Purdue University Purdue e-Pubs

ECE Technical Reports

Electrical and Computer Engineering

5-1-1993

A Neural Network Approach to Stationary Target Discrimination

Douglas J. Munro Purdue University School of Electrical and Computer Engineering

Okan K. Ersoy Purdue University School of Electrical and Computer Engineering

Follow this and additional works at: http://docs.lib.purdue.edu/ecetr

Munro, Douglas J. and Ersoy, Okan K., "A Neural Network Approach to Stationary Target Discrimination" (1993). *ECE Technical Reports*. Paper 32. http://docs.lib.purdue.edu/ecetr/32

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

A NEURAL NETWORK APPROACH TO STATIONARY TARGET DISCRIMINATION

Douglas J. Munro Okan K. Ersoy

TR-EE 93-21 May 1993



School of Electrical Engineering Purdue University West Lafayette, Indiana 47907-1285 A Neural Network Approach to Stationary Target Discrimination

Douglas J. Munro

and

Okan K. Ersoy

Purdue University School of Electrical Engineering West Lafayette, Indiana 47907

TABLE OF CONTENTS

LIST OF TABLES	iv
LIST OF FIGURES	v
ABSTRACT	vi
CHAPTER 1 - INTRODUCTION	1
1.1 Introduction1.2 Outline of Thesis	1 2
CHAPTER 2 . LITERATURE REVIEW	4
 2.1 Introduction	4 4 9
CHAPTER 3 A NEURAL NETWORK APPROACH TO STATIONARY TARGET DISCRIMINATION	16
3.1 Introduction 3.2 System Overview 3.3 Derivation of the Likelihood Ratio Weighting Function	16 17
 For a Two Class Problem. 3.4 Modification of the Backpropagation Algorithm. 3.5 Evaluation of the Likelihood Ratio Weighting Function. 3.6 Determining the Decision Threshold θ. 3.7 Data Sets Used In Simulations. 	19 26 29 34 37
CHAPTER 4 _ RESULTS	45
 4.1 Introduction	45 45 48 55

Page

CHAPTER 5 - CONCLUSIONS AND RECOMMENDATIONS	60
5.1 Discussion 5.2 Conclusions	60 64
CHAPTER 6 - FUTURE RESEARCH	65
6.1 Discussion	65
LIST OF REFERENCES	67

. .

LIST OF TABLES

Table	2	Page
4.1	Failure of conventionally trained network for the case $P^{*}[H_{1}] \ll P[H_{1}]$	49
4.2	Performance of conventionally trained network for the case $P^{*}[H_1] = P[H_1]$.50
4.3	Network performance with LRWF while attempting to maintain a constant value of P_D	51
4.4	Network performance with LRWF while attempting to maintain a constant value of FAR	52
4.5	Performance of conventionally trained neural network in a varying SCR environment	56
4.6	Performance of the neural network in a varying SCR environment while incorporating the LRWF	57

LIST OF FIGURES

Figu	re	Page
2.1	False alarm and detection probabilities	6
2.2	Single-pulse linear detector with cell-averaging CFAR	- 8
2.3	Block diagram representation of a sign detector	11
3.1	Schematic diagram of the proposed target discrimination scheme	. 17
3.2	Illustration of target and clutter p.d.f. overlap	30
3.3	Mapping characteristic for several training distributions	. 33
3.4	Target only return signal,	40
3.5	Clutter only return signal	41
3.6	Target plus clutter return signal	42
4.1	Contrasting performance in terms of the P _D values presented in Tables 4.1 through 4.4	. 53
4.2	Contrasting performance in terms of the FAR presented in Tables 4.1 through 4.4	54
4.3	Contrasting conventional and LRWF performance in terms of P_D over a 30 dB range of SCR	. 58
4.4	Contrasting conventional and LRWF network performance in terms of FAR ov a 30 dB range in SCR	er 59

ABSTRACT

The problem which motivated this research was that of stationary target identification (STI) with millimeter wave seekers in a heavy clutter environment. While investigating the use of neural networks to **perform** target discrimination phase for **STI** problem, we began to search for a method to reduce the computational overhead associated with training a neural network to recognize low probability events. Our search yielded the development of a likelihood ratio weighting function (LRWF), which is very similar to the weighting function used in importance sampling techniques employed in the simulation of digital communication systems. By incorporating the LRWF into the backpropagation algorithm, we were able to significantly reduce the computational burden associated with training a neural network to recognize events which occur with low probability. This reduction in computational overhead is realized due to the reduction in the size of the data sets required for training.

CHAPTER 1

INTRODUCTION

1.1 Introduction

The stationary target identification (STI) problem can be divided into three distinct phases, namely: (1) detection, (2) discrimination, and (3) recognition. The detection phase is a term used to describe the process by which the presence of a target is sensed while the return signal from the target is embedded in the presence of background clutter, atmospheric noise, and/or noise generated within the radar receiver itself. Potential targets of interest are usually separated from the noise and clutter returns by various constant false alarm rate (CFAR) processing techniques. The discrimination phase distinguishes between actual target returns and strong target-like clutter returns which were passed as potential targets during the detection phase. The recognition phase, the most demanding of waveform and signal processor design, identifies the targets of interest from the features gathered from the return signal during the previous two phases.

The research presented in this thesis focuses on the detection phase of the overall **STI** problem. **As** previously mentioned, the detection phase of the STI problem is usually implemented in the **form** of a CFAR processor. There are generally two classes of CFAR processors, parametric and non-parametric, which is sometimes called a distribution free

CFAR processor. **A** parametric CFAR processor is one which is specifically designed for an assumed clutter distribution and which performs well with this type of interference. However, a non-parametric CFAR processor, which is not designed for a specific clutter distribution, works fairly well for a wide variety of clutter distributions. The parametric CFAR processor exhibits superior performance over non-parametric techniques if the clutter environment is known and uniformly homogenous. However, if the clutter environment is unknown or contains many transitions from one type of distribution to another, the non-parametric CFAR processor would be the better choice.

The approach presented in this thesis involves using a neural network to construct a **non**parametric CFAR processor. A neural network is used to form a weighted least squares estimate of the probability of a target being present or absent while the target return signal is embedded in a background clutter process. These estimates are used to construct a likelihood ratio test with a fixed threshold which is calculated in such a manner as to minimize the Bayes risk.

1.2 Outline of Thesis

Chapter 2 discusses parametric and non-parametric CFAR processing techniques in preparation of contrasting their performance to that of a neural-network based classifier. Chapter **3** presents the neural network non-parametric processor. Contained in this chapter are discussions regarding training procedures, adaptive thresholding, and data sets used in computer simulations. Chapter 4 contains the results of the computer simulations with the neural network classifier. Presented in chapter **5** are discussions contrasting the performance of the neural network classifier to that of a linear detector for the case of

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

This chapter is a review of the most common types of parametric and non-parametric CFAR processors, namely, the cell averaging CFAR processor and the sign detector CFAR processor. While the parametric CFAR processor performs superior to the **non**-parametric CFAR processor when operated in the assumed clutter environment, its performance rapidly degrades when the actual clutter environment does not correspond to the one assumed when the processor was designed. This is the advantage of the **non**-parametric CFAR processor which makes weak assumptions about the statistics of the clutter environment within which it will be operating.

2.2 Parametric CFAR Processor

One of the most common parametric CFAR processors is the cell averaging CFAR processor. The cell averaging CFAR processor provides estimates of the linear detection thresholds, T, by forming an estimate of the expected value of the decision statistic E[D|n], for the resolution cell under test while all potential targets are assumed absent. This estimate is formed by averaging the decision statistics, D|n, of the resolution cells leading, trailing, or surrounding the cell under test.

4

In order to perform the required analysis to **determine** an expression for the detection threshold **T**, it is usually assumed that the targets of interest are being detected in an exactly known additive white Gaussian noise **environment[5][6]**. With this assumption, the output of the matched filters, which are matched to the in-phase and the quadrature components of the return signal, will also be Gaussian random variables in the case of target absence. These samples are then passed through an envelope detector to **form** the decision statistic for the **i**th resolution cell (range-Doppler) as

$$D_{i} = \sqrt{I_{i}^{2} + Q_{i}^{2}} , \qquad (2.1)$$

ł

where D_i , I_i , and Q_i represent the envelope sample, in-phase component, and quadrature component associated with the **i**th resolution cell respectively.

From basic probability theory we know that D_i will be a Rayleigh distributed random variable with a probability density function of

$$f_{D|n}(D_i) = \begin{cases} \frac{D_i}{\sigma^2} e^{-(\frac{D_i^2}{2\sigma^2})}, D_i \ge 0\\ 0, D_i < 0 \end{cases}$$
(2.2)

where σ^2 is the variance of the Gaussian random variables I_i and Q_i . Also, D_i is a sample drawn from the clutter envelope given by Eq. (2.1) under the assumption of no target present.

Under the assumption of a target present, the mean of the resulting process will generally be greater than the mean of the clutter only process. This is depicted below in Figure 2.1 where $f_{D|n}(D)$ represents the conditional probability density function of the envelope

I.

sample D and $f_{D|y}(D)$ represents the conditional probability density function of the envelope sample under the assumption of a target present.



Figure 2.1 False alarm and detection probabilities.

Hence, the threshold required to obtain a given value of false alarm probability can be calculated as

$$P_{fa} = \int_{T}^{\infty} f_{D|n}(D_i) dD$$

$$= -e^{-(\frac{D_i^2}{2\sigma^2})} \Big|_{T}^{\infty}$$

$$Ln(P_{fa}) = -\frac{T^2}{2\sigma^2}$$

$$T = \sqrt{-2\sigma^2 Ln(P_{fa})}$$

(2.3)

The mean of a Rayleigh distributed random variable can be expressed as

$$\mathbf{E}[\mathbf{D}|\mathbf{n}] = \sqrt{\frac{2}{\pi}} \,\boldsymbol{\sigma} \,. \tag{2.4}$$

Hence, the result produced by Eq. (2.3) can be expressed as

$$T = E[D|n] \sqrt{\frac{-4}{\pi} Ln(P_{fa})} . \qquad (2.5)$$

Thus, we have reduced the problem of finding an estimate of the optimum threshold, T, for a given value of P_{fa} to that of forming an estimate of E[D|n].

