Naval Research Laboratory

Washington, DC 20375-5000

NRL Memorandum Report 6739

AD-A229 038

Parametric and Non-Parametric Schemes for Discrete Time Signal Discrimination

JOSEPH A. HAIMERL

Advanced Techniques Branch
Tactical Electronic Warfare Division

November 13, 1990

DTIC
ELECTE
NOV 21 1990
S B D

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 1990 November 13	3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE
Parametric and Non-Parametric Schemes for Discrete Time Signal Discrimination

5. FUNDING NUMBERS
C- N00014-88-J-2003
PE- 63270N
PR-R2030E00T0
WU- DN156-065

6. AUTHOR(S)
Joseph Albert Haimerl

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Maryland
College Park, MD 20740

8. PERFORMING ORGANIZATION REPORT NUMBER
NRL Memorandum
Report 6739

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Technology
Arlington, VA 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

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

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

> In this thesis parametric and non-parametric schemes for discrete time signal discrimination are considered. Discrete time signal discrimination is the problem of classifying a random discrete time signal into one of two classes. The term *discrimination* arises from the more specific problem where the two classes are a target of interest and a decoy target.

In this thesis we consider both parametric and non-parametric schemes for discriminating between the two classes. In Chapter 2, we assume that first and second order probability density functions (pdfs) of the data under each class are known. Using these pdfs optimal memoryless quantizer discriminators are constructed. In Chapter 3, it is assumed that the pdfs are not known. Utilizing kernel density estimators and sample data from each class, estimates of the pdfs are formed for each class. Optimal memoryless quantizer discriminators are then constructed using the estimated pdfs and the expressions from Chapter 2. In Chapter 4, a perceptron neural network is trained with a supervised learning algorithm using sample data from each class. The perceptron neural network is utilized by a discriminator which uses memory.

Results for simulated radar data are presented for all schemes. Results show that the neural network discrimination scheme performs significantly better than the memoryless quantization schemes.

14. SUBJECT TERMS		15. NUMBER OF PAGES
Detection	Neural Networks	125
Optimal Discrimination	Kernel Density Estimation	16. PRICE CODE

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

CONTENTS

PARAMETRIC AND NON-PARAMETRIC SCHEMES FOR DISCRETE TIME SIGNAL DISCRIMINATION

Chapter 1

Introduction

In this thesis we consider the problem of discriminating between classes of discrete time signals. The simplest case of discrete time signal discrimination is the binary discrimination problem. In this case, a random discrete time signal is observed and must be classified into one of two categories. Typically the discrimination method is designed to optimize some measure of performance; this measure of performance is usually related to the probability of error and/or the number of samples used to make a decision. The binary discrimination problem is faced often in radar applications, where the receiver must decide whether the observed signal is from a target of interest or a decoy. Throughout this thesis, we present results on signal discrimination for arbitrary classes of data without assuming what the data represent or from what structure/implementation they are obtained. However, we shall often try to relate our results to the problem of binary discrimination faced by a radar receiver.

We refer to the two classes of signals from which the observed data originate as hypotheses H_1 and H_0. The observed data sequence is denoted as $\{Z_i\}_{i=1}^n$. Under hypothesis H_i, $i = 0, 1$, (i.e. hypothesis H_i is true,) the observed data sequence has the n-dimensional

---

probability density function (pdf) $f_i(z_1, z_2, \ldots, z_n)$. More specifically, we consider

$$H_1 : \{Z\}_{i=1}^n \text{ has pdf } f_1(z_1, z_2, \ldots, z_n) = f_1(\mathbf{z})$$

$$H_0 : \{Z\}_{i=1}^n \text{ has pdf } f_0(z_1, z_2, \ldots, z_n) = f_0(\mathbf{z}) \tag{1.1}$$

where $\mathbf{z}$ represents the n-tuple $(z_1, z_2, \ldots, z_n)$. Note that we do not constrain the data to be independent; various assumptions of the correlation between samples will be made later in this thesis.

If the n-dimensional pdfs under each hypothesis were known by the discriminator designer, a likelihood ratio test could be implemented. The likelihood ratio test is of the form

$$d(\mathbf{z}) = \begin{cases} 1, & \text{if } \dfrac{f_1(\mathbf{z})}{f_0(\mathbf{z})} \geq \eta \\ 0, & \text{if } \dfrac{f_1(\mathbf{z})}{f_0(\mathbf{z})} < \eta \end{cases} \tag{1.2}$$

where η is a constant to be determined. Hypothesis H_i is chosen by the discriminator when $d(\mathbf{z}) = i$, $i = 0, 1$. Likelihood ratio tests are well known and optimal in the Bayes, Neyman-Pearson, and minimax senses[1]; the choice of η depends upon which criterion the designer chooses to optimize. However, we assume that the n-dimensional pdfs are **not** known.

It is further assumed that the data sequence is strictly stationary that is, the statistics do not vary with time:

$$f_i(z_1, z_2, \ldots, z_n) = f_i(z_{k+1}, z_{k+2}, \ldots, z_{k+n}) \quad i = 0, 1; \quad k \text{ arbitrary.} \tag{1.3}$$

As mentioned above, we do not constrain the data to represent any specific signal. However, for the radar problem, some possibilities are samples of the envelope detector output, matched filter output, or even phase data. Figure 1.1 illustrates a scheme for discriminating between radar targets by using envelope samples.

