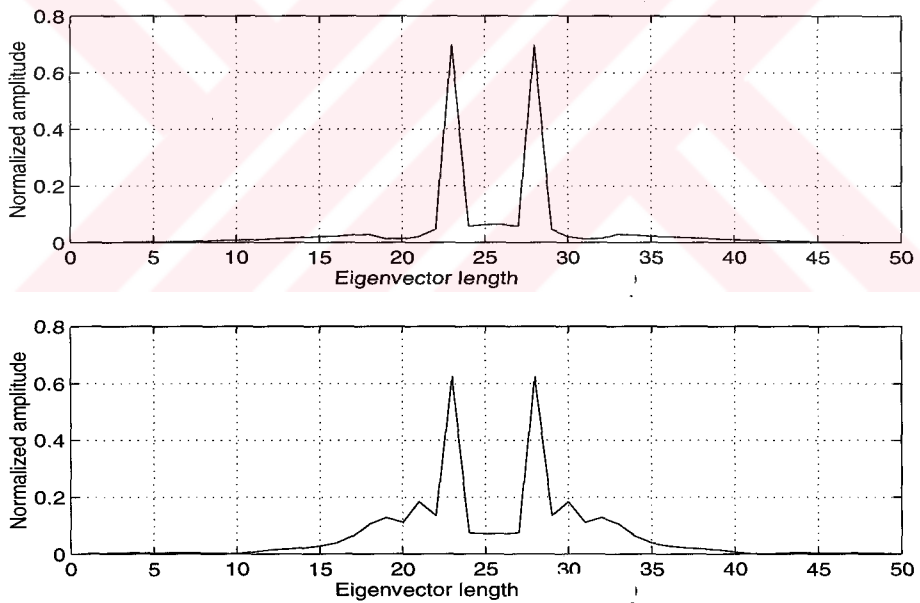


Figure 5.14: PCA outputs for 7 inch dia. sphere at 120° and 150° aspect angles

After finding the PCA outputs of the input data, classification is achieved by use of minimum Euclidian distance measure or cross-correlation measure. A sample run output of the classification program is given below :

```
>> euclid  
  
>> Input first data filename to load > f7bcks.mex  
  
>> Input second data filename to load > f745s.mex  
  
>> The euclidian distance measure between test inputs > 0.110054  
  
>> The cross-correlation measure between test inputs > 0.993944
```

In this approach, for a given test input vector and a database consisting of the known vectors, it is required to test the input vector with each vector in the database by using the test program and finally determine to which member of the database this vector is closest. From the simulation results, the correct classification rate for the sphere class, where approximately half of the total vectors is used for testing, is found to be nearly 90%.

For the classification scheme shown in Figure 5.12, the total cpu processing time of this classification scheme is measured approximately as 550 msec (190 msec for feature extraction stage and 360 msec for PCA and minimum distance classification stage) on a Sun Sparc 4 machine.

5.3.4 Simulation Problem 4

The block diagram of this classification scheme is shown in Figure 5.15.

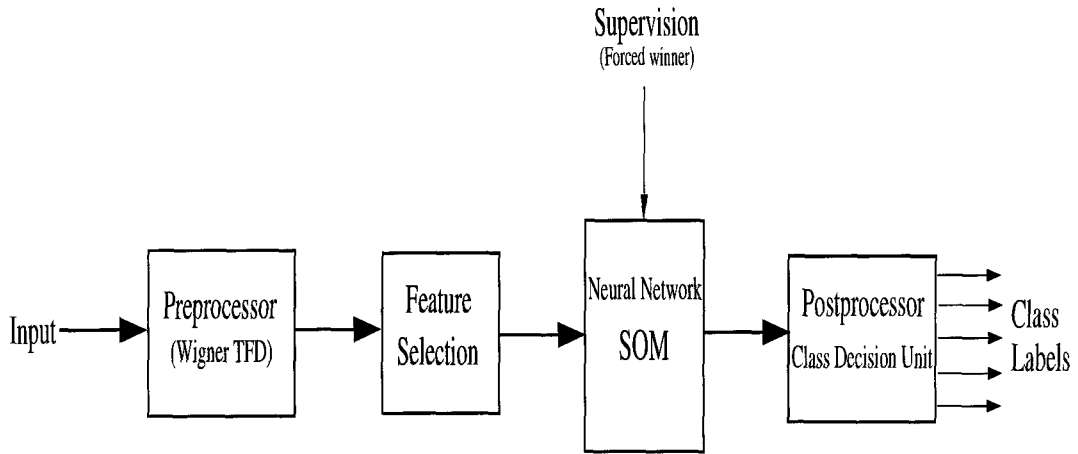


Figure 5.15: Block diagram of the classification scheme discussed in Simulation problem 4

The selected feature vectors are applied to the input of SOM network, which is trained with the forced winner algorithm discussed in Chapter 3. During the testing phase, after sufficient number of training and test inputs, the decision boundaries of the forced clusters for each class can be formed in order to automatically classify each input. This is the function of the specified decision unit shown in the block diagram.

The specifications of the SOM trained by using the forced winner algorithm can be listed as follows:

- SOM input dimensions :

pattern length = 50

number of patterns = 3

- SOM output dimensions :

Number of neurons in a row of output map = 10

Number of neurons in a column of output map = 10

- The first phase of training :

Learning rate ($\eta(t)$) : $\eta_0 = 0.1$ at $t=0$ (for $t > 0$, $\eta(t)$ decreases inversely with time)

Neighborhood function: bubble type, initial radius = 10, neigh. decreasing rate = 10^{-1} per weight adaptation

Maximum number of repetitions in training = 1000

- The second phase of training :

Learning rate ($\eta(t)$) : $\eta_0 = 0.03$ at $t=0$ (for $t > 0$, $\eta(t)$ decreases inversely with time)

Neighborhood function: bubble type, initial radius = 2, neigh. decreasing rate = 10^{-1} per weight adaptation

Maximum number of repetitions in training = 10000

- Training and test data size:

Number of input data sets used for training = 11 (52%)

{7 inch dia. sphere at 45° , 100° and 150° aspect angles,

10 inch dia. sphere at 0° , 60° , 135° and 180° aspect angles,

15 inch dia. sphere at 0° and 150° aspect angles,

20 inch dia. sphere at 0° and 180° aspect angles}

Number of input data sets used for testing = 10 (48%)

{7 inch dia. sphere at 0° and 120° aspect angles,

10 inch dia. sphere at 30°, 45°, 90° and 120° aspect angles,

15 inch dia. sphere at 30° and 120° aspect angles,

20 inch dia. sphere at 30° and 120° aspect angles}

A sample run output of the SOM test simulation program is given below:

```
>> somtest
```

```
>> Input the feature matrix file name to load for testing > os10180mnx.mex
```

```
>> Input the SOM weight file name > somwsph3.mex
```

```
>> Winner-row-col → 44-5-4
```

```
>> *** Classification Result → Target is 10 inch dia. sphere
```

The algorithms of the SOM simulation programs for training and testing phases, which are written in C, are given in Appendix A for further applications.

After the SOM is trained by 52 percent of the available data, the system is tested by using the whole sphere database. The final result of simulations, including the training and testing phases, for the sphere class by using the classification scheme in Figure 5.15 is shown in Figure 5.16. As seen from the figure, the overall correct classification rate is 100 percent, that is all test inputs for the sphere class are correctly classified.

From Figure 5.16, we see that it is possible to automatically differentiate four classes of sphere data by determining cluster regions indicated by the dashed lines after sufficient number of training and test data are used. Also note that in case of new unknown test data, which is far enough from all known clusters, a new cluster region corresponding to a new class can be constructed on the output map.