Since the in-phase and quadrature components are assumed to be drawn from an unvarying white Gaussian noise environment, the samples, D_i , drawn from the clutter envelope are taken to be independent identically distributed (i.i.d.) random variables. However, in practical environments the Rayleigh parameter, σ , will vary with time according to the terrain, weather conditions, etc. ... Therefore, the statistics of the clutter samples can only be viewed as locally stationary in the neighborhood of the resolution cell under test. An estimate of E[D|n], denoted as E'[D|n], is then formed as the sample mean of the resolution cells in the vicinity of the resolution cell under test. These estimates are then used to form an estimate of the value of the detection threshold, T, given by Eq. (2.5) as

$$T = E'[D_i |n] \sqrt{\frac{-4}{\pi} Ln(P_{fa})},$$
 (2.6)

where $\mathbf{E}'[\mathbf{D}|\mathbf{n}]$ is formed as the sample mean given by

single pulse detection in terms of the probability of detection, false alarm rate, and signal to noise ratio (SNR). Lastly, presented in chapter 6 is a discussion of future research.

$$E'[D|n] = \frac{1}{K} \sum_{i=1}^{K} D_i . \qquad (2.7)$$

The value of K in the above equation represents the CFAR window size. That is, the total number of samples used to construct an estimate of E[D|n]. A block diagram representation of a cell averaging CFAR processor is shown below in Figure 2.2.



Figure 2.2 Single-pulse linear detector with cell-averaging CFAR.

2.3 Non-Parametric CFAR Processors

The fundamental structure of non-parametric detectors involves the transformation of a clutter or noise only input data set into a decision statistic that can be compared against a fixed threshold to establish a constant false alarm rate under weak assumptions on the statistical character of the background noise or clutter environment. Transformations accomplishing this function generally out perform an optimal parametric detection strategy derived under more strict conditions imposed on the background clutter process when the more strict conditions are false.

Generally, most non-parametric detection strategies are modifications of the sign detector [5]. The sign detector operates by providing a test for the positive median shift in the return signal under the condition of a target being present. However, to accomplish this test, the sign detection strategy assumes that the phase of the return signal is known exactly. Since this is not possible in a practical radar system, most realizable non-parametric detection strategies are sub-optimal modifications of the sign detector.

Since the sign detection strategy is used in conjunction with coherent and non-coherent pulse train signals, these signals can be defined as follows. Suppose that $s_i(t)$, i = 1, ..., N is a coherent or a non-coherent pulse train of N received, narrow-band, radar signals of constant width and pulse repetition interval. Further suppose that $s_i(t)$ represents the radar signature of the target of interest. The signal $s_i(t)$ can be expressed as

$$s_i(t) = A \cos(\omega_d t + \phi_i) \quad (i-1)T_p \le t \le (i-1)T_p + \tau$$
(2.8)

where,

- A = Amplitude of the ith pulse for a non-coherent or coherent pulse train
- Doppler frequency in the return signal $\omega_d =$
- ϕ_i = Phase of the ith pulse T_p = Pulse repetition interval
- τ = Pulse width .

Define the signal which is actually observed as $v_i(t)$, i = 1, ..., N, where $v_i(t)$ represents the ith observation taken in some range cell following the transmission of the ith pulse of a coherent or a non-coherent pulse train. This signal can now be expressed in terms of the return signal from a potential target, $s_i(t)$, as

$$\mathbf{v}_{i}(t) = \begin{cases} \mathbf{s}_{i}(t) + \mathbf{n}_{i}(t) & \text{if a target signal is present in the} \\ \mathbf{n}_{i}(t) & \text{if a target signal is absent in the} \\ \mathbf{n}_{i}(t) & \text{if a target signal is absent in the} \\ \text{range cell under test} \end{cases}$$
(2.9)

where $n_i(t)$ represents the background clutter process in the range cell under test over the ith single-pulse observation interval.



Figure 2.3 Block diagram representation of a sign detector.

Refer to Figure 2.3, the block diagram representation of a sign detector. From Figure 2.3, it is seen that $\mathbf{v_i}(t)$ is **first** passed through a matched filter which is matched to the return signal from the target of interest, $\mathbf{s_i}(t)$. This matched filtering operation is equivalent to that of a **running** time **correlator** when the output of the filter is sampled at $t = (i-1)T_p + \tau$ for i = 1, ..., N which effectively is sampling at times of maximum correlation with the signal $\mathbf{s_i}(t)$. For detection within the current test cell, let $\mathbf{w_i}$ denote the *i*th sample of the matched filter output. This can be expressed as

$$w_{i} = \kappa \int_{(i-1)T_{p}}^{(i-1)T_{p}+\tau} v_{i}(t)s_{i}(t) dt , \qquad (2.9)$$

where κ denotes the gain of the matched filter. The sampled signal, w_i , is then passed through the function $\mu(w_i)$ which is defined as

$$\mu(\mathbf{w}_{i}) = \begin{cases} 1 & \text{if } \mathbf{w}_{i} \geq 0 \\ 0 & \text{else} \end{cases}$$
(2.10)

This function effectively quantizes the existence of a positive correlation between the observed signal $v_i(t)$ and the signal of interest $s_i(t)$. These values are then summed over each pulse of the entire pulse train to form the decision statistic D.

In order for the sign detector to be on optimal Bayes' detection strategy, the following assumptions must be made.

Т

1

- 1. $\mathbf{w}_i | \mathbf{n}$, which denotes the **i**th sample of the matched filter output under the assumption of no target present, must be a set of independent identically distributed random variables.
- 2. The probability density function of $\mathbf{w}_{ij}\mathbf{n}$ has zero median value.

3.
$$\kappa \int_{(t-1)T_p}^{(i-1)T_p+\tau} s_i^2(t) dt$$
 is constant for all i.

Under the first assumption, the decision statistic, D, will on average be equal to N/2 in the case of $s_i(t)$ being absent from the current range cell under test. This conclusion stems from the background noise process, $n_i(t)$, showing positive correlations with $s_i(t)$ with probability 1/2. Hence for the case of $s_i(t)$ being absent from the current range cell under test, the formation of the decision statistic D can be viewed as a "coin-flipping" situation. For a fair coin, which corresponds to the background noise process $n_i(t)$, the coin will show heads or tails (positive or negative correlation) with a probability of 1/2.

Consider the following expression for \mathbf{w}_i and recall that it was assumed that the probability density function of $\mathbf{w}_i | \mathbf{n}$ has zero median value.

$$\mathbf{w}_{i} = \kappa \int_{(i-1)T_{p}}^{(i-1)T_{p}+\tau} \mathbf{v}_{i}(t) \mathbf{s}_{i}(t) dt = \begin{cases} \kappa \int_{(i-1)T_{p}}^{(i-1)T_{p}+\tau} \mathbf{n}_{i}(t) \mathbf{s}_{i}(t) dt & (2.11) \\ \kappa \int_{(i-1)T_{p}}^{(i-1)T_{p}+\tau} \mathbf{s}_{i}^{2}(t) dt + \kappa \int_{(i-1)T_{p}}^{(i-1)T_{p}+\tau} \mathbf{n}_{i}(t) \mathbf{s}_{i}(t) dt \end{cases}$$

Clearly from Eq. (2.11) it is apparent that the median value of $\mathbf{w}_i | \mathbf{y}$ will be greater than the median value of $\mathbf{w}_i | \mathbf{n}$ which is assumed to be equal to zero. It is this positive shift in the value of \mathbf{w}_i which the sign detection strategy is designed to detect.

Also under the first and third assumptions it can be shown [5] that the sign detector defined by Eq. (2.15) is equivalent to a Bayes' likelihood ratio test. Hence the sign detector depicted in Figure 3 is an optimal Bayes' detection strategy.

The problem now becomes one of choosing the proper threshold T with which to compare to the decision statistic D in order to obtain some desired value of the false alarm rate P_{fa} .

Since w_i , i=1, ..., N, are assumed to be independent, identically distributed random variables, it follows that the decision statistic D can be characterized by a binomial distribution with parameter p as

$$Pr(D = k) = \frac{N!}{k!(N-k)!} p^{k}(1-p)^{N-k} \qquad k = 0, 1, \dots, N , \qquad (2.12)$$

where the parameter p is defined as

$$p = Pr(w_i > 0)$$
 . (2.13)

For the case of no target present, the values of w_i will be greater than zero with probability 1/2. Therefore, the threshold T can be determined by the solution of

$$P_{fa} = Pr(D \ge T \mid \text{no target present})$$

$$= \sum_{k=T}^{N} \frac{N!}{k!(N-k)!} p^{k}(1-p)^{N-k}$$

$$= \sum_{k=T}^{N} \frac{N!}{k!(N-k)!} (1/2)^{N}$$

$$= (1/2)^{N} \sum_{k=T}^{N} \frac{N!}{k!(N-k)!}$$
(2.14)

With both the decision statistic D and the value of the threshold T determined, the presence or absence of a target within the range cell under test is determined as

If
$$D \ge T$$
 then decide a target is present (2.15)
else decide a target is absent

CHAPTER 3

A NEURAL NETWORK APPROACH TO STATIONARY TARGET DISCRIMINATION

3.1 Introduction

This chapter is organized seven sections, the second of which, Section 3.2, contains a description of the proposed target discrimination scheme. Issues addressed in this section includes the training/testing data sets, training of the neural network, what quantities are actually being estimated by the neural network, and how these quantities are used to implement an optimal Bayes detection strategy. Contained in Section 3.3 is the derivation of the likelihood ratio weighting function (LRWF) which, when incorporated into the training algorithm, allows for a reduction in the size of the data sets used for training. Presented in Section 3.4 is a description of the modifications made to the training algorithm which were required in order to incorporate the LRWF. Section 3.5 contains a discussion of the weak assumptions made regarding the conditional probability density functions of the return signal with target present or absent. These assumptions allow the LRWF to be expressed in terms of the a *prwri* probabilities of target presence or absence in the training and testing data sets. Presented in Section 3.6 is a discussion of the likelihood ratio test used to form an optimal Bayes detection strategy. This chapter concludes with Section 3.7 which contains a description of the data sets used to model the return signal from a range-only millimeter wave radar.