The radar uses a simple pulse modulated waveform. The pulse modulator block generates the pulsed waveform. This in turn is fed into the transmitter to be modulated to radio
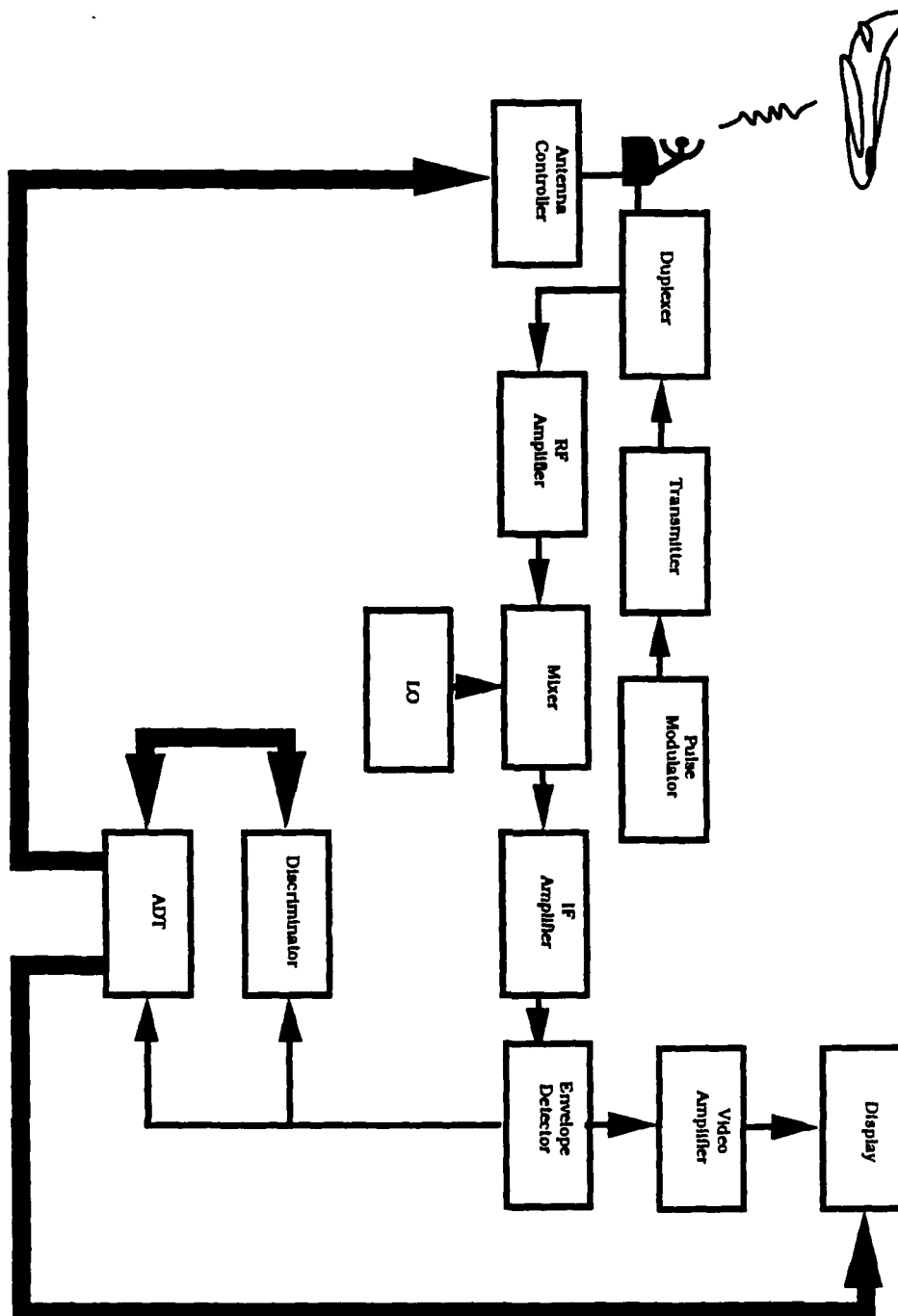
2

Figure 1.1: A Tracking Radar System Employing a Discriminator

frequency (RF). This signal is then input to the duplexer, which isolates the transmitter and receiver during transmission and reception. During the transmission, the receiver is effectively *disconnected* from the antenna, while during reception the transmitter is disconnected from the antenna. The pulsed radio frequency signal is then radiated through the antenna. If the antenna is pointing at an object, some portion of the signal may be reflected towards the antenna. By this time, the duplexer has switched the antenna to the receiver circuitry. The incoming waveform is amplified by a radio frequency amplifier and then mixed to an intermediate frequency (IF). This signal is then passed through the matched filter of the IF amplifier to maximize the signal to noise ratio (SNR). The output of this block is then envelope detected. A portion of the envelope signal is routed through the video amplifier and into a display: either an A-scope or a PPI (plan position indicator.)

The other portions of the envelope signal are routed to the ADT (automatic detection and tracking) circuitry and to the discriminator circuitry. The ADT determines if targets are present, initiates track on new targets, and determines how to set the pointing angles of the antenna. The ADT therefore communicates with the display circuitry and the antenna control circiutry. The ADT also notifies the discriminator circiutry that a target has been detected. The discriminator then begins its tests by obtaining samples of the envelope signal in the time intervals corresponding to the target's position. When the discriminator makes a decision, it can instruct the ADT to continue tracking the target (if it is a target of interest) or to drop the target from track (if it is a decoy or a target of little interest.) Figure 1.2 illustrates a method that a discriminator may possibly utilize in obtaining data samples. The figure is a diagram of five pulse repetition intervals (PRIs.) The rectangular pulses represent the pulse waveform to be modulated and transmitted. The random signal between the pulses represents the envelope signal. The data samples $Z_0, Z_1, Z_2, \ldots$ are obtained by sampling