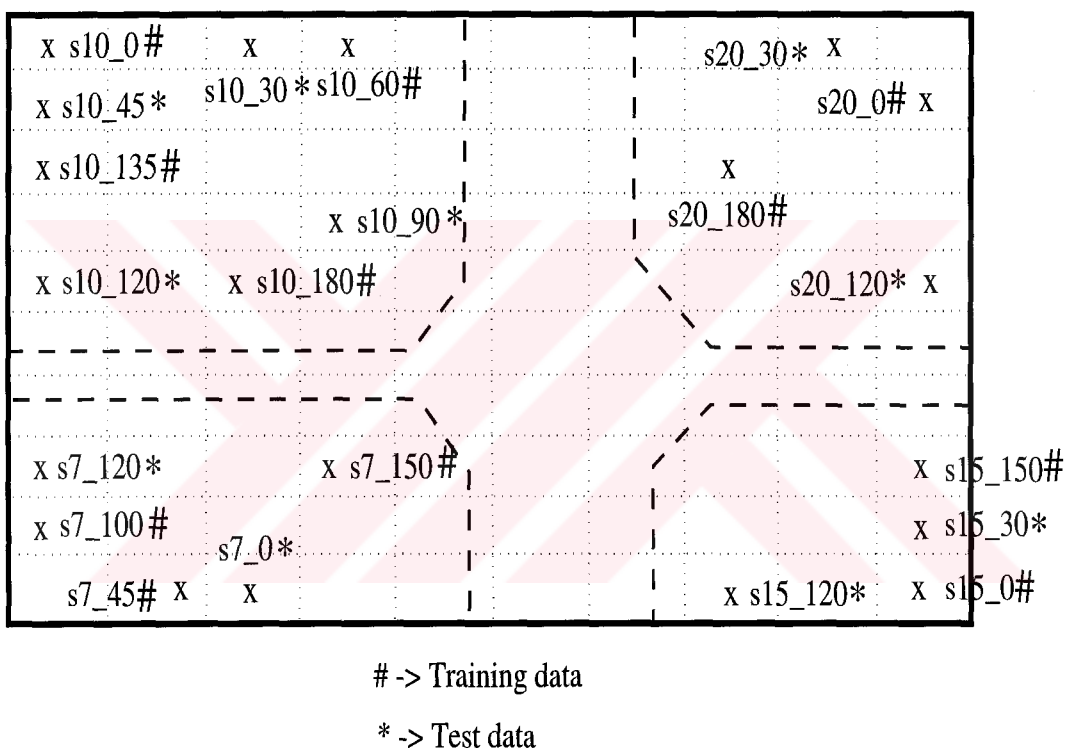


Figure 5.16: The final SOM output (10x10) of sphere class

For the classification scheme shown in Figure 5.15, the total cpu processing time of this classification scheme is measured approximately as 230 msec, where 190 msec for feature extraction stage and 40 msec for SOM classification stage, on a Sun Sparc 4 machine.

5.3.5 Simulation Problem 5

After the class label of an input data is determined by the system as explained in previous section, it is also feasible to estimate its aspect angle by using the classification scheme shown in Figure 5.17.

In this system, the aspect estimation is accomplished by training an aspect finder SOM for each sub-class using the multi-aspect data of the related target only.

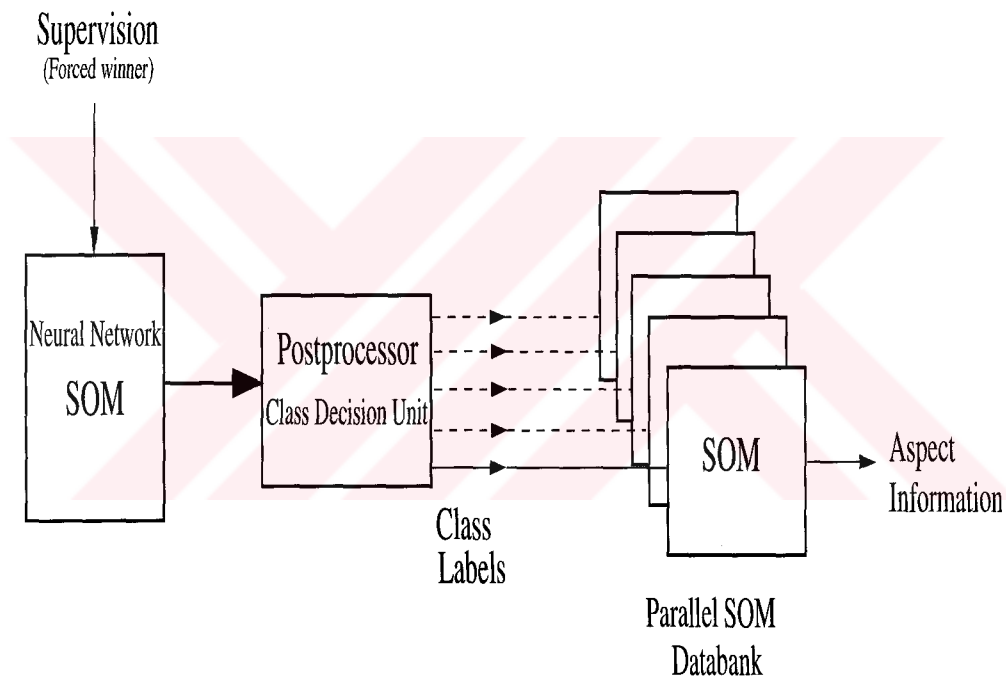


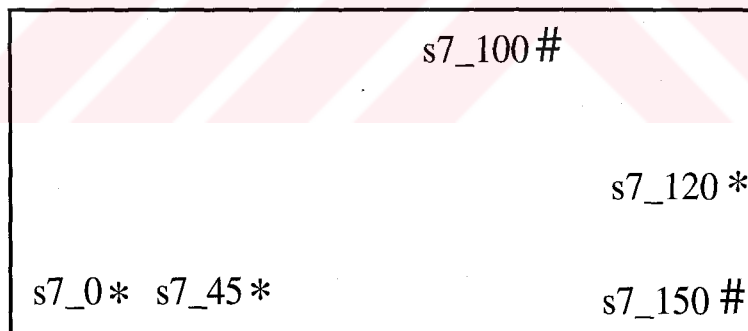
Figure 5.17: Block diagram of the classification scheme discussed in Simulation problem 5

A secondary SOM classifier is trained by the forced winner algorithm for each target type within the class to form subclusters with respect to aspect angle information. For instance, if the main SOM identifies the test target as the target "# j", then the j'th secondary SOM decides about the aspect angle range

for the test data. This secondary SOM must be trained a priori by the feature enhanced data which belong to the target "# j" at various aspect angles such as 0°, 45°, 90°, 135° and 180°.

The simulation results for 7 and 10 inch diameter spheres in this respect are shown in Figures 5.18 and 5.19.

We conclude, based upon all of the simulation results presented so far for the sphere class, that the WD+SOM scheme shown in Figure 5.15 is the most promising one considering the overall correct classification rates and the total cpu processing times. In addition, the statistical PCA classification scheme of Figure 5.12 has turned out to be performing better than the most widely used MLP classifier of Figure 5.4.



-> Training data

* -> Test data

Figure 5.18: The final SOM output (4x5) for aspect estimation of 7 inch dia. sphere data

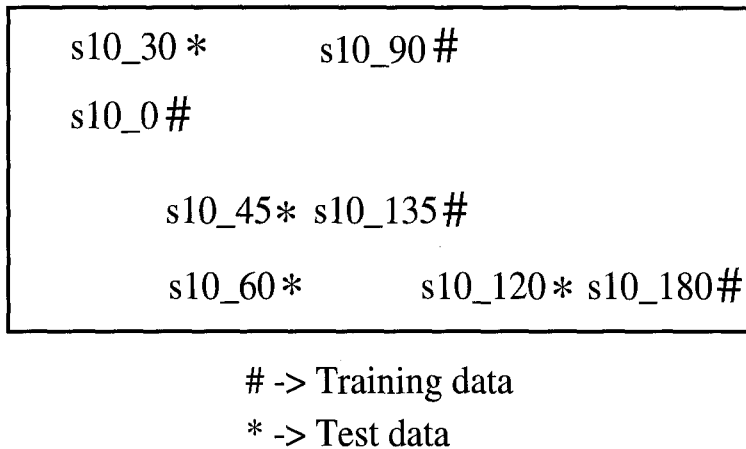


Figure 5.19: The final SOM output (4x5) for aspect estimation of 10 inch dia. sphere data

5.4 Simulation Results for the Aircraft Class

After presenting the target classification simulations for the sphere class, we will move onto a much more realistic class of model aircraft. The simulation results to be presented in this section are especially important to give an idea about the feasibility of the proposed classification schemes in real world radar system applications. The classifier schemes to be used in the simulations of this section have already been introduced in the sphere class applications. Therefore, the details related to the schemes will not be repeated here again.

5.4.1 Simulation Problem 6

In this application, the database for the aircraft class, including the models B-707, B-727, B-747, Concorde and DC-10, is applied to the input of the classification scheme shown in Figure 5.4, which is basically a WD+MLP system.

As an example, the time-domain impulse response data, which is normalized to have unit total energy, for all models of planes at 45° aspect angle are shown in Figures 5.20, 5.21, 5.22, 5.23 and 5.24.