16



Figure 3.1 Schematic diagram of the proposed target discrimination scheme.

In the above figure the input training/testing vectors of the neural network, \mathbf{x}_k , $\mathbf{k} = 1 \dots n$, consist of samples drawn from a statistical model of the return signal from a range only radar. This model, which is described in detail in Section 3.7, is based on the input vector \mathbf{x}_k and the desired output vector \mathbf{U}_k . The data vector \mathbf{Y}_k generated by concatenating \mathbf{x}_k and \mathbf{U}_k can be written as

$$Y_k = (x_k, U_k),$$
 (3.1)

where,

$$U_{k} = (U_{k}^{0}, U_{k}^{1}) = \text{Desired Output Vector} = \begin{cases} (0,1) \text{ Target Present} \\ (1,0) \text{ Target Absent} \end{cases}$$
(3.2)

Т

Also note that the conditional probability density function (p.d.f.) of $\mathbf{x}_{\mathbf{k}}|\mathbf{U}_{\mathbf{k}} = (1,0)$ and the conditional p.d.f. of $\mathbf{x}_{\mathbf{k}}|\mathbf{U}_{\mathbf{k}} = (0,1)$ are distinct.

The training of the multi-layered neural network is carried out with the backpropagation algorithm. The backpropagation algorithm is a supervised learning algorithm involving the presentation of input/output pairs $(\mathbf{x_k}, \mathbf{U_k})$. The algorithm attempts to minimize the square error between the desired output vector, $\mathbf{U_k}$, and the actual output vector. By so doing, the algorithm actually forms a mean squared error estimate of the conditional probabilities $\mathbf{P}(\mathbf{H_i}|\mathbf{x})$ where the vector x is a sample drawn from the ensemble of all possible input vectors χ .

After the network has been trained, the weights matrices are held fixed, and the network is used to classify the current range cell of the input vector $\mathbf{x}_{\mathbf{k}}$ according to target presence or absence. This is done by using the estimates of the conditional probabilities $\mathbf{P}(\mathbf{H}_1|\mathbf{x})$ and $\mathbf{P}(\mathbf{H}_0|\mathbf{x})$, where \mathbf{H}_0 and \mathbf{H}_1 represent the hypothesis of target absence or presence respectively, to form a likelihood ratio test. The likelihood ratio test formed by taking the ratio of $\mathbf{P}(\mathbf{H}_1|\mathbf{x})$ to $\mathbf{P}(\mathbf{H}_0|\mathbf{x})$ is actually a Bayes detector. If the losses associated with an incorrect decision are expressed as,

> L_{01} = Loss associated with deciding H_0 when H_1 is in force L_{10} = Loss associated with deciding H_1 when H_0 is in force,

and the losses associated with a correct decision are both set equal to zero, $L_{00} = L_{11} = 0$, the resulting likelihood ratio test is actually an optimum Bayes detection strategy with the Bayes risk minimized for a threshold choice of $\theta = \frac{L_{01}}{L_{10}}$.

There are, however, problems to be faced when implementing this scheme. The first and obvious problem is that of training the neural network. If the neural network is to be trained to operate in a realistic scenario, the size of the required data sets will become cumbersomely large. This is due to the low probability of target occurrence, **P**[H₁], in any realistic scenario. This problem led to the development of a technique which allows one to train the neural network utilizing data sets with a much higher $P[H_1]$, but the same network can be used to classify data with a much lower value of P[H₁]. This technique, which is very similar to importance sampling (IS) techniques used during the simulation of digital communication systems to estimate bit error rates (BER), will reduce the size of the required data sets and result in a substantial savings in the computational overhead during the training procedure. This technique involves constructing a likelihood ratio weighting function (LRWF) which, when incorporated into the backpropagation algorithm, forces the algorithm to form its estimates of the conditional probabilities of target presence and absence as if the targets were occurring with a probability of $\mathbf{P}^{*}[\mathbf{H}_{1}]$ rather than the probability of target occurrence associated with the training data set, $P[H_1].$

3.3 Derivation of the Likelihood Ratio Weighting Function For a Two Class Problem

Assuming a two class problem, target present/absent, define the k^{th} , k = 1, ..., n, complete data vector as

$$Y_k = (x_k, U_k),$$
 (3.3)

where,

$$Ur = (U_k^0, U_k^1) = Desired Output Vector = \begin{cases} (0,1) Target Present \\ (1,0) Target Absent \end{cases}$$
(3.4)

Also note that the conditional probability density functions (p.d.f.) of the input vectors $\mathbf{x}_{\mathbf{k}} | [\mathbf{U}_{\mathbf{k}} = (0,1)]$ and $\mathbf{x}_{\mathbf{k}} | [\mathbf{U}_{\mathbf{k}} = (1,0)]$ are distinct. Furthermore, define the ensemble of all possible input data vectors as χ , where the set { $\mathbf{x}_{\mathbf{k}} | [\mathbf{U}_{\mathbf{k}} = (0,1)] \cup \mathbf{x}_{\mathbf{k}} | [\mathbf{U}_{\mathbf{k}} = (1,0)] \} \in \chi$.

Define the outputs of nodes 0 and 1 of the multilayer neural network as

$$F_0(\mathbf{x}_k, \mathbf{w}) = \begin{cases} 0 \text{ Target Present} \\ 1 \text{ Target Absent} \end{cases}$$
(3.5)

$$F_1(\mathbf{x}_k, \mathbf{w}) = \begin{cases} 1 \text{ Target Present} \\ 0 \text{ Target Absent} \end{cases}$$
(3.6)

where $F_i(x_k, w) \in \Re$, 0 s $F_i(x_k, w)$ s 1, and w represents the weights matrices of the neural network.

As an example of supervised learning based on least-squares, the backpropagation algorithm will be used. The backpropagation algorithm is a supervised learning algorithm involving the presentation of training and testing sets of input and output patterns. The algorithm attempts to minimize the square error given by the actual and the desired output values summed over the output nodes and all training pairs of **input/output** vectors [1]. Using the previously established notation, for a network with two output nodes this error can be expressed as [3]

$$E_{s}^{n}(w) = \frac{1}{n} \sum_{k=1}^{n} \left[(F_{0}(x_{k}, w) - U_{k}^{0})^{2} + (F_{1}(x_{k}, w) - U_{k}^{-1})^{2} \right], \quad (3.7)$$

where **n** represents the total number of vectors contained in the training set. Eq. (3.7) represents the error to be minimized in the least squares sense. It will be modified below by the LRWF in the weighted least squares sense.

Define the LRWF as

$$g(x) = \frac{f_{x}^{*}(x)}{f_{x}(x)} = \frac{P^{*}[H_{0}] f_{x}(x|H_{0}) + P^{*}[H_{1}] f_{x}(x|H_{1})}{P[H_{0}] f_{x}(x|H_{0}) + P[H_{1}] f_{x}(x|H_{1})},$$
(3.8)

where $f_x(x)$ represents the probability density function (p.d.f.) of $x, x \in \chi$, for the training set; $f_x^*(x)$ represents the p.d.f. for the testing set; $P[H_0]$ and $P[H_1]$ are the a *priori* probabilities of target being absent and present, respectively. Note that $P^*[H_1] \ll P[H_1]$ in the present application.

The backpropagation algorithm will be modified by minimizing the new error function defined as

$$\widehat{E}_{s}^{n}(w) = \frac{1}{n} \sum_{k=1}^{n} \left[(F_{0}(x_{k}, w) - U_{k}^{0})^{2} + (F_{1}(x_{k}, w) - U_{k}^{1})^{2} \right] g(x_{k}) .$$
(3.9)

In this way, the neural network can be forced to form its mean-square error estimates of $P[H_i|x]$ according to the testing distribution $f_x^*(x)$ rather than the training distribution $f_x(x)$. Below, it is shown theoretically why this is the case.

The average error over the entire ensemble, $E_s(w)$, can be defined as

I.

$$E_{s}(w) \approx \lim_{n \to \infty} \widehat{E}_{s}^{n}(w), \qquad (3.10)$$

where 'n' represents the total number of vectors contained in the training set. Since the backpropagation algorithm seeks a minimum of the function $\widehat{\mathbf{E}}_{s}^{n}(\mathbf{w})$, the algorithm will also form an estimate to the minimum value of $\mathbf{E}_{s}(\mathbf{w})$. The accuracy of this estimate depends upon how accurately the training set models the actual statistics of the ensemble χ . If the training set poorly represents the statistics of χ , the minimization of $\widehat{\mathbf{E}}_{s}^{n}(\mathbf{w})$ will not correspond to a minimization of $\mathbf{E}_{s}(\mathbf{w})$, and poor classification performance during testing will result.