4
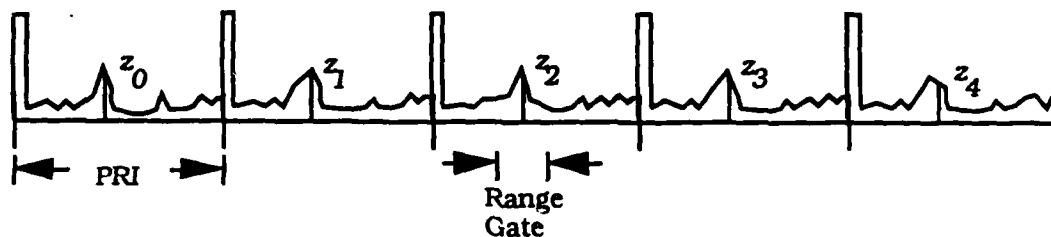
Figure 1.2: Extracting Data Samples from the Radar Return

the envelope signal within the range gate corresponding to the object being discriminated. In Figure 1.2, only one sample per target per range bin is obtained.

The above implementation is just one example of how a discriminator can be implemented in a practical system. However, structure of the discriminator block was not detailed in the above example. There are several approaches to designing the discriminator block. Figure 1.3 illustrates some possible approaches to designing a discriminator. The first approach is to model the physics generating the data under each hypothesis. Then the pdfs of the data under each hypothesis may be assumed or derived, thus allowing a discriminator to be implemented. Another approach is to collect actual data, estimate pdfs of the data under each hypothesis, and then implement a discriminator. The last method is to collect data, train a discriminator with a supervised learning algorithm via simulation, and then implement the discriminator. The first approach may be very difficult and mathematically intractable. The other two approaches are more easily adaptable to any problem since they assume no model of the physics which generate the data sequence.

In this thesis, we consider all three of the above approaches to designing a discriminator. In Chapter 2, it is assumed that marginal and bivariate pdfs of the data under each hypothesis are known to the discriminator designer. Optimal memoryless quantizer discriminators are designed using the marginal and bivariate pdfs (actually cumulative dis-

5
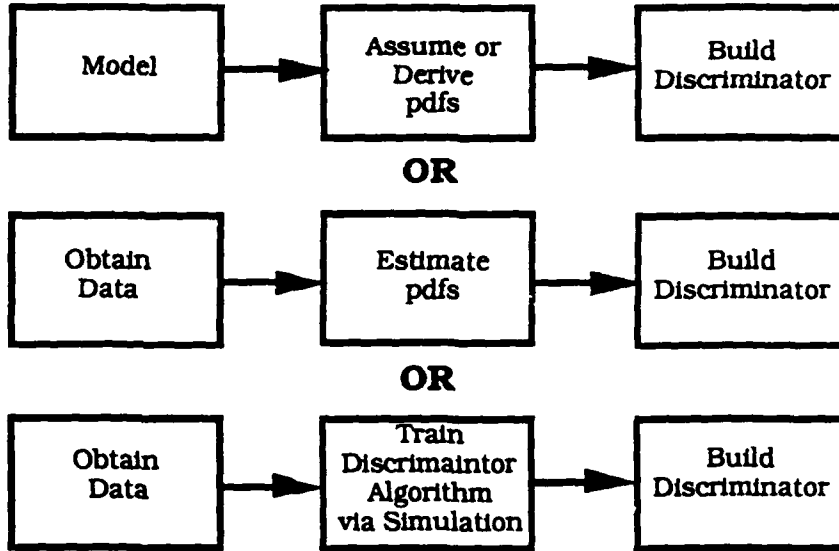
Figure 1.3: Possible Approaches to Designing a Discriminator

tribution functions, denoted as cdfs). The discriminators use a test statistic of the form $T_j = \sum_{i=1}^{j} Q(Z_i)$, where $Q(x)$ is a quantization function chosen to maximize a suitable performance measure. The approach used to design the discriminator corresponds to the first approach of Figure 1.3 and is parametric since pdfs are assumed unavailable.

In Chapter 3 it is assumed that the pdfs are not known. The approach used to design a discriminator in this chapter corresponds to the second approach in Figure 1.3. Non-parametric estimates of the marginal and bivariate pdfs of the data under each hypothesis are formed and fed into the expressions for the optimal memoryless quantizer discriminators derived in Chapter 2. The estimates are formed by collecting data prior to the design of the discriminator; the data are fed into kernel density estimators. The data for estimation are denoted as

$$\zeta_{m,j}^{i} \tag{1.4}$$

where $i = 0, 1$ denotes hypothesis H_0 or H_1 respectively, where $m = 0, 1, \ldots, M - 1$ denotes

6

the sample path number, and where $j = 0, 1, 2, \ldots, N-1$ denotes the sample number. Thus under each hypothesis, we have M sample paths (i.e. sequences) which are N data samples long. It is assumed that the M sample paths are independent of each other. Throughout this thesis we refer to the these data sequences as the *training data*.

In Chapter 4, the final approach to designing a discriminator is considered. The discriminators from Chapters 2 and 3 required only marginal and bivariate pdfs due to their memoryless property. However, it is suspected that memory improves performance for correlated data. In Chapter 4, discriminators which have a test statistic of the form $T_j = \sum_{i=K}^{j} \gamma(Z_{K-1+i}, Z_{k-2+i}, \ldots, Z_i)$ are considered. To find the optimal nonlinearity $\gamma(\cdot)$, pdfs of higher order than the bivariate pdfs would have to be assumed or estimated; the estimation of the higher-order pdfs may not be practical and the assumption or derivation of such pdfs may be mathematically intractable. To avoid the difficulty in obtaining these pdfs, multiple-layer perceptron *neural networks* are trained to act as the nonlinearity $\gamma(x_1, x_2, \ldots, x_K)$ using the back propagation algorithm.