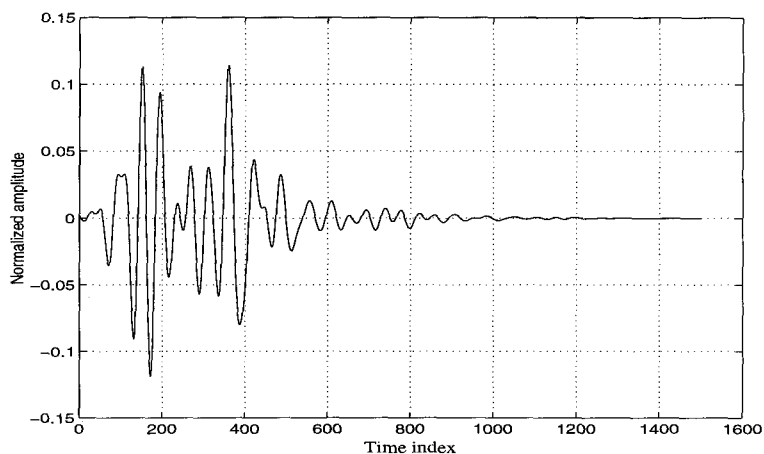


Figure 5.20: Normalized impulse response data for plane B-707

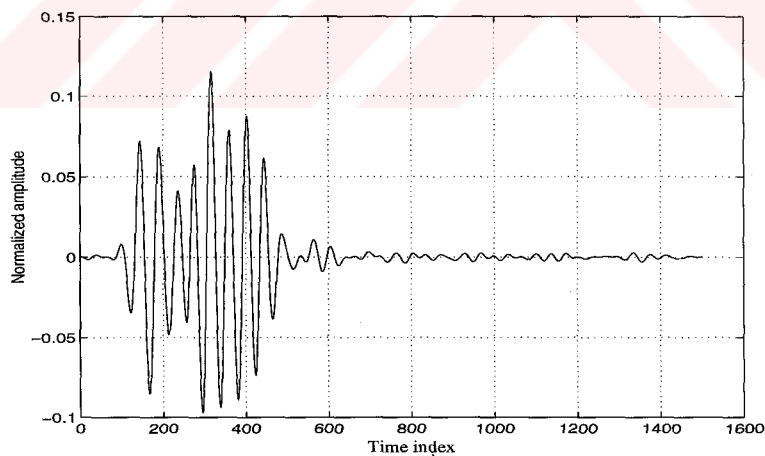


Figure 5.21: Normalized impulse response data for plane B-727

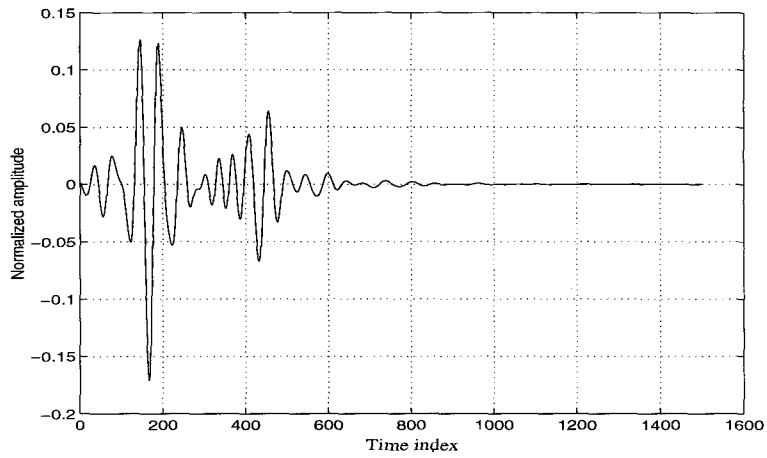


Figure 5.22: Normalized impulse response data for plane B-747

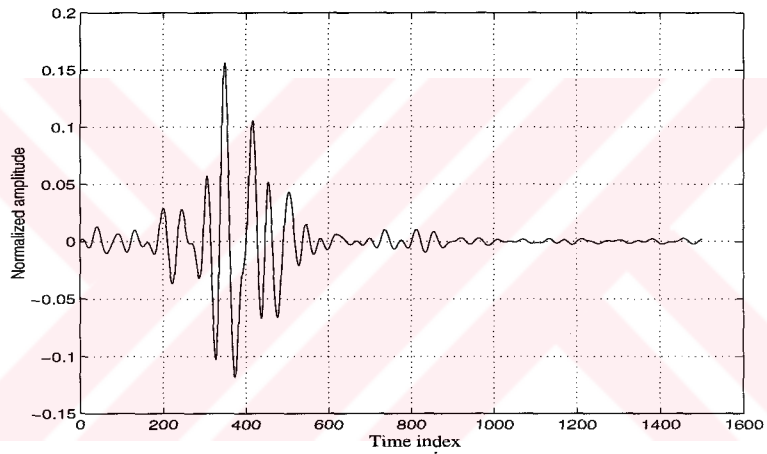


Figure 5.23: Normalized impulse response data for plane Concorde

The WD output matrix for the time-domain signal of aircraft class is computed with 1500 sampling points in the time range $[0, 7.32\text{nsec}]$ and 50 sampling points in the frequency range $[-4\text{ GHz}, 4\text{ GHz}]$. Similarly, a partially time-averaged WD output is obtained by dividing the time span into ten bands and then averaging the WD values over each band at each sample frequency. Each band corresponds to 150 samples, i.e. to 0.73 nanoseconds.

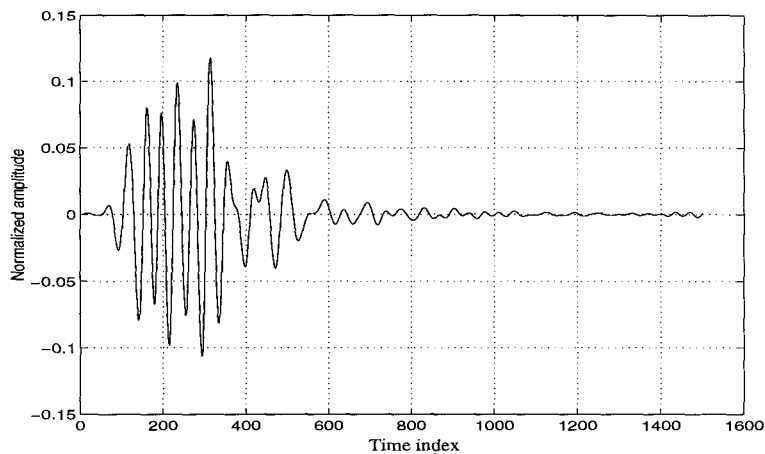


Figure 5.24: Normalized impulse response data for plane DC-10

Following the similar steps discussed in simulation problem 2, the selected feature vectors over the time bands 6, 7 and 8 for each model aircraft at 45° aspect angle are shown in Figures 5.25, 5.26, 5.27, 5.28 and 5.29 as an example.

The important parameters of the MLP trained with the backpropagation algorithm can be listed as follows:

- MLP layer dimensions :

Number of neurons in the input layer = 150

Number of neurons in the first hidden layer = 100

Number of neurons in the second hidden layer = 30

Number of neurons in the output layer = 5

- Training phase parameters:

Learning rate ($\eta(t)$) : $\eta_0 = 0.35$ at $t=0$ (for $t > 0$, $\eta(t)$ decreases inversely with time)

Momentum constant (α) = 0.1

Maximum number of repetitions in training = 20000

- Training and test data size :

Number of input data sets used for training = 17 (55%)

{B-707 model plane at 45°, 120° and 180° aspect angles,

B-727 model plane at 30°, 60°, 150° and 180° aspect angles,

B-747 model plane at 0° and 90° aspect angles,

Concorde model plane at 0°, 30°, 90° and 180° aspect angles

DC-10 model plane at 0°, 30°, 90° and 180° aspect angles}

Number of input data sets used for testing = 14 (45%)

{B-707 model plane at 60°, 90° and 150° aspect angles,

B-727 model plane at 45°, 90°, and 120° aspect angles,

B-747 model plane at 45° aspect angle,

Concorde model plane at 45°, 60°, and 120° aspect angles

DC-10 model plane at 45°, 60°, 120° and 150° aspect angles}

The given parameters of the MLP are determined after making some simulations until which the better classification performance is achieved.