Eq. (3.10) can be rewritten using the number of vectors in each class as

$$E_{s}(w) = \lim_{n \to \infty} \left\{ \frac{n_{0}}{n} \cdot \frac{1}{n_{0}} \sum_{k=1}^{n} \left[(F_{0}(x_{k},w)-1)^{2} + F_{1}^{2}(x_{k},w)]g(x_{k})\delta(U_{k}^{0}) + \frac{n_{1}}{n} \cdot \frac{1}{n_{1}} \sum_{k=1}^{n} \left[F_{0}^{2}(x_{k},w) + (F_{1}(x_{k},w)-1)^{2}]g(x_{k})\delta(U_{k}^{-1}) \right] \right\},$$
(3.11)

where $\mathbf{n_0}$ and $\mathbf{n_1}$ represent the total number of vectors of the training set which are associated with target absence or presence respectively. Also note that $\delta(.)$ represents a Dirac delta function used to segment the training set. By the law of large numbers, as n, no, $\mathbf{n_1}$ increase, Eq. (3.11) can be rewritten as

$$\begin{split} E_{s}(w) &= P[H_{0}] \int_{x} \left[(F_{0}(x,w)-1)^{2} + F_{1}^{2}(x,w)]g(x)f_{x}(x|H_{0}) dx \\ &+ P[H_{1}] \int_{x} \left[F_{0}^{2}(x,w) + (F_{1}(x,w)-1)^{2}]g(x)f_{x}(x|H_{1}) dx \\ &= \sum_{i=0}^{1} \int_{x} F_{1-i}^{2}(x,w)g(x)f_{x}(x|H_{i})P(H_{i}) dx \\ &+ \sum_{j=0}^{1} \int_{x} \left[F_{j}(x,w)-1 \right]^{2}P(H_{j})g(x)f_{x}(x|H_{j}) dx . \end{split}$$
(3.12)

Using Bayes' formula, namely,

$$f_x(x|H_i) P(H_i) = P(H_i|x)f_x(x)$$
 (3.13)

The expression for $E_s(w)$ can be rewritten as

-

$$E_{s}(w) = \sum_{i=0}^{1} \int_{x} F_{1\cdot i}^{2}(x,w)g(x)f_{x}(x)P(H_{i}|x) dx + \sum_{j=0}^{1} \int_{x} [F_{j}(x,w)-1]^{2}g(x)f_{x}(x)P(H_{j}|x) dx .$$
(3.14)

Expanding the first term and the first term of the second summand yields

$$\begin{split} E_{s}(w) &= \int_{x} F_{1}^{2}(x,w)g(x)f_{x}(x)P(H_{0}|x) dx \\ &+ \int_{x} F_{0}^{2}(x,w)g(x)f_{x}(x)P(H_{1}|x) dx \\ &+ \int_{x} F_{0}^{2}(x,w)g(x)f_{x}(x)P(H_{0}|x) dx \\ &+ \int_{x} F_{1}^{2}(x,w)g(x)f_{x}(x)P(H_{1}|x) dx \\ &+ \int_{x} F_{1}^{2}(x,w)g(x)f_{x}(x)P(H_{1}|x) dx \\ &- \sum_{j=0}^{1} [2F_{j}(x,w)-1]g(x)f_{x}(x)P(H_{j}|x) dx . \end{split}$$
(3.15)

Rearranging terms and noting that $[P(H_1|x) + P(H_0|x)] = 1$ gives

$$E_{s}(w) = \sum_{i=0}^{1} \left[\int_{x} F_{i}^{2}(x,w)g(x)f_{x}(x) dx - \int_{x} [2F_{i}(x,w)-1]g(x)f_{x}(x)P(H_{i}|x) dx \right].$$
(3.16)

Noting that $[F_i(x,w) - P(H_i|x)]^2 = F_i^2(x,w) - 2F_i(x,w)P(H_i|x) + P^2(H_i|x)$, Eq. (3.16) can be expressed as

$$E_{s}(w) = \sum_{i=0}^{1} \left[\int_{x} (F_{i}(x,w) - P(H_{i}|x))^{2}g(x)f_{x}(x) dx + \int_{x} [P(H_{i}|x) - P^{2}(H_{i}|x)]g(x)f_{x}(x) dx \right].$$
(3.17)

Since $g(x) = \frac{f_x^*(x)}{f_x(x)}$ as defined by Eq. (3.8), Eq. (3.17) can be expressed as

$$E_{s}(w) = \sum_{i=0}^{1} \left[\int_{x} (F_{i}(x,w) - P(H_{i}|x))^{2} f_{x}^{*}(x) dx + \int_{x} [P(H_{i}|x) - P^{2}(H_{i}|x)] f_{x}^{*}(x) dx \right].$$
(3.18)

I.

Since the neural network, when trained using the backpropagation algorithm, is made to minimize the error function $\widehat{E}_{s}^{n}(w)$ with respect to w, it also attempts to minimize $E_{s}(w)$ with respect to w with the given training set. Since the first term of Eq. (3.18) is the only term which is a function of w, clearly the network is forming a minimum mean squared error approximation to the a *posteriori* probabilities $P(H_{i}|x)$. Note also that this minimum mean squared error approximation is being computed as if the feature vectors x were drawn from $f_{x}(x)$ rather than $f_{x}(x)$. Therefore the network is trained to recognize targets occurring with a probability P*[H₁] rather than $P[H_{1}]$ as desired.

It was mentioned in Section 3.2 that this method was very similar to importance sampling techniques used in conjunction with forming estimates of the bit error rate (BER) during the simulation of a digital communication system. The simulation being performed is of the standard Monte **Carlo** type. By employing IS techniques, the goal is to greatly reduce the simulation run time of the Monte **Carlo** simulation while achieving the same degree of accuracy in the estimate of the BER. This is done by biasing the input signal process to the system so as to artificially increase the BER. However, during the simulation this bias is removed from the estimate of the BER by weighting each error by a weighting function $\mathbf{w}(\mathbf{x}) = \frac{\mathbf{f}_{\mathbf{x}}^{\prime}(\mathbf{x})}{\mathbf{f}_{\mathbf{x}}(\mathbf{x})}$, where $\mathbf{f}_{\mathbf{x}}(\mathbf{x})$ represents the multi-dimensional **p.d.f.** of the input vector **x** associated with the biased input signal process and $f_x(x)$ represents the **p.d.f.** of the unbiased input signal process. Clearly there is a direct analogy between the IS weighting function and the LRWF $g(\mathbf{x})$. In the case of the IS weighting function $\mathbf{w}(\mathbf{x})$ the input signal process is biased so as to increase the BER of the system being simulated and in the case of the LRWF the input signal process is biased so as to increase the probability of target occurrence. In both cases the end result is to decrease the amount of run time required by the simulation/training procedure. For an excellent review of IS the reader is referred to a recent text by Jeruchim, et. al. [2].

3.4 Modification of the Backpropagation Algorithm

Contained in this section is the derivation of a modified **form** of the backpropagation algorithm which includes the **LRWF**. Here it is shown that the only modification which needs to be made is to include the weighting function, $g(\mathbf{x})$, in the computation of the error terms associated with the final output layer of the neural network.

Consider a specific weighted error, E_p , due to the presentation of the input vector p as

$$E_{p} = \frac{1}{2} \sum_{j} [Y_{pj} - F_{pj}^{N}(x_{p}, w)]^{2} g(x_{p}), \qquad (3.19)$$

where Y_{pj} is the jth component of the desired output vector due to the presentation of input vector p. The output of node j of the output layer, which is the Nth layer, is denoted as $F_{pj}^{N}(x_{p},w)$. The LRWF, evaluated at the present input vector, is defined by Eq. (3.8). The dependence of F_{pj}^{N} on the present input vector x_{p} and the weights, denoted by w, will be suppressed in the following notation.

The input to node j of the mth layer due to the presentation of input vector p is defined as

$$net_{pj}^{m} = \sum_{i} w_{ji}^{m} F_{pi}^{m-1} , \qquad (3.20)$$

$$F_{pj}^{m} = f(net_{pj}^{m}),$$
 (3.21)

where f(.) is a continuously differentiable, nondecreasing, nonlinear activation function such as a sigmoid.

The negative of the gradient vector components of the error E_p with respect to **net**^m are given by

$$\delta_{pj}^{m} = -\frac{\partial E_{p}}{\partial net_{pj}^{m}}.$$
(3.22)

Applying the chain rule allows this partial derivative to be written as

$$\delta_{pj}^{m} = -\frac{\partial E_{p}}{\partial net_{pj}^{m}} = -\frac{\partial E_{p}}{\partial F_{pj}^{m}} \frac{\partial F_{pj}^{m}}{\partial net_{pj}^{m}}.$$
(3.23)

The second factor can be easily computed from Eq. (3.21) as

$$\frac{\partial F_{pj}^{m}}{\partial net_{pj}^{m}} = f'(net_{pj}^{m}), \qquad (3.24)$$

which is simply the first derivative of the activation function evaluated at the present input to that particular node.

In order to compute the first term consider two cases. The first case is when the error signal is developed at the output layer N. This can be computed from Eq. (3.19) as

$$\frac{\partial E_p}{\partial F_{pj}^N} = -[Y_{pj} - F_{pj}^N]g(x_p).$$
(3.25)

Substituting Eqs. (3.24) and (3.25) into Eq. (3.23) yields

$$\delta_{pj}^{N} = [Y_{pj} - F_{pj}^{N}]g(x_{p})f'(net_{pj}^{N}).$$
(3.26)

For the second case, when computing the error terms for some layer other than the output layer, the δ_{pj} 's can be computed recursively from those associated with the output layer as

$$\frac{\partial E_{p}}{\partial F_{pj}^{m}} = \sum_{k} \left[\frac{\partial E_{p}}{\partial net_{pk}^{m+1}} \frac{\partial net_{pk}^{m+1}}{\partial F_{pj}^{m}} \right]$$

$$= \sum_{k} \left[\frac{\partial E_{p}}{\partial net_{pk}^{m+1}} \frac{\partial}{\partial F_{pj}^{m}} \sum_{i} w_{ki}^{m+1} F_{pi}^{m} \right]$$

$$= \sum_{k} \frac{\partial E_{p}}{\partial net_{pk}^{m+1}} w_{kj}^{m+1}$$

$$= -\sum_{k} \delta_{pk}^{m+1} w_{kj}^{m+1}.$$
(3.27)

.....