It is likely that once a discriminator is implemented, it will encounter data from pdfs different from those with which it was designed. Obviously, the designer wants the discriminator to be robust to these conditions. In Chapter 5, some simulation results are presented on the mismatch of the pdfs. These results give some indication of the robustness characteristics of the discriminators presented in this thesis.

Note that all discriminator models in this thesis make decisions on the basis of sequential tests. These tests, upon obtaining a new data sample, either classify the sequence or decide to obtain a new data sample. Sequential tests are used in situations requiring fast and accurate decisions.

Discriminators using similar structures to the ones in this thesis can be implemented to form decisions on the basis of fixed length blocks of data. However fixed sample size tests are beyond the scope of this thesis.

Chapter 2

Memoryless Quantizer Discriminators

In this chapter, we derive optimal memoryless quantizer discriminators for use in our binary discrimination problem. The data sequence is assumed stationary and m-dependent. m-dependent means that, under H_i, Z_k and Z_l are correlated for $|k - l| \leq m_i$ and are independent for $|k - l| > m_i$. These discriminators operate on the data sequence $\{Z_i\}_{i=1}^{\infty}$ by computing the test statistic $T_n = \sum_{i=1}^{n} Q(Z_i)$. $Q(x)$ is a quantizer function chosen to maximize a suitable performance measure.

The test may be performed in either a block or sequential fashion. Both tests are based on the fact that, as n tends towards infinity, T_n converges to a Gaussian distribution with mean $n\mu_i$ and variance $n\sigma_i^2$, for $i = 0, 1$, corresponding to hypotheses H_1 and H_0 respectively. μ_i is defined by

$$\mu_i(Q) = E_i\{Q(Z_1)\}, \quad i = 0, 1 \tag{2.1}$$

where E_i denotes expectation under hypothesis H_i. σ_i^2 is defined by

$$\sigma_i^2(Q) = \lim_{n \to \infty} n^{-1} \text{Var}_i(T_n), \quad i = 0, 1 \tag{2.2}$$

9

and is given by

$$\sigma_i^2(Q) = Var_i(Q(Z_1)) + 2\sum_{j=1}^{m_i} Cov_i\{Q(Z_1), Q(Z_{j+1})\}. \qquad (2.3)$$

Var_i and Cov_i denote variance and covariance under hypothesis H_i, and m_i is the m-dependence length under hypothesis H_i. [2] gives a proof using a central limit theorem which shows that T_n is asymptotically Gaussian under hypothesis H_i with mean $n\mu_i$ and variance $n\sigma_i^2$, provided that $\sigma_i^2 > 0$.

Optimal quantization has been studied by others (see [3] and [4]) for the related hypothesis testing problem concerning the detection of weak signals in additive noise. These employed block tests, where T_n was compared to a decision threshold. Quantizer functions were chosen to maximize the well known asymptotic relative efficiency (ARE.) Given two detectors, φ_1 and φ_2, the ARE of detector φ_1 relative to detector φ_2 is defined as

$$ARE(1,2) = \lim_{n\to\infty, \theta\to 0} e(\alpha, \theta, n), \qquad (2.4)$$

where $e(\alpha, \theta, n)$ is the relative number of samples φ_2 required to achieve the same probability of detection that φ_1 achieves for sample size n when both φ_1 and φ_2 have false alarm probability α and signal strength θ. Under certain regularity conditions (see [2].) the ARE for two quantizer detectors has the form

$$ARE(Q_1, Q_2) = \frac{\eta(Q_1)}{\eta(Q_2)}, \qquad (2.5)$$

where the quantity $\eta(Q)$ is the efficacy of the detector φ using Q, and is given by

$$\eta(Q) = \frac{(\int Qf')^2}{\sigma_0^2(Q)}. \qquad (2.6)$$

Here, f' is the derivative of the noise marginal probability density function with respect to signal strength θ. The optimal quantizer is the quantizer which optimizes the ARE: this quantizer also maximizes the efficacy.

10

[4] derived the optimal quantizer for the weak signal detection case with independent noise, while [2] derived the optimal quantizer for the m-dependent noise case. For the discrimination problem, [5] has derived optimal nonlinearities which maximize signal to noise type performance measures of the form

$$\bar{S}_4(g) = \frac{(\mu_1(g) - \mu_0(g))^2}{(\nu\sigma_1^2(g) + (1 - \nu)\sigma_0^2(g))}, \tag{2.7}$$

where $\nu \in [0, 1]$. These detectors operated in a block fashion forming a test statistic $T_n = \sum_{i=1}^n g(Z_i)$ which was compared to a decision threshold.