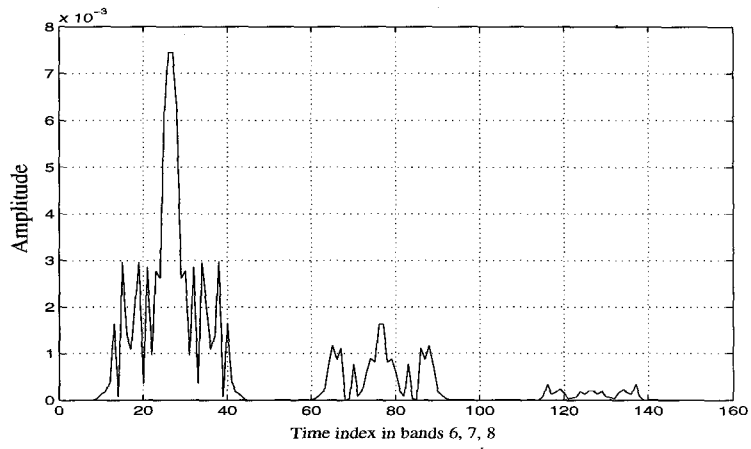


Figure 5.25: The selected feature vector for plane B-707 at 45° aspect angle

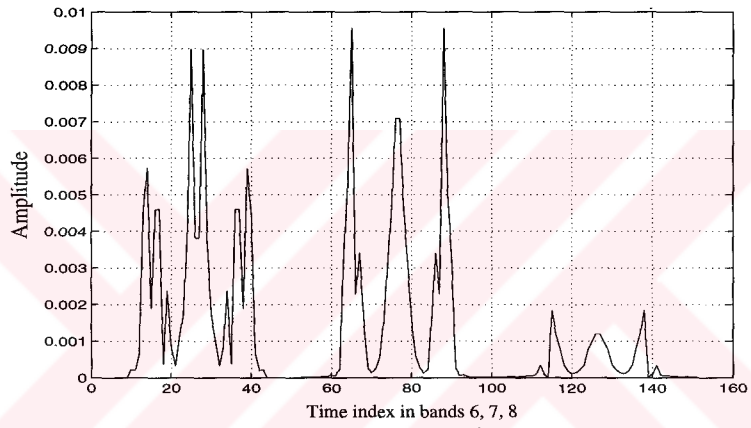


Figure 5.26: The selected feature data for plane B-727 at 45° aspect angle

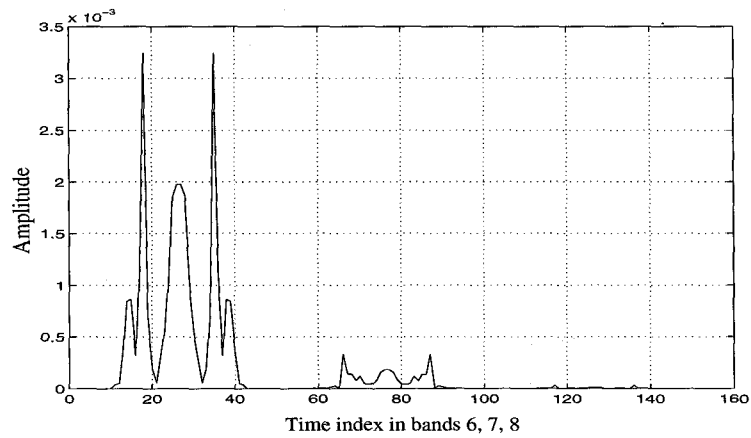


Figure 5.27: The selected feature data for plane B-747 at 45° aspect angle

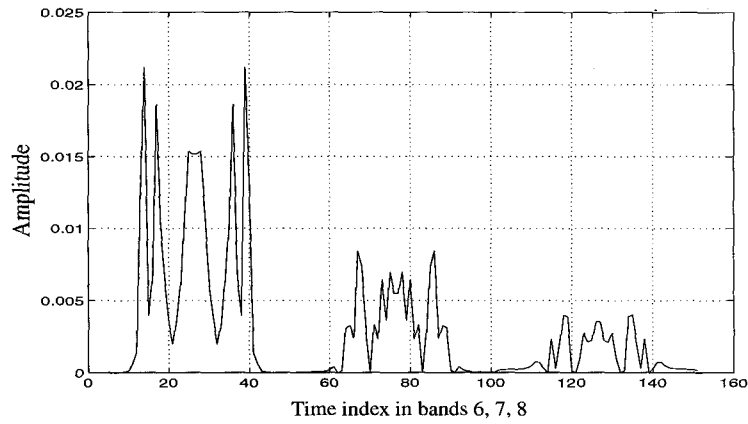


Figure 5.28: The selected feature data for plane Concorde at 45° aspect angle

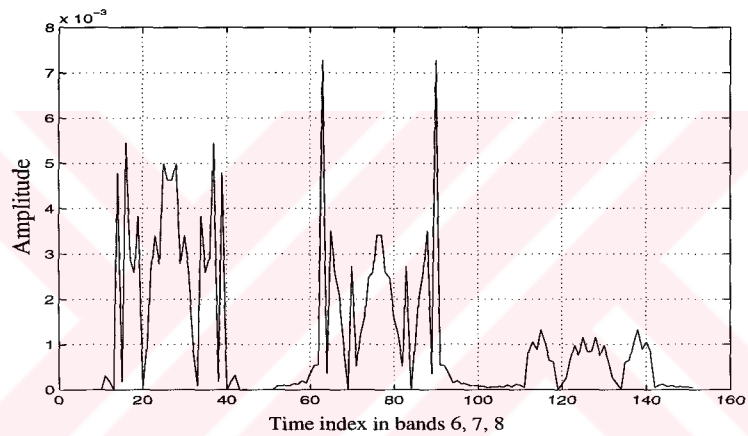


Figure 5.29: The selected feature data for plane DC-10 at 45° aspect angle

After the MLP is trained by 55 percent of the available data, the system is tested by using the whole aircraft database. As shown in Table 5.3, the overall correct classification rate is 84 percent. On the other hand, the correct classification rate for the completely new test data is found to be 64 percent.

By comparing the results in Table 5.2 and Table 5.3, we see that the performance of the WD+MLP classifier is nearly same for the sphere class and the aircraft class. Also the total cpu processing time for the testing phase of aircraft

Data set	Size ratio	Correct decision rate
Test only	45% (14/31)	64% (9/14)
Train only	55% (17/31)	100% (17/17)
Train+Test	100% (31/31)	84% (26/31)

Table 5.3: Classification results of the aircraft class by using the WD+MLP classifier

classification is measured approximately as 270 msec on a Sun Sparc 4 machine.

5.4.2 Simulation Problem 7

In this case, the extracted time-frequency localized energy matrix of an aircraft impulse response data is applied to the statistical classifier using PCA with minimum distance measure (see Figure 5.12).

The PCA outputs corresponding to the major eigenvalue for each model aircraft at 45° aspect angle are shown in Figures 5.30, 5.31, 5.32, 5.33 and 5.34.