Combining this result with Eq. (3.24) gives

dented to an exception of the second second
$$\delta_{pj}^{m} = f'(net_{pj}^{m}) \sum_{k} \delta_{pk}^{m+1} w_{kj}^{m+1}.$$
(3.28)

These results can be summarized in 3 equations. First an input vector, x_p , is propagated through the network until an output is computed for each of the output nodes of the output layer. These values are denoted as F_{pj}^N . Next, the error terms associated with the output layer are computed by Eq. (3.26). The error terms associated with each of the other m-1 layers of the network are computed by Eq. (3.28). Finally, the weights are updated as

$$\Delta_{\rm p} \, {\rm w}_{ij}^{\rm m} = \eta \, \delta_{\rm pj}^{\rm m} \, F_{\rm pj}^{\rm m}, \qquad (3.29)$$

where η represents the learning rate of the network. Usually η is chosen to be some nominal value such as 0.01.

From [1], it is seen that the only change to the backpropagation algorithm is the inclusion of the likelihood ratio weighting function in Eq. (3.26). All other steps of the algorithm remain the same.

3.5 Evaluation of the Likelihood Ratio Weighting Function

The problem associated with directly evaluating the likelihood ratio weighting function given by Eq. (3.8) is that $f_x(x|H_0)$ and $f_x(x|H_1)$ must be known. If $f_x(x|H_0)$ and $f_x(x|H_1)$ were known, one could use a variety of parametric constant false alarm rate (CFAR) processing techniques. These techniques employ certain assumptions about the target and

clutter environment such as assuming that the p.d.f. of the return signal from clutter only can be modeled according to a Rayleigh, lognormal, or Weibull distribution[5]. Similarly, fluctuations in the target radar cross section (RCS) are often modeled according to one of the four Swerling fluctuating target models [5]. However, in this case, the generalization capabilities of the neural network are used to form estimates to these unknown conditional probability density functions based on the training data sets. Therefore, for this method it would be desirable to construct a way of evaluating the LRWF which makes as little a *priori* assumptions on the conditional probability density functions as possible. That is, develop a non-parametric processing scheme.

In the remainder of this thesis, the problem is restricted to case depicted in Figure 3.2.



Figure **3.2** Illustration of target and clutter **p.d.f.** overlap.

In the scenario depicted in Figure 3.2,

$$\overline{X_{t}} - \sigma_{t} > \overline{X_{c}} + \sigma_{c} , \qquad (3.30)$$

where $\overline{X_1}$ and $\overline{X_c}$ represent the mean of the target distribution and the mean of the clutter distribution, respectively; σ_t^2 and $\sigma_{\overline{c}}^2$ represent the target and the clutter distribution variance, respectively. It is assumed that the distributions are fairly well removed from one another as depicted above in Figure 3.2, such that Eq. (3.30) is true. This assumption is valid since this is the situation created by pulse integration techniques which are employed in most modern radar systems. Pulse integration techniques are based on the premise that the return signal from background clutter tends to be relatively uncorrelated pulse to pulse. By illuminating the target by many pulses and summing over these return signals the clutter response can be greatly reduced . With this assumption the evaluation of $g(\mathbf{x})$ can be greatly simplified.

With Eq. (3.30) satisfied, the following can be stated.

$$f_{\mathbf{x}}(\mathbf{x}|\mathbf{H}_1)\Big|_{\mathbf{x}\in\text{Target Absent}} \sim 0 \tag{3.31}$$

$$f_x(x|H_0)|_{x \in \text{Target Present}} \approx 0$$
 (3.32)

This allows the weighting function to be written as

$$g(\mathbf{x}) = \begin{cases} \frac{P^{*}[H_{0}]}{P^{*}[H_{1}]}, \text{ given } \mathbf{x} \in \text{Target Absent} \\ \frac{P^{*}[H_{1}]}{P[H_{1}]}, \text{ given } \mathbf{x} \in \text{Target Present} \end{cases}$$
(3.34)

1

This expression is used to evaluate the **LRWF** in all of the computer simulations presented in Chapter 4.

Therefore, the requirement of obtaining expressions for $f_x(x|H_0)$ and $f_x(x|H_1)$ has been reduced to requiring knowledge of the a *priori* probabilities $P^a[H_0]$ and $P^*[H_1]$. From Eq. (3.34) it can be seen that as the probability of target occurrence, $P^*[H_1]$, tends to zero the function g(x) tends to

$$\lim_{P^{*}[H_{1}] \to 0} g(\mathbf{x}) = \begin{cases} \frac{1}{P[H_{0}]} & \text{, given } \mathbf{x} \in \text{Target Absent} \\ 0 & \text{, given } \mathbf{x} \in \text{Target Present} \end{cases}$$
(3.35)

From the above observation one would expect a degradation in mapping performance as the mapping ratio, defined by

Mapping Ratio =
$$\frac{P[H_1]}{P^{\bullet}[H_1]}$$
 (3.36)

.....

tends towards infinity. This characteristic is depicted below in Figure 3.3. Note that the notation 'p_0.XX' refers to the probability of target occurrance, $P[H_1]$ equal to 0.XX.



Figure 3.3 Mapping characteristic for several different training distributions.

From Figure 3.3, one would expect excellent performance for mapping ratios in the range of 1.0 to 5.0 due to the linear nature of the characteristic in this region. However, as the mapping ratio exceeds this range, the ability of this method to distinguish between adjacent mapping ratios becomes increasingly difficult. In the **experimants** presented in chapter 4, it was found that for a training distribution with $P[H_1] = 0.248$, proper mappings were achieved for mapping ratios as large as 31.0. Above this value, little control over the probability of detection or the false alarm rate was achieved.

Ideally, the decision threshold θ would be determined in such a manner as to maintain a constant false alarm rate regardless of the clutter environment. Also, it would be desirable for θ to be independent of the *a priori* probabilities **P**[**H**₀] and **P**[**H**₁].

With the type of likelihood ratio test described in the overview, Section 3.2, these ideals are difficult if not impossible to achieve. In order to design a parametric type of CFAR processor as discussed in Section 2.2, the probability density function of the likelihood ratio itself would have to be known. If this were known one could develop a threshold according to **Eq.** (3.37) below.

$$P_{fa} = \int_{\theta}^{\infty} f_{l|H_0}(z) dz$$
(3.37)

where,

 $f_{1|H_0}(.) = Conditional probability density function of the ratio <math>P(H_1|x)$ $P(H_0|x)$ with H_0 in effect.

$$\mathbf{z} = \frac{\mathbf{P}(\mathbf{H}_1 | \mathbf{x})}{\mathbf{P}(\mathbf{H}_0 | \mathbf{x})} = \frac{\text{Output of Node 1}}{\text{Output of Node 0}}.$$

This would accomplish our first goal of maintaining a constant false alarm rate invariant to the clutter environment, assuming that the data set from which z is drawn adequately represents the statistics of the clutter process our CFAR processor encounters. However, from Bayes formula it is seen that z is a function of priors. This is illustrated by **Eq.** (3.38).

$$z = \frac{P(H_1|x)}{P(H_0|x)}$$

= $\frac{P(x|H_1|)P(H_1)}{P(x|H_0|)P(H_0)}$
= $\frac{f_{x|H_1}(x) dx P(H_1)}{f_{x|H_0}(x) dx P(H_0)}$
= $\frac{f_{x|H_1}(x) P(H_1)}{f_{x|H_0}(x) P(H_0)}$ (3.38)

Another disadvantage of developing a decision threshold, θ , via Eq. (3.37) is that assumptions would have to be made concerning the statistics of the clutter environment in order to arrive at a tractable result for the **p.d.f**, $f_{1|H_0}(.)$. This directly conflicts with one of our stated advantages of the neural network technique. Namely, that no such assumptions are required regarding the clutter environment. However, the neural network technique would be expected to exhibit a more graceful decay in performance as compared to the cell averaging CFAR processor when it is operating in a clutter environment which does not represent the one assumed during its design.

From Eq. (3.38) it is straight forward to see that the actual problem faced is that of optimizing the threshold choice for a Bayes detector. From previously established theory [8], the risk or cost associated with **performing** a hypothesis test of H_0 vs. H_1 can be written as

$$R(H) = \begin{cases} L_{00}P_{00} + L_{10}P_{10}, H = H_0 \text{ in effect} \\ L_{01}P_{01} + L_{11}P_{11}, H = H_1 \text{ in effect} \end{cases}$$
(3.39)

where,

 P_{ij} = Probability of choosing hypothesis i when hypothesis j is in effect. L_{ij} = Loss associated with choosing hypothesis i when hypothesis j is in effect. Normally the loss function L_{ij} is set equal to zero for i = j since no loss is incurred for a correct decision.

Averaging the risk function presented by Eq. (3.39) over $H = \{H_0, H_1\}$ and setting $L_{ii} = 0$ yields the Bayes risk as

$$E[R(H)] = R(p) = p_0 L_{10} P_{10} + p_1 L_{01} P_{01} , \qquad (3.40)$$

where $p_i = P[H_i]$.