Recently, [6] derived optimal nonlinearities for use in a sequential discrimination scheme. These sequential discriminators operated by forming a test statistic of the form $T_n = \sum_{i=1}^n g(Z_i)$. Another test statistic was formed, either as a linear expression of T_n: $S_n = \bar{A}T_n + \bar{B}n$, or as a quadratic expression of T_n: $S_n = \bar{A}T_n^2 + \bar{B}T_n + \bar{C}n + D$. S_n was then compared to two decision thresholds: if the upper threshold, b, was exceeded, H_1 was declared. If S_n dropped below the lower threshold, a, H_0 was declared. Otherwise, another sample Z_{n+1} was obtained, T_{n+1} and S_{n+1} were computed, and the threshold tests were repeated. This continued until one of the thresholds was crossed. The nonlinearities were chosen to minimize the average sample size required to terminate the test. This criterion is important for the class of problems where a fast decision is needed as well as a reliable decision (i.e. small error probabilities.) [6] used the well known Wald thresholds [7] of $b = \ln\left((1 - \beta)/\alpha\right)$ and $a = \ln\left(\beta/(1 - \alpha)\right)$. The corresponding optimal values of $\bar{A}$ and $\bar{B}$ were $\bar{A} = 2(\mu_1 - \mu_0)/\left(\sigma_1^2 + \sigma_0^2\right)$ and $\bar{B} = -2(\mu_1 - \mu_0)\left(\mu_1\sigma_0^2 + \mu_0\sigma_1^2\right)/\left(\sigma_1^2 + \sigma_0^2\right)^2$.

[6] showed that the optimal nonlinearity solved a nonlinear integral equation; this was the result of a more complex performance measure than (2.7). However [6] also considered a suboptimal nonlinearity which solved a linear integral equation; this nonlinearity was

11

the result of a performance measure with the form of (2.7). The suboptimal nonlinearity performed nearly as well as the optimal nonlinearity and was much easier to solve for because of the linear integral equation. Since [5] and [6] have shown performance measures with the form of (2.7) which result in good block and sequential discriminators, we only consider quantizers that maximize a performance measure of the form given in (2.7). Being consistent with the subscript notation in [5], we state our prc · n as finding a quantizer that maximizes the performance measure

$$\tilde{S}_4(Q) = \frac{(\mu_1(Q) - \mu_0(Q))^2}{(\nu \sigma_1^2(Q) + (1 - \nu)\sigma_0^2(Q))}.$$

(2.8)

Now we define the notation used in this chapter. The quantizer function $Q(x)$ is defined as $Q \equiv (\mathbf{q}, \mathbf{t})$, where $\mathbf{q} = (q_1, q_2, \ldots, q_M)^T$ is the quantization level vector and $\mathbf{t} = (t_0, t_1, \ldots, t_M)$ is an ordered breakpoint vector. These define $Q(x)$ by

$$Q(x) = q_k \quad \text{when} \quad x \in (t_{k-1}, t_k], \quad k = 1, \ldots, M.$$

(2.9)

With this definition of $Q(x)$, we sometimes also use the notation $\tilde{S}_4(Q) = \tilde{S}_4(\mathbf{q})$.

2.1 Derivation the Mean and Variance for a Quantizer Function

To maximize the performance measure $\tilde{S}_4(\mathbf{q})$, the mean μ_i and variance σ_i^2 of the quantizer function must be evaluated under both hypotheses ($i = 0, 1$). The mean of the quantizer is

12

given by

$$\mu_i\left[Q(Z_1)\right] \triangleq E_i\left[Q(Z_1)\right] = \int_0^{+\infty} \sum_{k=1}^{M} q_k I_{(t_{k-1},t_k]}(z) f_i(z) dz$$

$$= \sum_{k=1}^{M} q_k \int_0^{+\infty} I_{(t_{k-1},t_k]}(z) f_i(z) dz$$

$$= \sum_{k=1}^{M} \int_{t_{k-1}}^{t_k} q_k f_i(z) dz \qquad (2.10)$$

$$= \sum_{k=1}^{M} q_k Pr_i\{z \in (t_{k-1}, t_k]\}$$

$$= \sum_{k=1}^{M} q_k \left[F_i(t_k) - F_i(t_{k-1})\right]$$

$$= \mathbf{q}^T(\mathbf{\Delta F}_i) = (\mathbf{\Delta F}_i)^T \mathbf{q}.$$

In the above expressions we used $I_A(x)$, the standard indicator function defined as

$$I_A(x) \triangleq \begin{cases} 1 \text{ for } x \in A \\ 0 \text{ for } x \notin A. \end{cases} \qquad (2.11)$$

(2.10) also used $(\mathbf{\Delta F}_i)$, which is defined as

$$(\mathbf{\Delta F}_i) \triangleq \begin{bmatrix} F_i(t_1) - F_i(t_0) \\ F_i(t_2) - F_i(t_1) \\ \vdots \\ F_i(t_M) - F_i(t_{M-1}) \end{bmatrix}. \qquad (2.12)$$

$F_i(x)$ is the cumulative distribution function under hypothesis H_i and $Pr_i(A)$ is the probability of event A occurring under hypothesis H_i. Note that, in the above integrals, we have assumed that $f_i(x) = 0$, for $x < 0$; this is the result of the envelope detector output of the radar system being always non-negative. The variance of the quantizer $Q(Z)$ is evaluated