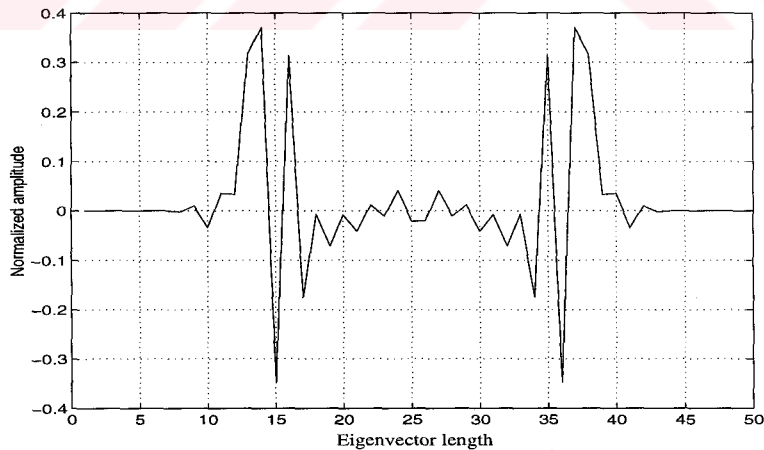


Figure 5.30: PCA output for plane B-707 at 45° aspect angle

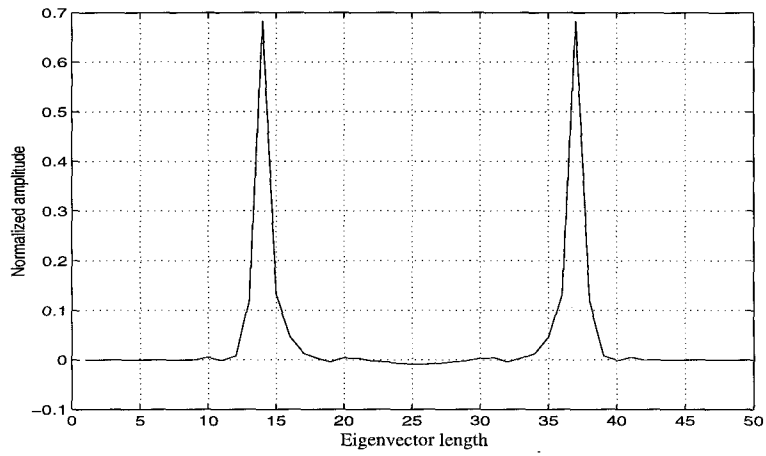


Figure 5.31: PCA output for plane B-727 at 45° aspect angle

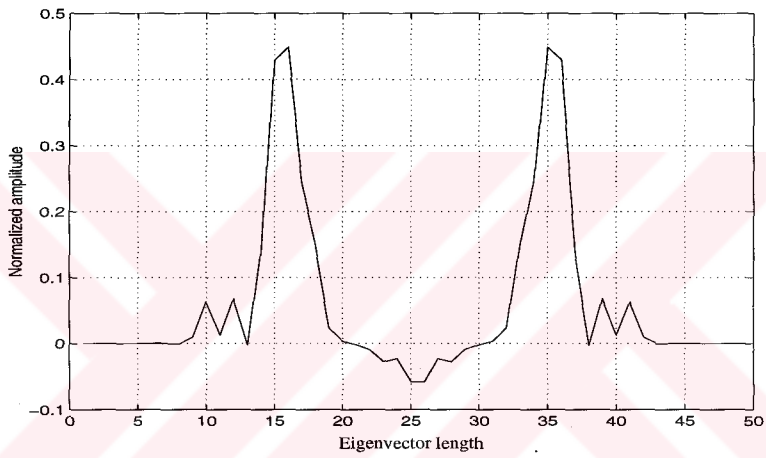


Figure 5.32: PCA output for plane B-747 at 45° aspect angle

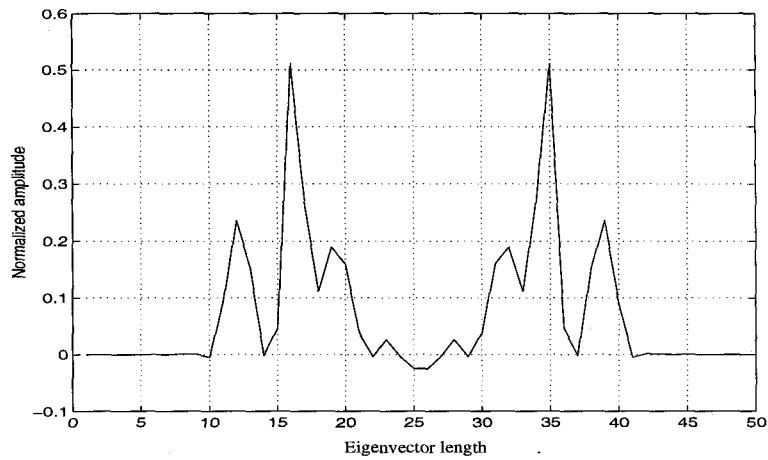


Figure 5.33: PCA output for plane Concorde at 45° aspect angle

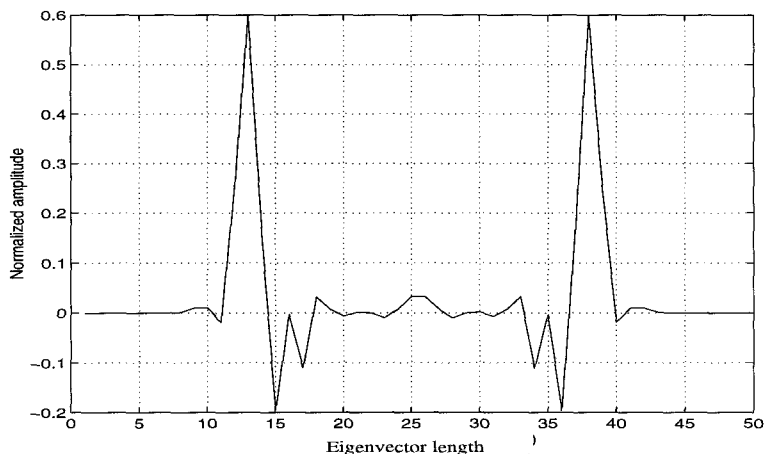


Figure 5.34: PCA output for plane DC-10 at 45° aspect angle

After obtaining the PCA outputs for the whole available aircraft database, the classification is achieved by use of the minimum Euclidian distance measure or cross-correlation measure as previously done for the sphere class.

From the simulation results, where 45 percent of all available plane data is used for testing, the successful classification rate is found nearly 80%. In this case the total cpu processing time of this classification scheme is measured approximately as 620 msec (190 msec for feature extraction stage and 430 msec for PCA and minimum distance classification stage) on a Sun Sparc 4 machine.

5.4.3 Simulation Problem 8

In this problem, the selected feature vectors which are obtained as discussed previously in Simulation problem 4 for the sphere class, are the inputs of the SOM network trained with the forced winner algorithm. The block diagram of the classification scheme used in this problem is shown in Figure 5.15.

In this case, the specifications of the SOM trained by using the forced winner algorithm can be listed as follows:

- SOM input dimensions :

pattern length = 50

number of patterns = 3

- SOM output dimensions :

Number of neurons in a row of output map = 15

Number of neurons in a column of output map = 15

- The first phase of training :

Learning rate ($\eta(t)$) : $\eta_0 = 0.2$ at $t=0$ (for $t > 0$, $\eta(t)$ decreases inversely with time)

Neighborhood function: bubble type, initial radius = 15, neigh. decreasing rate = 10^{-1} per weight adaptation

Maximum number of repetitions in training = 3000

- The second phase of training :

Learning rate ($\eta(t)$) : $\eta_0 = 0.05$ at $t=0$ (for $t > 0$, $\eta(t)$ decreases inversely with time)

Neighborhood function: bubble type, initial radius = 3, neigh. decreasing rate = 10^{-1} per weight adaptation

Maximum number of repetitions in training = 30000

- Training and test data size:

Number of input data sets used for training = 17 (55%)

{B-707 model plane at 45°, 120° and 180° aspect angles,

B-727 model plane at 30°, 60°, 150° and 180° aspect angles,

B-747 model plane at 0° and 90° aspect angles,

Concorde model plane at 0°, 30°, 90° and 180° aspect angles

DC-10 model plane at 0°, 30°, 90° and 180° aspect angles}

Number of input data sets used for testing = 14 (45%)

{B-707 model plane at 60°, 90° and 150° aspect angles,

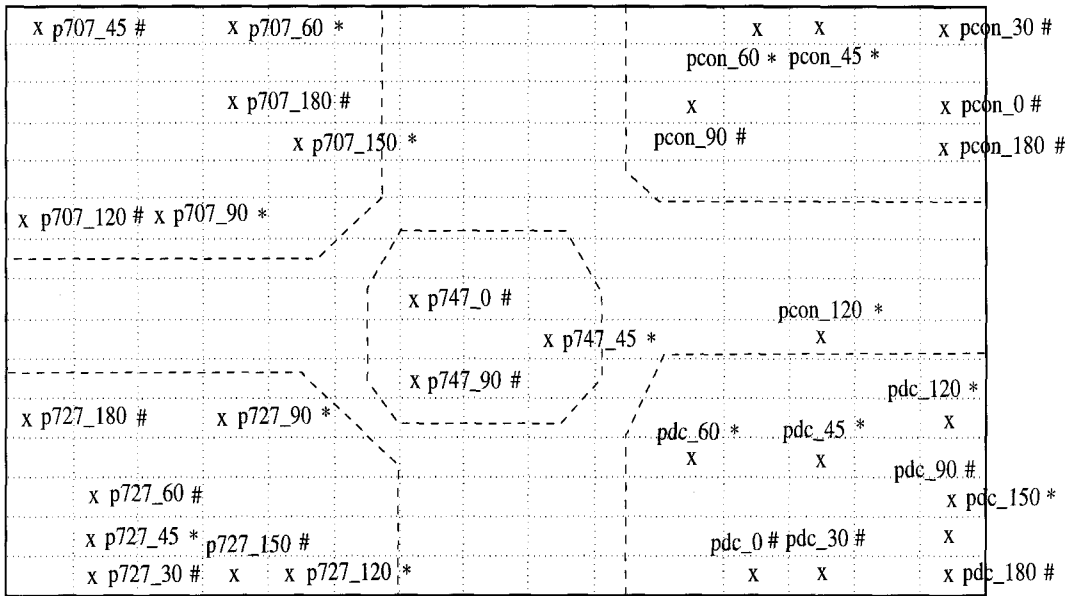
B-727 model plane at 45°, 90°, and 120° aspect angles,

B-747 model plane at 45° aspect angle,

Concorde model plane at 45°, 60°, and 120° aspect angles

DC-10 model plane at 45°, 60°, 120° and 150° aspect angles}

After the SOM is trained by 55 percent of the available data, the system is tested by using the whole aircraft database. The final result of simulations, including the training and testing phases, for the aircraft class by using the classification scheme in Figure 5.15 is shown in Figure 5.35. As seen from the figure, the overall correct classification rate is nearly 97 percent as there is only one case of ambiguous decision (for Concorde data at 120°) out of the total of 31 cases.