The optimum choice of 8 which will minimize Eq. (3.40) is found by first noting the following relations.

$$\mathbf{P}_{10} = \prod \mathbf{f}_{\mathbf{x}|\mathbf{H}_0}(\mathbf{x}) \, \mathbf{d}\mathbf{x} = \text{Probability of False Alarm}$$
(3.41)

$$\mathbf{P}_{01} = 1 \cdot \text{Probability of Detection} = 1 \cdot \prod_{\mathbf{x} \mid \mathbf{H}_1(\mathbf{x}) \text{ dx}} \mathbf{f}_{\mathbf{x} \mid \mathbf{H}_1(\mathbf{x}) \text{ dx}}$$
(3.42)

Gathering Eqs. (3.41) and (3.42) into Eq. (3.40) gives

Red address and the second sec

$$R(p) = p_1 L_{10} + \int_{\theta}^{\bullet} \left[p_0 L_{10} f_{x|H_0}(x) - p_1 L_{01} f_{x|H_1}(x) \right] dx \quad . \tag{3.43}$$

Eq. (3.43) is minimized by including those measurement vectors x in the range 8 to ∞ which reduce the Bayes risk, **R(p)**. This is done whenever the integrand of Eq. (3.43) is negative. Hence the test becomes choosing those vectors such that

$$\{ x : \frac{f_{x|H_1}(x) P[H_1]}{f_{x|H_0}(x) P[H_0]} > \frac{L_{10}}{L_{01}} \} .$$
(3.44)

As with this and most other problems it is difficult to assign numerical values to L_{01} and L_{10} . Opting for the frequent assumption of $L_{01} = L_{10} = 1$ gives a decision threshold $\theta = 1$.

3.7 Data Sets Used In Simulations

In this section, the data sets used to model the return signal from a range-only millimeter wave (MMW) radar operating in a heavy clutter environment are described. The specific scenario being modeled is that of a lock-on-after-launch (LOAL) air-to-surface missile employing the MMW radar to provide high resolution target discrimination for the task of stationary target identification (STI) [6]. Our STI model is a simple two-class problem in which the objective is to determine which range cells within the **maximum** unambiguous range of the radar contain target returns.

The operating frequency of the radar was chosen to be 35 GHz due to the increased attention to this portion of the electromagnetic spectrum in air-to-surface applications. This attention is due to the favorable characteristics of the MMW frequency band such as smaller antenna beam-width and higher gain as compared to microwave frequencies, high resolution in range and Doppler, reduced electronic countermeasures (ECM) vulnerability, and a reduction in multipath and ground clutter at low elevation angles [6]. The pulse repetition frequency (PRF) of the radar was chosen as 30 kHz in order to set the maximum unambiguous range of the radar to be equal to 5 km. The radar waveform itself was chosen as a coherent pulse train with the pulse bandwidth of the radar set equal to 30 MHz. The pulse bandwidth of the radar was chosen so as to enable the radar to

resolve individual point scatterers separated by distances as small as 5 meters. For simulation purposes, these specifications **fix** the size of the array which represents the range returns for a single radar pulse to be **2000** elements. Therefore, each element of this array represents a single range cell with each cell being **2.5** meters in length. Recall that in this application it is assumed that the targets are stationary and as such exhibit zero Doppler shift in the return signal. Also, it is assumed that the Doppler shift in the return signal caused by the motion of the **platform** upon which the radar is mounted has been negated. Hence, every element of our array will be a random variable, $r \in \Re$, drawn from a probability distribution representing the return signal from clutter only or target only.

The terrain which the MMW seeker is traversing is assumed to be that of a heavily forested region containing deciduous trees during summer. As such, the region illuminated by the radar is assumed to consist of a large number of equally sized point scatterers. In accordance with the results of classical radar analysis, the amplitude statistics of the return signal from **clutter only** are assumed to be Rayleigh distributed. Though this is the classical model used at microwave frequencies, it should be used with caution in the MMW band. Due to the shorter wavelengths of the MMW band, individual point scatterers sometimes appear to have a larger radar cross section (RCS). This manifests itself as a longer "tail" in the clutter **p.d.f.** leading to the assumption of a lognormal **p.d.f.** for the higher frequency portion of the MMW band, **35** GHz, modeling the clutter **p.d.f.** according to a Rayleigh distribution is still a valid assumption since data gathered at this frequency and presented in [4] appears to be closely approximated by a Rayleigh distribution.

The **Marcum** and Swerling models have been widely used for modeling fluctuating targets. However, due to the shorter wavelengths of the MMW band, the target RCS will be very sensitive to the viewing aspect angle of the radar. This will cause the target RCS to fluctuate rapidly with time. Due to this sensitivity to the viewing aspect angle, the **Marcum** and Swerling models are usually inadequate for the **MMW** band. Experimental results in the MMW band indicate that ground vehicles exhibit a lognormal probability distribution of RCS values when all viewing aspect angles are considered [4][6]. Therefore, we chose to model the cell-to-cell amplitude statistics of the return signal from target only according to a lognormal probability distribution.

The occurrence of targets in range were modeled using a marked point process [9]. At every occurrence of a target, a group of four random variables drawn from a lognormal distribution were inserted into the array representing the return signal. Therefore, the array representing the return signal consists of Rayleigh distributed random variables which represent the return signal from clutter only and groups of random variables drawn from a lognormal distribution which represent the return signal from target only. Since the target return is assumed to be dominant in the presence of clutter, no intermixing of the target and clutter returns within an individual range cell was done. Adding the clutter return to the target return within an individual range cell would complicate the analysis while adding little accuracy to the return signal model.

A portion of the return signal array at different points in its construction is shown below. Note that the targets were modeled to occur at a $P[H_1] = 0.248$ rate with a mean value of **10.0** and a variance of **25.25**. The background clutter process was modeled with a mean value of **4.4** and a variance of **5.50**. These values give a SCR of **5.0** (6.98 dB).



Figure 3.4 Target only return signal.



Figure 3.5 Clutter only return signal.



Figure 3.6 Target plus clurrer return signal.

Therefore, the complete data vector can be expressed as

$$\mathbf{Y}_{\mathbf{k}} = (\mathbf{x}_{\mathbf{k}}, \mathbf{U}_{\mathbf{k}}), \qquad (3.45)$$

where,

$$U_k = (U_k^0, U_k^1)$$
 = Desired Output Vector = { (0,1) Target Present
(1,0) Target Absent

 $\mathbf{x}_{k} | [\mathbf{U}_{k} = (0,1)] =$ Sample vector the components of which were drawn from a lognormal distribution.

 $\mathbf{x}_{\mathbf{k}} | [\mathbf{U}_{\mathbf{k}} = (1,0)] =$ Sample vector the components of which were drawn from a Rayleigh distribution.

Sample vectors were formed from this data set by including the range sample immediately before and after the present range sample. Thus, each sample vector, $\mathbf{x}_{\mathbf{k}}$, was formed as

$$\mathbf{x}_{k} = \{ s_{k-1}, s_{k}, s_{k+1} \}, \qquad (3.46)$$

where $\mathbf{x}_{\mathbf{k}}$ represents the \mathbf{k}^{th} sample vector drawn from the ensemble of all input data vectors, χ , and $\mathbf{s}_{\mathbf{k}}$ represents the \mathbf{k}^{th} range sample which is a random variable drawn either from a Rayliegh distribution, representing the return signal from clutter only, or a lognormal distribution which represents the return signal from target only.

As discussed before, the desired output vector which the neural network trains towards is denoted as $U_k = \{ U_k^0, U_k^1 \}$ where U_k^0 equals 1 when the target is absent, 0 otherwise, and U_k^1 equals 1 when a target is present, and 0 otherwise. This fixes the number of output nodes of the neural network to be equal to two. The training data set was formed

by employing a majority voting scheme imposed on the range samples making up a sample vector \mathbf{x}_k . If the majority of the range samples, \mathbf{s}_k , making up some particular sample vector, \mathbf{x}_k , belong to the class associated with a target present, then $U_k = \{0,1\}$. Otherwise, $U_k = \{1,0\}$, signifying that a target is absent.

CHAPTER 4

RESULTS

4.1 Introduction

This chapter begins with Section 4.2 entitled, "Computer Simulations," which discusses the computer simulations used to illustrate the performance of the LRWF as well as the overall performance of the neural network to correctly detect targets embedded in the background clutter process. Presented in Section 4.3, "Performance Testing of the LRWF," are the results used to illustrate the success of the LRWF to force the neural network to detect targets occurring with probability **P**[•][**H**₁], probability of target occurrence associated with the testing data set, rather than **P**[**H**₁], which is the probability of target occurrence associated with the training data set. Lastly, Section 4.4, "Network Performance in a Varying SCR Environment," contains simulation results used to illustrate the performance of the neural network trained with and without the LRWF to detect targets embedded in a varying SCR environment.

4.2 Computer Simulations

The data sets used for training and testing of the neural network were constructed according to the descriptions given in Section 3.7. Accordingly, the clutter only signal was modeled as a sequence of independent random variables drawn from a Rayleigh distribution with a variance of 5.50. Likewise, the target only signal was modeled as a sequence of independent random variables drawn from a log-normal distribution with a mean value of 10.0 and a variance of 25.0.

The training of the neural network was **carried** out using the backpropagation algorithm modified as discussed in Section 3.2. Training was continued until the mean square error (MSE) between the desired and the actual outputs decreased to some sufficiently low value. Testing was done by fixing the weights of the neural network and using the network to classify similarly modeled data, similar in terms of the **PDFs** used to represent the target only signal and the clutter only signal. Classification of the individual feature vectors, \mathbf{x}_i , was done by allowing the feature vector \mathbf{x}_i to propagate through the network until an output is developed at the two nodes of the final output layer. **As** previously shown in Section 3.2, the output of these two nodes represent estimates of the *a posteriori* probabilities $P[H_0|\mathbf{x}]$ and $P[H_1|\mathbf{x}]$ for nodes 0 and 1, respectively. A ratio test was then constructed as shown by Eq. (3.44) and repeated below for the reader's convenience,

$$\{ x : f_{x|H_{1}}(x) P[H_{1}] = \frac{F_{10}}{L_{01}} \}$$

$$(4.1)$$

where the ratio $\frac{L_{10}}{L_{01}}$ was set equal to 1.0. The detection rate (P_D) was determined as the percentage of target range cells which were classified as containing target returns. Similarly, the false alarm rate (FAR) was determined as the percentage of the total number of range cells falsely declared as containing target returns.