13

as

$$\sigma_i^2[Q] \triangleq Var_i[Q(Z_1)] + 2 \sum_{j=1}^{m_i} Cov_i[Q(Z_1), Q(Z_{j+1})]$$

$$= E_i[Q^2(Z_1)] - \mu_i{}^2[Q(Z_1)]$$

$$+ 2 \sum_{j=1}^{m_i} E_i\{[Q(Z_1) - \mu_i[Q(Z_1)]][Q(Z_{j+1}) - \mu_i[Q(Z_1)]]\}$$

$$= E_i[Q^2(Z_1)] - \mu_i{}^2[Q(Z_1)] \qquad (2.13)$$

$$+ 2 \sum_{j=1}^{m_i} \{E_i[Q(Z_1)Q(Z_{j+1})] - \mu_i{}^2[Q(Z_1)]\}$$

$$= E_i[Q^2(Z_1)] - (2m_i + 1)\mu_i{}^2[Q(Z_1)]$$

$$+ 2 \sum_{j=1}^{m_i} E_i[Q(Z_1)Q(Z_{j+1})].$$

The power $E_i[Q^2(Z_1)]$ is evaluated by

$$E_i[Q^2(Z_1)] = \int_0^{+\infty} Q^2(z) f_i(z) dz$$

$$= \int_0^{+\infty} \sum_{k=1}^{M} q_k^2 I_{(t_{k-1}, t_k]}(z) I_{(t_{k-1}, t_k]}(z) f_i(z) dz$$

$$= \int_0^{+\infty} \sum_{k=1}^{M} q_k^2 I_{(t_{k-1}, t_k]}(z) f_i(z) dz$$

$$= \sum_{k=1}^{M} \int_0^{+\infty} q_k^2 I_{(t_{k-1}, t_k]}(z) f_i(z) dz \qquad (2.14)$$

$$= \sum_{k=1}^{M} \int_{t_{k-1}}^{t_k} q_k^2 f_i(z) dz$$

$$= \sum_{k=1}^{M} q_k^2 Pr_i\{z \in (t_{k-1}, t_k]\}$$

$$= \sum_{k=1}^{M} q_k^2 [F_i(t_k) - F_i(t_{k-1})]$$

$$= \mathbf{q}^T \hat{\mathbf{F}}_i \mathbf{q}$$

14

where the matrix $\hat{\mathbf{F}}_i$ is defined as

$$\hat{\mathbf{F}}_i \triangleq \mathrm{diag}\{F_i(t_1) - F_i(t_0), F_i(t_2) - F_i(t_1), \ldots, F_i(t_M) - F_i(t_{M-1})\}. \qquad (2.15)$$

The squared term in (2.13) can be rewritten as

$$
\begin{aligned}
\mu_i{}^2\left[Q(Z_1)\right] &= \sum_{k=1}^{M} q_k \left[F_i(t_k) - F_i(t_{k-1})\right] \sum_{l=1}^{M} q_l \left[F_i(t_l) - F_i(t_{l-1})\right] \\
&= \sum_{k=1}^{M} \sum_{l=1}^{M} q_k q_l \left[F_i(t_k) - F_i(t_{k-1})\right]\left[F_i(t_l) - F_i(t_{l-1})\right] \\
&= \mathbf{q}^T (\mathbf{\Delta F}_i)(\mathbf{\Delta F}_i)^T \mathbf{q}.
\end{aligned}
\qquad (2.16)
$$

Finally, the last term in (2.13) is given by

$$
\begin{aligned}
2\sum_{j=1}^{m_i} E_i \left[Q(Z_1)Q(Z_{j+1})\right] &= 2\sum_{j=1}^{m_i} E_i \left[\sum_{k=1}^{M} q_k I_{(t_{k-1},t_k]}(Z_1) \sum_{l=1}^{M} q_l I_{(t_{l-1},t_l]}(Z_{j+1})\right] \\
&= 2\sum_{j=1}^{m_i} E_i \left[\sum_{k=1}^{M} \sum_{l=1}^{M} q_k q_l I_{(t_{k-1},t_k]}(Z_1) I_{(t_{l-1},t_l]}(Z_{j+1})\right] \\
&= 2\sum_{j=1}^{m_i} \sum_{k=1}^{M} \sum_{l=1}^{M} q_k q_l E_i \left[I_{(t_{k-1},t_k]}(Z_1) I_{(t_{l-1},t_l]}(Z_{j+1})\right] \\
&= 2\sum_{j=1}^{m_i} \sum_{k=1}^{M} \sum_{l=1}^{M} q_k q_l Pr_i\{Z_1 \in (t_{k-1},t_k] \ \text{ AND } \ Z_{j+1} \in (t_{l-1},t_l]\} \\
&= \mathbf{q}^T \bar{\mathbf{P}}_i \mathbf{q},
\end{aligned}
$$

$$(2.17)$$

where the matrix $\bar{\mathbf{P}}_i$ is defined by its elements

$$\left[\bar{\mathbf{P}}_i\right]_{k,l} \triangleq 2\sum_{j=1}^{m_i} Pr_i\{Z_1 \in (t_{k-1},t_k] \ \text{ AND } \ Z_{j+1} \in (t_{l-1},t_l]\}. \qquad (2.18)$$

So now the variance $\sigma_i^2\left[Q(Z)\right]$ can be obtained by combining equations (2.14) through (2.18)

15

to yield

$$\sigma_i^2 \left[Q(Z)\right] = \mathbf{q}^T(\Delta \mathbf{F}_i)\mathbf{q} - (2m_i + 1)\mathbf{q}^T(\Delta \mathbf{F}_i)(\Delta \mathbf{F}_i)^T\mathbf{q} + \mathbf{q}\tilde{\mathbf{P}}_i\mathbf{q}$$

$$= \mathbf{q}^T\left[(\Delta \mathbf{F}_i) - (2m_i + 1)(\Delta \mathbf{F}_i)(\Delta \mathbf{F}_i)^T + \tilde{\mathbf{P}}_i\right]\mathbf{q} \qquad (2.19)$$

$$= \mathbf{q}^T[\hat{\mathbf{P}}_i + \hat{\mathbf{F}}_i]\mathbf{q}.$$

The matrix $\hat{\mathbf{P}}_i$ is defined as

$$\hat{\mathbf{P}}_i \triangleq \tilde{\mathbf{P}}_i - (2m_i + 1)(\Delta \mathbf{F}_i)(\Delta \mathbf{F}_i)^T. \qquad (2.20)$$