-> Training data

* -> Test data

(15x15 SOM Output Map)

Figure 5.35: The final SOM output (15x15) of the aircraft class

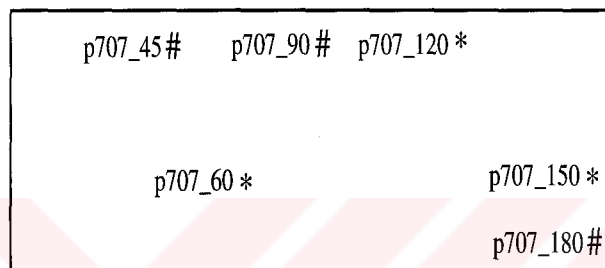
From Figure 5.35, we see that it is again possible to automatically differentiate five classes of aircraft data by determining cluster regions indicated by the dashed lines after sufficient number of training and test data are used. Also note that in case of new unknown test data, which is far enough from all known clusters, a new cluster region corresponding to a new class can be constructed on the output map.

The total cpu processing time of this classification scheme for the aircraft class data inputs is measured approximately as 250 msec, where 190 msec for feature extraction stage and 60 msec for SOM classification stage, on a Sun Sparc 4 machine.

5.4.4 Simulation Problem 9

By using similar discussions made for Simulation problem 5, after the class label of an input data is determined (as in Simulation problem 8), it is also feasible to estimate its aspect angle by using the classification scheme shown in Figure 5.17.

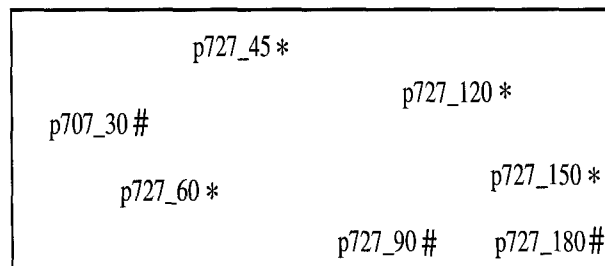
The simulation results in this respect for model B-707, B-727 and DC-10 planes are given in Figures 5.36, 5.37 and 5.38.



-> Training data

* -> Test data

Figure 5.36: The final SOM output (4x5) for aspect estimation of B-707 plane data



-> Training data

* -> Test data

Figure 5.37: The final SOM output (4x5) for aspect estimation of B-727 plane data

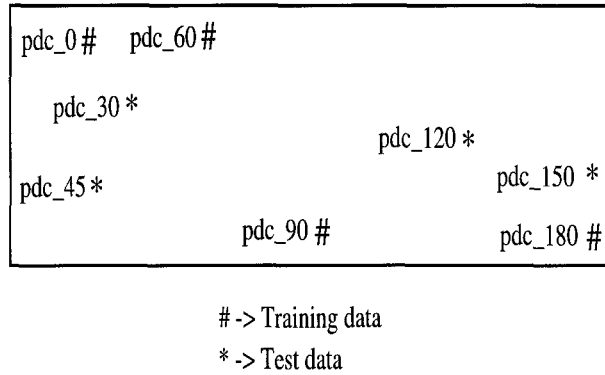


Figure 5.38: The final SOM output (4x5) for aspect estimation of DC-10 plane data

Based upon the simulation results for the aircraft class the WD+SOM scheme of Figure 5.15 is again the most promising one considering the overall performances of the alternative classifiers. As compared to the sphere class results, this time for the same comparison criteria, the MLP classifier of Figure 5.4 is found to be performing better than the statistical PCA classification scheme of Figure 5.12.

5.5 Noise Performance of the Proposed Classification System

One of the most important criteria to judge a target classifier is its noise performance. We have examined the noise behaviour only for that classification scheme which uses WD and SOM combination as shown in Figure 5.15, since this scheme has achieved the best classification performance otherwise, among the other alternatives.

As stated earlier, the database used for the sphere class simulations is composed of synthetic data. The database for the aircraft class, on the other hand, was measured on a compact RCS measurement range which can be assumed as an almost noise-free environment. Then, the noisy data required to test the noise-handling capacity of the proposed target classifier can be obtained by adding specific amounts of uncorrelated random Gaussian noise to the real and imaginary parts of the original frequency domain back-scattered data [36]. The details of the noise addition process are given in Appendix B.

Note that the effect of SNR (signal-to-noise ratio) of the input signal is examined only for the testing phase in this section. Effects of training the SOM network with noisy data could have been examined as well for a more detailed noise analysis.

For demonstration purposes, the back-scattered impulse response of the 10 inches diameter sphere at 0° bistatic angle is contaminated with noise at the SNR levels of 15 dB, 10 dB and 5 dB. Obviously, the original noise-free data corresponds to SNR of infinity. The shift in the location of the winning neuron (in the testing phase) for these data with respect to the level of added noise is presented in Figure 5.39. We observe that for SNR = 15dB case the location of related winning neuron deviates 1 unit below, for SNR = 10dB case its location deviates 2 units below, and finally for SNR = 5 dB case (corresponds to limiting excessive noise case) its location deviates 3 units right from the winning neuron location of the noise-free case. So, for the noisy data at these SNR levels the resulting winning neurons still stay in the cluster region for the 10 inch diameter sphere making correct decision possible.

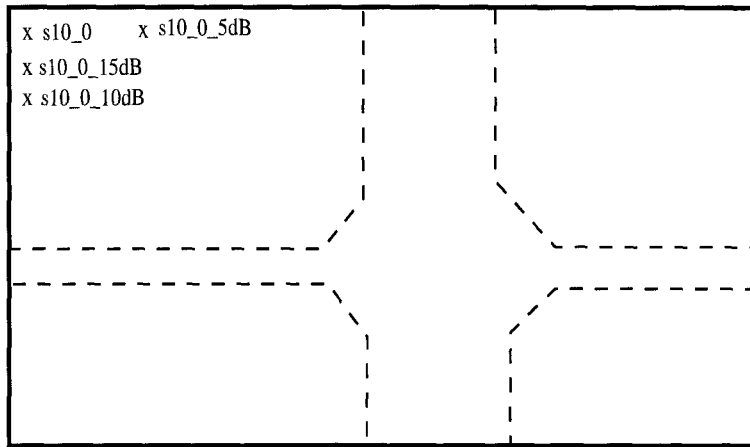


Figure 5.39: Noise analysis for 10 inch sphere at 0° aspect angle with 15dB, 10dB and 5dB SNR levels

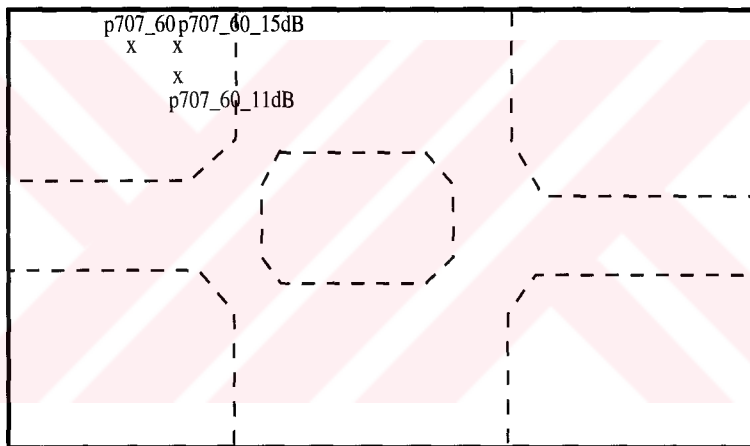


Figure 5.40: Noise analysis for B-707 aircraft at 60° aspect angle with 15dB and 11dB SNR levels

In aircraft class, the impulse response of B-707 aircraft at 60° aspect angle is contaminated with noise at the SNR levels of 15 dB and 11dB. Again, the original noise-free data corresponds to SNR of infinity. The shift in the location of the winning neuron (in the testing phase) for these data with respect to the level of added noise is presented in Figure 5.40. This time we observe that for the B-707 aircraft input data with SNR = 15 dB, the location of related winning neuron

deviates 1 unit right and for the same input data with $\text{SNR} = 11$ dB its location deviates 1 unit below and 1 unit right from the winning neuron location of the noise-free case. Again, using these noisy aircraft data in the testing phase does not affect the decision.