In order to obtain some degree of confidence in the values of P_D and the FAR, the testing simulations were allowed to continue until the values of P_D and FAR were known to three significant figures. This was done by simulating a vast number of radar pulses and

using the neural network to classify the individual range cells until the following condition was satisfied for both P_D and FAR [7],

$$\frac{1}{N}\sigma_{\mathbf{x}}^{2} < \mathbf{E}[\mathbf{X}]^{2}\varepsilon^{2}$$
(4.2)

where,

N = Total number of pulses simulated thus far σ_x^2 = Variance of either P_D or FAR E[X²] = Mean squared value of either P_D or FAR ε = Confidence interval

Since three significant figures were desired in the values of P_D and FAR, ϵ was set equal to **0.001**.

In the beginning, many different neural network architectures were investigated, ranging in complexity from single hidden layer networks containing as few as 47 weights to as many as **302.** However, the experiments with single hidden layer networks were unsuccessful in that reasonably low values of P_D and FAR were unable to be achieved, with or without the use of the **LRWF**, for events occurring with low probability ($P[Y_0] \leq$ **0.03**). However, by using networks with 2 hidden layers, where the number of weights ranges between 65 and 317, reasonably stable results for P_D and FAR were obtained.

The experimantal results which follow were obtained using a network with 3 nodes in the first layer, 12 nodes in the second and third layers, and 2 nodes in the final layer output layer. This network was chosen since we found little improvement in the values of P_D and FAR for networks of higher complexity.

4.3 Performance Testing of the LRWF

Presented in this section are the simulation results contrasting the performance of a conventionally trained neural network (i.e., no LRWF incorporated in the training algorithm) to a neural which incorporated the LRWF into its training algorithm. Table 4.1 illustrates the failure of the conventionally trained neural network to correctly classify targets occurring with low probability when the network was trained using a data set for which $P[H_1] = 0.248$. The total number of feature vectors making up the training set was set equal to 2000.

Table 4.2 presents results which illustrate the performance of a conventionally trained network for which $P^{\bullet}[H_1] = P[H_1]$. Note that during the training of each network the total number of range cells representing the target signature was held constant at 496. In order to do this, the total number of feature vectors making up the training set ranged from 2000 to 31,000 so as to generate the desired value of $P[H_1]$. Note that we were unable to train the network to recognize targets for which the probability of target occurrence was less than 0.016.

Table 4.3 illustrates the performance of the network which results from incorporating the LRWF into the backpropagation algorithm. The training distribution was one for which $P[H_1] = 0.248$ and the LRWF was used to perform the mapping to a space with a low value of $P^{\bullet}[H_1]$. The initial training distribution was made up of 2000 feature vectors. Note that these results were obtained while attempting to maintain a constant value of P_{D} .

Lastly, Table 4.4 illustrates the performance of the network which results from incorporating the LRWF into the backpropagation algorithm while attempting to maintain a constant value of FAR. As before in Table 4.3, the initial training distribution is made up of 2000 feature vectors with a value of $P[H_1] = 0.248$.

P[H ₁]	P*[H ₁]	P _D	FAR	Training Information
0.248	0.248	0.766	0.079	η = 0.01 MSE = 0.0789
0.248	0.124	0.844	0.140	η = 0.01 MSE = 0.0789
0.248	0.048	0.921	0.264	η = 0.01 MSE = 0.0789
0.248	0.024	0.948	0.375	η = 0.01 MSE = 0.0789
0.248	0.016	0.967	0.453	η = 0.01 MSE = 0.0789
0.248	0.012	0.971	0.490	η = 0.01 MSE = 0.0789
0.248	0.010	0.981	0.544	η = 0.01 MSE = 0.0789

0.981

η = 0.01 MSE = 0.0789

0.566

0.248

0.008

Table 4.1 Failure of conventionally trained network for the case $P^{\bullet}[H_1] \ll P[H_1]$.

	Dorformonoo of	conventially t	rained network for	the enco D	'IH.1_1	pru.i
1 able 4.2	Performance of	conventially t	rained network for	me case P	$[\mathbf{n}_1] = 1$	г[п ₁].

P[H ₁]	P [*] [H ₁]	P _D	FAR	Trifaining ion
0.248	0.248	0.766	0.079	η = 0.01 MSE = 0.0789
0.120	0.120	0.740	0.075	η = 0.01 MSE = 0.0269
0.048	0.048	0.737	0.076	η = 0.01 MSE = 0.0468
0.024	0.024	0.724	0.079	η = 0.01 MSE = 0.0239
0.016	0.016	0.797	0.122	$\eta = 0.01$ MSE = 0.0171

P[H ₁]	P*[H ₁]	բ*[ӊ ₀]/Ҏ[ӊ ₀]	р*[H ₁]/Р[H ₁]	P _D	FAR	Training Information
0.248	0.248	1.000	1.000	0.766	0.079	η = 0.01 MSE = 0.0789
0.248	0.124	1.165	0.500	0.767	0.091	η = 0.01 MSE = 0.0905
0.248	0.048	1.264	0.193	0.761	0.096	$\eta = 0.01$ MSE = 0.1172
0.248	0.024	1.298	0.0968	0.762	0.115	$\eta = 0.01$ MSE = 0.1460
0.248	0.016	1.308	0.0645	0.770	0.115	$\eta = 0.01$ MSE = 0.1596
0.248	0.012	1.314	0.0484	0.761	0.115	η = 0.01 MSE = 0.1709
0.248	0.010	1.316	0.0403	0.756	0.114	$\eta = 0.01$ MSE = 0.1704
0.248	0.008	1.319	0.0323	0.751	0.112	$\eta = 0.01$ MSE = 0.1823

Table 4.3 Network performance with LRWF while attempting to maintain a constant value of P_D .

Т

P[H1]	P*[H ₁]	₽ [*] [H ₀]/₱[H ₀]	₽ ^ŧ [H ₁]/₽[H ₁]	P _D	FAR	Training Information
0.248	0.248	1.000	1.000	0.766	0.079	$\eta = 0.01$ MSE = 0.0789
0.248	0.124	1.165	0.500	0.753	0.080	η = 0.01 MSE = 0.0867
0.248	0.048	1.264	0.193	0.737	0.076	$\eta = 0.01$ MSE = 0.1258
0.248	0.024	1.298	0.0968	0.715	0.081	η = 0.01 MSE = 0.1556
0.248	0.016	1.308	0.0645	0.705	0.080	η ⁼ 0.01 MSE = 0.1668
0.248	0.012	1.314	0.0484	0.703	0.082	η = 0.01 MSE = 0.1789
0.248	0.010	1.316	0.0403	0.695	0.079	η = 0.01 MSE = 0.1876
0.248	0.008	1.319	0.0323	0.688	0.078	η = 0.01 MSE = 0.1849

 Table 4.4 Network performance with LRWF while attempting to maintain a constant value of FAR.



Figure 4.1 Contrasting performance in terms of the P_D values presented in Tables 4.1 thru 4.4.



Figure 4.2 Contrasting performance in terms of the FAR values presented in Tables 4.1 thru 4.4.

4.4 Network Performance in a Varying SCR Environment

The results presented in this section illustrate the classification performance of the network with and without the LRWF in a varying SCR environment. For each case, the training distribution was constructed in such a manner as to a SCR equal to 5.0 (6.98 dB). This value of SCR the assumptions made in Section 3.5 concerning the simplification of the LRWF, g(x), are valid. Values of SCR below below 5.0 were found to give unsatisfactory performance in both the training and testing phases. The training data set was also constructed to produce have the probability of target occurance equal to $P[H_1] = 0.248$. As before in Section 4.3, the network itself was structured to have 3 nodes in the first layer, 12 nodes in the second and third layers, and 2 nodes in the fourth layer.

The results presented in Table 4.5 illustrate the performance of the convetionally trained neural network in a varying SCR environment for the case $P^{*}[H_{1}] = P[H_{1}]$.

SCR (dB)	P _D	FAR
0	0.800	0.572
3	0.780	0.350
6	0.756	0.118
9	0.737	0.022
12	0.717	0.006
15	0.705	0.003
18	0.696	0.002
21	0.689	0.001
24	0.683	0.001
27	0.680	0.001
30	0.677	0.001

 Table 4.5
 Performance of convetionally trained neural network in a varying SCR envirionment.

The results presented in Table 4.6 illustrate the performance of the neural network in varying SCR environment while incorporating the LRWF to perform a mapping from the training distribution for which $P[H_1] = 0.248$ to the testing distribution for which $P^*[H_1] = 0.008$.

P [H ₁]	P*[H ₁]	SCR (dB)	PD	FAR
0.248	0.008	0	0.327	0.356
0.248	0.008	3	0.636	0.307
0.248	0.008	6	0.741	0.163
0.248	0.008	9	0.733	0.035
0.248	0.008	12	0.719	0.002
0.248	0.008	15	0.712	0.0001
0.248	0.008	18	0.706	0.000 ¹
0.248	0.008	21	0.707	0.000 ²
0.248	0.008	24	0.703	0.000^{2}
0.248	0.008	27	0.701	0.000 ²
0.248	0.008	30	0.700	0.000 ²

Table 4.6Performance of the neural network in a varying SCR environment
while incorporating the LRWF.

¹ For these values of SCR the FAR was less than 0.001

.....