2.2 Evaluation of the Performance Measure for a Quantizer Function

Using the expressions for $\mu_i\left[Q(Z)\right]$ and $\sigma_i^2\left[Q(Z)\right]$ from the previous section the value of the performance measure for a quantizer function is given by

$$\begin{aligned}
\tilde{S}_4(\mathbf{q}) &\triangleq \frac{[\mu_1 - \mu_0]^2}{\nu\sigma_1^2 + (1-\nu)\sigma_0^2} \\
&= \frac{\left[\mathbf{q}^T(\Delta \mathbf{F}_1) - \mathbf{q}^T(\Delta \mathbf{F}_0)\right]^2}{\nu\mathbf{q}^T[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1]\mathbf{q} + (1-\nu)\mathbf{q}^T[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\mathbf{q}} \\
&= \frac{\left[\mathbf{q}^T[(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)]\right]^2}{\mathbf{q}^T\left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\right]\mathbf{q}} \qquad (2.21) \\
&= \frac{\left[[(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)]^T\mathbf{q}\right]^2}{\mathbf{q}^T\left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\right]\mathbf{q}}.
\end{aligned}$$

2.3 Evaluation of the Optimal Quantizer Function for Specified Breakpoints

A necessary condition for the performance measure to be maximized is for the gradient with respect to the level vector q to be zero. So we need to evaluate the gradient of equation (2.21). This is given by

$$
\nabla_{\mathbf{q}} \tilde{S}_4(Q) = \nabla_{\mathbf{q}} \left[\frac{\left[\mathbf{q}^T [(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)] \right]^2}{\mathbf{q}^T \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1 - \nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0] \right] \mathbf{q}} \right]
$$

$$
= \frac{2 \left[\mathbf{q}^T [(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)] \right] [(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)]}{\mathbf{q}^T \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1 - \nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0] \right] \mathbf{q}} \tag{2.22}
$$

$$
- \frac{2 \left[\mathbf{q}^T [(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)] \right]^2 \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1 - \nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0] \right] \mathbf{q}}{\left[\mathbf{q}^T \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1 - \nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0] \right] \mathbf{q} \right]^2}
$$

where $\nabla_{\mathbf{q}}$ denotes gradient with respect to the level vector q. Now define $\mathbf{q}^\circ$ as the vector which maximizes $\tilde{S}_4(Q)$.

$$
\mathbf{q}^\circ = \arg\{ \max_{\mathbf{q} \in \mathbf{R}^M} \tilde{S}_4(Q) \}. \tag{2.23}
$$

The necessary condition is

$$
\left[\nabla_{\mathbf{q}} \tilde{S}_4(Q) \right] \Big|_{\mathbf{q} = \mathbf{q}^*} = 0. \tag{2.24}
$$

If $\left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1 - \nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0] \right]$ is positive definite and $\mathbf{q}^T[(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)] > 0$ (i.e. $\mu_1 > \mu_0$), then equation (2.24) reduces to

$$
[(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)] - \lambda(\mathbf{q}^\circ) \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1 - \nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0] \right] \mathbf{q}^\circ = 0 \tag{2.25}
$$

where the multiplier $\lambda(\mathbf{q})$ is defined as

$$
\lambda(\mathbf{q}) = \frac{\mathbf{q}^T[(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)]}{\mathbf{q}^T \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1 - \nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0] \right] \mathbf{q}}. \tag{2.26}
$$

17

So we have

$$\mathbf{q}^{\circ} = \frac{\left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1 - \nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\right]^{-1}[(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)]}{\lambda(\mathbf{q}^{\circ})} \qquad (2.27)$$

as an expression for the optimal quantization function for fixed breakpoints. $\tilde{S}_4(Q)$ remains unchanged by scaling Q (i.e. $\tilde{S}_4(Q) = \tilde{S}_4(aQ)$, where a is a constant). This implies that $\lambda(\mathbf{q}^{\circ})$ does not affect the value of $\tilde{S}_4(Q)$, so all solutions to (2.25) are equivalent. One particular solution is where $\lambda(\mathbf{q}^{\circ}) = 1$:

$$\mathbf{q}^{\circ} = \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1 - \nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\right]^{-1}[(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)]. \qquad (2.28)$$

This is equivalent to

$$\left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1 - \nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\right]\mathbf{q}^{\circ} - [(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)] = 0. \qquad (2.29)$$

The matrix $\left[\nu\hat{\mathbf{F}}_1 + (1 - \nu)\hat{\mathbf{F}}_0\right]$ has the form

$$\left[\nu\hat{\mathbf{F}}_1 + (1 - \nu)\hat{\mathbf{F}}_0\right] = \text{diag}\{\nu[F_1(t_1) - F_1(t_0)] + (1 - \nu)[F_0(t_1) - F_0(t_0)],$$

$$\nu[F_1(t_2) - F_1(t_1)] + (1 - \nu)[F_0(t_2) - F_0(t_1)], \ldots,$$

$$\nu[F_1(t_M) - F_1(t_{M-1})] + (1 - \nu)[F_0(t_M) - F_0(t_{M-1})]\}. \qquad (2.30)$$