CHAPTER 6

CONCLUSIONS

In this thesis, we have designed mainly three target classification schemes by using the Wigner distribution for feature extraction part and a multilayer perceptron, self-organizing mapping network and principal component analysis for classification part. All of the proposed classifiers utilize the information in the time-frequency content of the signal to obtain the time-frequency localized energy feature vectors from the impulse response type input data. Then, by selecting mostly the features in the late-time bands, the resulting reduced feature vectors are used as input data of the classification parts of systems. The aim of this feature selection process is to decrease the system complexity and the total training time by preserving most of the useful information for classification. From the comparison of simulation results for problems 1 and 2 in Chapter 5, we see that to achieve acceptable classification accuracy, further signal processing for feature enhancement such as using the time-averaged WD matrices must be provided.

We have used an MLP trained with the backpropagation algorithm, the sta-

tistical PCA method with a minimum distance (or correlation) measure, and a SOM network trained with the forced winning neuron algorithm in the classification parts of the proposed classification systems. As presented in Chapter 5, we have performed the simulations using the sphere class and the aircraft class time domain data in these classification schemes.

The simulation results in Chapter 5 indicate that for both classes of available input data the WD+SOM classification scheme is the most promising one according to the relative classification performance and the total cpu processing time in testing phase of each classifier. For the sphere class, we have found the statistical PCA classifier performing better than the mostly used MLP classifier. However, for the realistic aircraft class, the performance of MLP is found slightly better. Specifically, the cpu processing time for the testing phase of the WD+SOM classifier is approximately 230 msec for sphere class and 250 msec for aircraft class, while for the WD+MLP classifier it is approximately 270 msec for both class inputs and for the WD+PCA classifier it is approximately 550 msec for sphere class and 620 msec for aircraft class. Also, it is not difficult to see that for the WD+SOM and WD+MLP classifiers the cpu processing time depends on the input vector length, not on the size of the training database, on the other hand, for the WD+PCA classifier it depends directly on the size of the training database as the disadvantage of this scheme.

As an additional study, after determining the class label of an input data we have shown the feasibility of estimating its aspect angle by training an aspect finder SOM for each sub-class using the multi-aspect data of that target only. Based on the results of simulation problems 5 and 9 in Chapter 5, it is certain

that better performances can be achieved by using a larger training database with additional aspect angles.

There are some advantages of the proposed classification schemes which are especially important for practical applications such as radar target classification. These systems can operate at reasonably high correct decision rates without large database of priori knowledge requirements. Their response times in the decision phase are short (fraction of a second) as expected and the software implementations of these classifiers are not complicated at all. Also, the robustness of the most promising classification scheme with respect to noisy inputs must be considered as another advantage.

The results of noise analysis, given in Chapter 5, for only the most promising classification scheme of WD+SOM, seem to be satisfactory but an extensive noise analysis, including all proposed classification schemes, for more dependable results is left as a future work. Training with noisy data and its effects on classifier performance is another topic for future research. In addition, the feasibility of aspect angle estimation can be improved by further research studies. A similar study of the wavelet transform for feature extraction and the resulting performances of the proposed classification schemes can be an interesting future work. Also, the proposed target classification schemes must be tested by larger databases to verify the results on the correct decision rates.

REFERENCES

- [1] Edward M. Kennaugh, "The K-Pulse Concept," *IEEE Transactions on Antennas and Propagation*, vol. AP-29, pp. 327–331, March 1981.
- [2] D. L. Moffatt and R. K. Mains, "Detection and Discrimination of Radar Targets," *IEEE Transactions on Antennas and Propagation*, vol. AP-23, pp. 358–367, May 1975.
- [3] C.W. Chuang and D. L. Moffatt, "Natural resonances via Prony's method and target discrimination," *IEEE Transactions on Aerosp. Electron. Syst.*, vol. AES-12, 1976.
- [4] C. Leonard Bennett and John P. Toomey, "Target Classification with Multiple Frequency Illumination," *IEEE Transactions on Antennas and Propagation*, vol. AP-29, pp. 352–358, March 1981.
- [5] J.S. Chen and E.K. Walton, "Comparison of two target classification techniques," *IEEE Transactions on Aerosp. Electron. Syst.*, vol. AES-22, pp. 15–21, Jan. 1986.
- [6] D. P. Nyquist K. M. Chen, E. Rothwell and B. Drachman, "Radar Target Discrimination using the Extinction-Pulse Technique," *IEEE Transactions on Antennas and Propagation*, vol. AP-33, pp. 929–937, Sept. 1985.
- [7] E. J. Rothwell K. M. Chen D. P. Nyquist, J. E. Ross and R. Bebermeyer, "A Radar Target Discrimination Scheme Using the Discrete Wavelet Transform for Reduced Data Storage," *IEEE Transactions on Antennas and Propagation*, vol. 42, pp. 1033–1038, July 1994.
- [8] Edward M. Kennaugh, "Opening Remarks," *IEEE Transactions on Antennas and Propagation*, vol. AP-29, pp. 190–191, March 1981.
- [9] E.M. Kennaugh and D.L. Moffatt, "Transient and Impulse Response Approximations," *Proceedings of the IEEE*, vol. 53, pp. 893–901, August 1965.
- [10] C.E. Baum. *The Singularity Expansion Method*, chapter 3, pp. 129–179. Springer-Verlag, 1976.
- [11] M.W. Roth, "Survey of Neural Network Technology for Automatic Target Recognition," *IEEE Transactions on Neural Networks*, vol. 1, pp. 28–41, March 1990.

- [12] Finnish Artificial Intelligence Society. *Visualizing the clusters on the Self-Organizing Map*. J. Iivari, T. Kohonen, J. Kangas, S. Kaski, 1994.
- [13] IEEE International Conference on Neural Networks. *Kohonen self-organizing maps: exploring their computational capabilities*, July 1988.
- [14] J. Edward Jackson. *A User's Guide to Principal Components*. Wiley Series, 1991.
- [15] Leon Cohen, "Time-Frequency Distributions-A review," *Proceedings of IEEE*, vol. 77, pp. 941-979, 1989.
- [16] J.B. Allen and L.R. Rabiner, "A unified approach to short-time Fourier analysis and synthesis," *Proc. IEEE*, vol. 65, pp. 1558-1564, 1977.
- [17] R. Altes, "Detection, estimation and classification with spectrograms," *J. Acoust. Soc. Am.*, vol. 67, pp. 1232-1246, 1980.
- [18] N.G. Bruijn. *Uncertainty Principles in Fourier Analysis*. Academic, New York, 1967.
- [19] G. Arens, E. Fourgeau, J. Morlet and D. Glard, "Wave Propagation and Sampling Theory," *Geophysics*, vol. 47, pp. 222-236, 1982.
- [20] I. Daubechies, "The Wavelet Transform, Time-Frequency Localization and Signal Analysis," *IEEE Trans. Info. Th.*, vol. 36, pp. 961-1005, 1990.
- [21] O. Rioul, "A discrete time multiresolution theory unifying octave-band filter banks. Pyramid and Wavelet Transforms," *IEEE Trans. Sig. Proc.*, 1990.
- [22] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Pat. Anal. Machine Intell.*, vol. 11, pp. 674-693, 1989.
- [23] S. Kadambe and G. F. Boudreaux-Bartels, "Application of the Wavelet Transform for pitch detection of speech signals," *IEEE Trans. Informat. Theory*, March 1992.
- [24] Int. Conf. Acoust., Speech, Sig. Proc. *Antonini, M., Barlaud, M., and Mathieu, P.*, Toronto, Canada, May 1991.
- [25] E. P. Wigner, "On the Quantum Correction for Thermo-Dynamic Equilibrium," *Physics Review*, vol. 40, pp. 749-759, 1932.
- [26] J. Ville, "Theorie et Applications de la Notion de Signal Analytique," *Cables et Transmission*, vol. 2 A, pp. 61-74, 1948.
- [27] T.A.C.M. Claasen and W.F.G. Mecklenbrauker, "The Wigner Distribution - A Tool for Time-Frequency Signal Analysis - Part I: Continuous time signals," *Philips J. Res.*, vol. 35, pp. 222-226, 1980.