² For these values of SCR the FAR was equal to zero, i.e., no false alrams occured during the entire simulation.



Figure 4.3 Contrasting conventional and **LRWF** performance in terms of P_D over a 30 dB range of SCR.

ļ

CHAPTER 5

CONCLUSIONS AND RECOMENDATIONS

5.1 Discussion

The results presented in Table 4.1 were obtained by training the network on a data set where the probability of target occurrence, $P[H_1]$, was set equal to 0.248. This same network was then used to classify an identically modeled data set where the probabilities of target occurrence, $P^e[H_1]$, are those listed in Table 4.1. For the classification scheme described by Eq. (3.44), these results indicate that, as the value of $P^*[H_1]$ decreases, the values of P_D and FAR increase. This result is to be expected by considering the likelihood ratio given by Eq. (3.44) and repeated below as Eq. (5.1).

$$\{ x : \frac{f_{x|H_{1}}(x) P[H_{1}]}{f_{x|H_{0}}(x) P[H_{0}]} > \frac{L_{10}}{L_{01}} \}$$
(5.1)

From Eq. (5.1) it is easily seen that when the network is used to classify targets occurring with probabilities $P^{\bullet}[H_1]$ given in Table 4.1, the likelihood ratio will become positively biased. This is due to the network expecting to classify data for which the probability of target occurrence is equal to $P[H_1]$ when in fact the probability of target occurrence is equal to $P^{\bullet}[H_1]$, where $P^{\bullet}[H_1]$ are $P^{\bullet}[H_1]$. This positive bias is reflected in Table 4.1 by

The results presented in Table 4.2 were obtained by training and testing the neural network utilizing data sets with a low probability of target occurrence. In order to correctly contrast these results to those presented in Tables 4.3 and 4.4, the training set was expanded to consist of many simulated radar pulses so as to keep the total number of range cells containing target returns constant at 496 which equals the total number of range cells containing target returns for the case $P^{\bullet}[H_1] = P[H_1] = 0.248$. These results were obtained so as to provide a benchmark of performance against which the results obtained utilizing the (LRWF) could be compared. However, we were unable to successfully train the network to recognize events occurring at rates below 0.016 using this method. Note the increase in memory requirements as the value of $P^{\bullet}[H_1]$ is decreased from 0.248 to 0.016 which corresponds to an expansion in the size of the training data set from 2000 to 31,000 input vectors.

The results presented in Tables 4.3 and 4.4 illustrate the performance of the network when trained utilizing the LRWF. The results presented in Table 4.3 represent an attempt to maintain a constant value of P_D equal to that obtained for the case $P[H_1] = P^{\bullet}[H_1]$ while noting the resulting values of FAR. Similarly, the results presented in Table 4.4 represent our attempt to maintain a constant value of FAR equal to that obtained for the case $P[H_1] = P^{\bullet}[H_1]$ while noting the resulting values of P_D . Ideally, we desire to be able to train the network utilizing the LRWF so as to be able to classify data with a much lower value of $P^{\bullet}[H_1]$ while still preserving the performance of the baseline case $P[H_1] = P^{\bullet}[H_1]$ in terms of the values for both P_D and FAR. By examining the results presented in' Tables 4.3 and 4.4, one can see that we were able to accomplish our goal only partially in that we were able to achieve the desired value of P_D at the expense of an increase in the value of FAR. Similarly, we were able to achieve the desired value of FAR at the expense

of a reduction in the value of P_D . Therefore, a **tradeoff** situation exists between the values of P_D and FAR. If training was allowed to continue in search of the minimum achievable value of **MSE** between the desired and the actual outputs, the value of P_D will be comparable to the baseline case of $P[H_1] = P^{\bullet}[H_1]$ at the expense of an increase in the value of FAR. However, if training of the neural network was suspended once the MSE between the desired and the actual outputs ceases to change by an appreciable amount, the weights matrices which result would be to maintain a value of FAR comparable to the baseline case of $P[H_1] = P^{\bullet}[H_1]$.

When contrasting the results presented in Tables 4.1 through 4.4, it is interesting to note the close similarity between the results presented in Tables 4.2 and 4.4 as well as their associated plots in Figures 4.1 and 4.2. These results show that by including the LRWF in the backpropagation algorithm it is possible to achieve relatively the same performance as the conventionally trained network with far less training time due to the reduced size of the training set. Also, it was mentioned that we were unable to successfully train the conventionally trained network to recognize targets occurring at rates below 0.016. This is due to the reduced number of target samples as compared to the size of the entire data set. Therefore, as illustrated in Tables 4.3 and 4.4 the LRWF offers the means to train a neural network to recognize targets which occur with low probability by keeping the ratio of targets samples to clutter samples relatively independent of the desired value of $P^{\bullet}[H_1]$.

Upon comparing the curves labeled Table2_Pd' and 'Table4_Pd' in Figure 4.1, we see that we actually have superior performance in terms of P_D for mapping ratios in the range zero to five. This corresponds directly to the performance predicted by the curves presented in Figure 3.3.

The results pertaining to network performance in a varying SCR environment presented in Section 4.4 seem to indicate that the network trained utilizing the LRWF produces higher values of P_D and lower values of FAR as compared to the conventionally trained network over most of the **30** dB SCR range for which results were obtained. However, examination of Figure **4.3** reveals disturbing results obtained from the LRWF trained network for SCRs less than 6.98 dB, the value of SCR for the training data set. Recall that in order to simplify the evaluation of the LRWF it was assumed that little overlap existed between the **PDFs** of the target and clutter distributions. As the SCR is decreased, this assumption is no longer valid explaining the degradation in network **performance** for decreasing SCR.

For the conventionally trained network, we see that the probability of detection increases with decreasing SCR. By refemng to Figure 4.4 it can be seen that this increase in P_D is accompanied by a substantial increase in the FAR. Therefore, as the noise level is increased the conventionally trained network is unable to differentiate between the target and clutter distributions. For the network trained utilizing the LRWF some degree of discrimination between target and clutter distributions is preserved, though a marked increase in FAR still occurs.

For increasing values of SCR it is seen that the LRWF trained network produces zero false alarms during simulations for SCRs in the range **21** dB to **30** dB while the conventionally trained network produced a FAR equal to **0.001** over this same range of SCR. Note, however, that these results were obtained to three significant figures only. In order to properly evaluate values of FAR less than **0.001** more exhaustive simulations would be required.

5.2 Conclusions

The results presented in Section **4.3** clearly indicate the value of LRWF as a tool capable of significantly reducing the training time of a neural network to detect targets (or events) occurring with low probability. The capability of a neural network to perform the task of stationary target discrimination is also evident in Figures **4.3** and **4.4**. The network is capable of maintaining a reasonably high detection rate and a relatively low FAR over a wide SCR environment.

CHAPTER 6 FUTURE RESEARCH

6.1 Discussion

Our future research will be primarily directed towards generalizing the LRWF to an N class pattern recognition problem with each class occurring at an equally likely rate. **As** the number of classes, N, increases the probability of some **i**th class occurring decreases. This, of course, increases the size of the training data set for the neural network. We hope to show that by incorporating the LRWF into the backpropagation algorithm for a general N class problem, we will be able to increase the ability of the neural network to correctly classify each of the N classes while at the same time reduce the computational overhead of the entire training procedure.

Specifically, we plan to first generalize the proof presented in Section 3.3 of this thesis to that of an N class problem. Modification of the backpropagation algorithm to incorporate the LRWF will remain the same. Next, we plan to simulate the performance of the generalized LRWF (GLRWF) by first generating a sixteen class data set, each class being distributed according to a Gaussian distribution with a distinct mean and variance. **A** neural network with an appropriately chosen architecture will then be trained to classify this data set using our modified version of the backpropagation algorithm. Once the neural network has been trained, it will be tested on a similarly generated data set in the
manner described in Section 4.2 of this thesis. These results will then be compared to those obtained from a similarly structured neural network whose training algorithm does not incorporate the **GLRWF**. These results will be compared in terms of classification performance and required CPU time for training. Note that two figures will be obtained for the classification performance and the required CPU time for training. The first will be for the case where the training data set will be identical to that used for the network which incorporates the **GLRWF**. The second will be for the case where the training data set will be for the case where the training data set used for the case where the training data utilizing the **GLRWF**.

LIST OF REFERENCES

- 1. D. E. **Rumelhart,** J. L. **McClelland,** and the PDP Research Group, Parallel Distributed Processing, Vol. 1: Foundations. M.I.T. Press, Cambridge, MA., 1986.
- 2. M. C. Jeruchim, P. **Balaban**, and K. Sam **Shanmugan**, Simulation of CommunicationsSystems, Plenum, New York, 1992.
- 3. D. W. Ruck, S. K. Rogers, M. **Kabrisky**, M. E. Oxley, and B. W. Suter, "The **multilayer** perceptron as an approximation to a bayes optimal discriminant **function**," IEEE Transactions on Neural Networks, Vol. 1, No. 4, December 1990.
- 4. N. C. Currie, *Radar Reflectivity* Measurement: Techniques and Applications, Artech House, 1989.
- 5. G. Minkler and J. **Minkler**, The Principles *of Automatic* Radar Detection CFAR, Magellan Book Company, 1990.
- 6. C. E. Brown and N. C. Currie, Principles *and Applications* of Millimeter Wave Radar, Artech House, 1989.
- 7. P. W. **Glynn** and W. Whitt, "The asymptotic validity of sequential stopping rules for stochastic simulation," Annual of Applied Probability, Vol. 2, pp. 180-198, 1992.
- 8. L. L. Scharf, Statistical Signal Processing, Addison-Wesley Publishing Company, 1990.
- 9. D. L. Suyder, M. I. Miller, Random *Point Processes* in Time and Space, New York, New York, Singer-Verlag, 1991.