- [28] T.A.C.M. Claasen and W.F.G. Mecklenbrauker, "The Wigner Distribution - A Tool for Time-Frequency Signal Analysis - Part II: Discrete time signals," *Philips J. Res.*, vol. 35, pp. 279–283, 1980.
- [29] Kai-Bor Yu and Siuling Cheng, "Signal Synthesis form Pseudo-Wigner Distribution and Applications," *IEEE Trans. Acoust., Speech and Signal Proc.*, vol. ASSP-35, pp. 1289–1301, 1987.
- [30] P. Flandrin, "Affine Smoothing of the Wigner-Ville Distribution," pp. 2455–2458. *IEEE Int. Conf. Acoust. Speech. Sig. Proc.*, 1990.
- [31] N.G. de Bruijn, "A theory of generalized functions, with applications to Wigner distribution and Weyl correspondence," *Nieuw Archief voor Wiskunde*, vol. XXI, pp. 205–280, 1973.
- [32] F. Hlawatsch and G.F. Boudreaux-Bartels, "Linear and Quadratic Time-Frequency Signal Representations," *IEEE SP Magazine*, vol. , pp. 21–67, April 1992.
- [33] W. Rihaczek, "Signal energy distribution in time and frequency," *IEEE Trans. Informat. Theory*, vol. IT-14, pp. 369–374, 1968.
- [34] C.H. Page, "Instantaneous Power Spectra," *J. Appl. Phys.*, vol. 23, pp. 103–106, 1952.
- [35] H.I. Choi and W.J. Williams, "Improved time-frequency representation of multicomponent signals using exponential kernels," *IEEE Trans. Acoust. Speech. Signal Processing*, vol. ASSP-37, 1989.
- [36] Gönül Turhan Sayan. *K-Pulse Estimation and Target Identification*. PhD thesis, Ohio State University, 1988.
- [37] Simon Haykin. *Neural Networks - A Comprehensive Foundation*, chapter 2, 6, 9, 10.
- [38] Richard P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, vol. 4, pp. 4–22, 1987.
- [39] D.E. Rumelhart and J.L. McClelland. *Parallel Distributed Processing*. MIT Press, 1986.
- [40] B. Widrow and M.A. Lehr, "30 years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation," *IEEE Proceedings*, August 1990.
- [41] L.W. Nolte Z.H. Michalopoulou and D. Alexandrou, "Performance Evaluation of Multilayer Perceptrons in Signal Detection and Classification," *IEEE Trans. on Neural Networks*, vol. 6, pp. 381–386, March 1995.
- [42] Teuvo Kohonen, "Self-organized Formation of Topologically Correct Feature Maps," *Biol. Cybern.*, pp. 59–69, 1982.

- [43] Teuvo Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, 3 edition, 1989.
- [44] Teuvo Kohonen, “The self-organizing map,” *Proceedings of the IEEE*, pp. 1464–1480, 1990.
- [45] Teuvo Kohonen Erkki Oja Olli Simula, Ari Visa and Jari Kangas. “Engineering applications of the self-organizing map,”. Manuscript submitted to a journal.
- [46] Bernd Fritzke, “Self-organizing Feature Maps with Problem Dependent Cell Structure,” in T. Kohonen, ed., *Artificial Neural Networks*, pp. 403–408. North-Holland, 1991.
- [47] Teuvo Kohonen. *Self-Organizing Maps*. Springer-Verlag, 1995.
- [48] K. Karhunen, “Über lineare methoden in der wahrscheinlichkeitsrechnung,” *Mathematica-Physica*, 1947. Transl: RAND Corp. T-131, 1960.
- [49] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison Wesley Publishing Comp., 1992.
- [50] Charles W. Therrien. *Decision, Estimation and Classification*. Wiley, 1989.

APPENDIX A

ALGORITHMS OF THE SIMULATION PROGRAMS

i- The Wigner distribution simulation program:

- Load input impulse response signal
- Normalize signal to unit total energy
- For time index n taking values from 0 to N_{max} , which is equal to signal length
 - for angular frequency θ_n taking values from θ_{min} to θ_{max} with discrete steps of $\frac{\theta_{max}-\theta_{min}}{N_\theta}$
 - set sum to zero
 - * for time shift index k taking values from $-N_{max}$ to N_{max} , compute $2 * \cos(2 * k * \theta_n) * f[n + k] * f[n - k]$ and add to sum, where f is the sampled input signal.

- put the resulting sum value into a 2-D array of $W[n][\theta_{index}]$, which corresponds to the discrete Wigner distribution of input signal.
- Then compute the time-averaged WD by summing $W[n][\theta_{index}]$ over pre-specified time bands.

Note: By use of the fast Fourier transform (FFT), the WD computation time can be effectively reduced.



ii- Multilayer Perceptron Classifier simulation programs:

For training,

- Load all training examples of a training set, including the input vectors and the desired output vectors which represent the corresponding classes of input vectors
- Initialize all weights of MLP network to random numbers in $[-0.1,0.1]$ and thresholds to 1
- Set initial learning rate and momentum constant values
- For repetition index taking values from 1 to maximum iteration number
 - randomly select an epoch of training example from the training set
 - for each neuron in the first hidden layer, calculate the neuron output using the Equations 3.2 and 3.3
 - again for each neuron in the second hidden layer, calculate the neuron output using the Equations 3.2 and 3.3
 - similarly for each neuron in the output layer, calculate the neuron output using the Equations 3.2 and 3.3
 - first calculate the error at each output neuron using Equation 3.4, and then find the sum of squared errors
 - calculate the average of sum of squared errors at the output of network up to iteration N by using Equation 3.5
 - adjust the weights in the output layer by using Equations 3.6 and 3.8

- adjust the weights in the second hidden layer by using Equations 3.7 and 3.8
- similarly adjust the weights in the first hidden layer by using Equations 3.7 and 3.8
- decrease the learning rate linearly with time as follows,

$$\alpha(t) = \alpha(0)[1 - t/(maxrep + 100)] \quad (\text{A.1})$$

- Finally, save all weights of MLP network.

For testing,

- Load the test input vector
- Load all weights of the trained MLP network
- For each neuron in the first hidden layer, calculate the neuron output using the Equations 3.2 and 3.3
- Similarly, for each neuron in the second hidden layer, calculate the neuron output using the Equations 3.2 and 3.3
- Finally, for each neuron in the output layer, calculate the neuron output using the Equations 3.2 and 3.3
- Determine the class of the input vector.

iii- Self-Organized Mapping Network Classifier simulation programs:

For training,

- Load all training examples of a training set, including the input vectors and their corresponding assigned winning neurons
- Initialize all weights of SOM network to random numbers in $[-0.1,0.1]$

In the first phase (ordering phase),

- Set initial learning rate, initial neighborhood radius, maximum update time and maximum repetition number accordingly,
- For repetition index taking values from 1 to maximum iteration number
 - randomly select an epoch of training example from the training set
 - update the weight of each neuron in the neighborhood of the specific winning neuron by using Equation 3.10
 - after maximum update time, decrease the size of neighborhood by one
 - calculate the output neuron values from the Euclidian norm of the input vector and the unit's weight vector as follows,

$$o_i = \sqrt{\sum_j (x_j - w_{ij})^2} \quad (\text{A.2})$$

In the second phase (fine-tuning phase),

- Set initial learning rate (much smaller than in the first phase), initial neighborhood radius (smaller than in the first phase), maximum update time and maximum repetition number (much greater than in the first phase) accordingly,

- Do the same steps as in the first phase
- Finally, save all weights of SOM network.

For testing,

- Load the test input vector
- Load all weights of the trained SOM network
- Calculate the output neuron values from the Euclidian norm of the input vector and the unit's weight vector as follows,

$$o_i = \sqrt{\sum_j (x_j - w_{ij})^2} \quad (\text{A.3})$$

- Determine the winning output neuron for the input vector by using Equation 3.9
- Then, determine the class of the input vector.

iv- Principal component analysis based classifier simulation programs:

The eigenvalues and the corresponding eigenvectors of input data vector is obtained by use of the Tooldiag simulation software package. By selecting only the eigenvectors corresponding to dominant eigenvalues, first the feature vectors are formed. Then by using the test simulation program, whose algorithm is given below, classification is accomplished.

For classification,

- Load the input feature vector to be tested
- Load the input feature vector whose class is known
- Calculate the Euclidian distance between two vectors
- Calculate the cross-correlation of two vectors
- After calculating two measures with different feature vectors corresponding to different classes in the second step, the class of input test vector is determined.

APPENDIX B

GENERATION OF NOISY DATA

In this thesis, the original sphere class and aircraft class data are used in simulations. The signal-to-noise ratio (SNR) of the sphere class data, which are synthetically generated [36], equals infinity. The SNR of the aircraft class data, which are measured on the OSU compact measurement range [36], is finite but much larger than the SNR of a full scale radar measurement. The noisy data used in Chapter 5 for noise performance testing is produced by adding uncorrelated white Gaussian noise to the real and imaginary parts of the frequency spectra of the original data. The SNR of the noisy data is defined as [36],

$$SNR \text{ in dB} = 10 \log_{10} \left[\sum_{i=1}^N |H_i|^2 / 2N\sigma^2 \right] \quad (\text{B.1})$$

where H_i is the complex value of the original signal spectrum at sampling frequency f_i , N is the total number of used samples and σ is the standard deviation of random Gaussian noise.