All terms of $\left[\nu\hat{\mathbf{F}}_1 + (1 - \nu)\hat{\mathbf{F}}_0\right]$ are positive, since its terms are probabilities, so its inverse $\left[\nu\hat{\mathbf{F}}_1 + (1 - \nu)\hat{\mathbf{F}}_0\right]^{-1}$ exists. This allows equation (2.29) to be written as

$$\left[\nu\hat{\mathbf{F}}_1 + (1 - \nu)\hat{\mathbf{F}}_0\right]^{-1}\left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1 - \nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\right]\mathbf{q}^{\circ}$$

$$-\left[\nu\hat{\mathbf{F}}_1 + (1 - \nu)\hat{\mathbf{F}}_0\right]^{-1}[(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)] = 0. \qquad (2.31)$$

This can be simplified to

$$\left[\mathbf{I} + \hat{\mathbf{K}}_4\right]\mathbf{q}^{\circ} - \left[\nu\hat{\mathbf{F}}_1 + (1 - \nu)\hat{\mathbf{F}}_0\right]^{-1}[(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)] = 0 \qquad (2.32)$$

18

where $\mathbf{I}$ is the $(M \times M)$ identity matrix and where we define

$$\hat{\mathbf{K}}_4 \triangleq \left[\nu\hat{\mathbf{F}}_1 + (1-\nu)\hat{\mathbf{F}}_0\right]^{-1}\left[\nu\hat{\mathbf{P}}_1 + (1-\nu)\hat{\mathbf{P}}_0\right]. \tag{2.33}$$

The components of $\hat{\mathbf{K}}_4$ are given by

$$\left[\hat{\mathbf{K}}_4\right]_{k,l} = \frac{\left[\nu\hat{\mathbf{P}}_1 + (1-\nu)\hat{\mathbf{P}}_0\right]_{k,l}}{\left[\nu\left[F_1(t_k) - F_1(t_{k-1})\right] + (1-\nu)\left[F_0(t_k) - F_0(t_{k-1})\right]\right]}. \tag{2.34}$$

We can define the vector $\mathbf{b}_4$ as

$$\begin{aligned}\mathbf{b}_4 &\triangleq \left[\nu\hat{\mathbf{F}}_1 + (1-\nu)\hat{\mathbf{F}}_0\right]^{-1}\left[(\boldsymbol{\Delta}\mathbf{F}_1) - (\boldsymbol{\Delta}\mathbf{F}_0)\right]\\[2mm]
&= \begin{bmatrix} \dfrac{F_1(t_1) - F_1(t_0) - F_0(t_1) + F_0(t_0)}{\nu\left[F_1(t_1) - F_1(t_0)\right] + (1-\nu)\left[[F_0(t_1) - F_0(t_0)]\right]} \\[3mm] \dfrac{F_1(t_2) - F_1(t_1) - F_0(t_2) + F_0(t_1)}{\nu\left[F_1(t_2) - F_1(t_1)\right] + (1-\nu)\left[[F_0(t_2) - F_0(t_1)]\right]} \\[2mm] \vdots \\[2mm] \dfrac{F_1(t_M) - F_1(t_{M-1}) - F_0(t_M) + F_0(t_{M-1})}{\nu\left[F_1(t_M) - F_1(t_{M-1})\right] + (1-\nu)\left[[F_0(t_M) - F_0(t_{M-1})]\right]} \end{bmatrix}.\end{aligned} \tag{2.35}$$

So we can rewrite equation (2.29) as $\left[\mathbf{I} + \hat{\mathbf{K}}_4\right]\mathbf{q}^\circ - \mathbf{b}_4 = 0$.

2.4 Evaluation of the Optimal Performance Measure

In this section, the optimal performance measure for fixed breakpoints is derived. This value is the maximum a value a quantizer with the given breakpoint vector t can achieve. We start with the performance measure

$$\tilde{S}_4(Q) = \tilde{S}_4(\mathbf{q(t)})\bigg|_{\mathbf{q}=\mathbf{q}^\bullet} = \frac{\left[\mathbf{q}^T\left[(\boldsymbol{\Delta}\mathbf{F}_1) - (\boldsymbol{\Delta}\mathbf{F}_0)\right]\right]^2}{\mathbf{q}^T\left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\right]\mathbf{q}}\bigg|_{\mathbf{q}=\mathbf{q}^\bullet}. \tag{2.36}$$

Now the expression for the optimal quantizer levels for fixed breakpoints,

$$\mathbf{q}^\circ = \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\right]^{-1}\left[(\boldsymbol{\Delta}\mathbf{F}_1) - (\boldsymbol{\Delta}\mathbf{F}_0)\right] \tag{2.37}$$

19

is substituted into the above expression to yield

$$\tilde{S}_4(\mathbf{q}^\circ) = \frac{\left[[(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)]^T \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\right]^{-1} [(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)]\right]^2}{[(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)]^T \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\right]^{-1} [(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)]}$$

$$= [(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)]^T \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\right]^{-1} [(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)].$$

$$(2.38)$$

2.5 Sufficiency of the Solution (2.28)

The solution (2.28) has been showed to be a necessary condition for maximizing the performance measure $\tilde{S}_4(\mathbf{q})$. In this section, the Schwartz inequality is used to show that (2.28) is also a sufficient condition for maximizing

$$\tilde{S}_4(\mathbf{q}) = \frac{[\mu_1(\mathbf{q}) - \mu_0(\mathbf{q})]^2}{\nu \sigma_1^2(\mathbf{q}) + (1-\nu)\sigma_0^2(\mathbf{q})}. \tag{2.39}$$

For simplification purposes define

$$C \triangleq \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\right] \tag{2.40}$$

and

$$\mathbf{v} \triangleq [(\Delta \mathbf{F}_1) - (\Delta \mathbf{F}_0)]. \tag{2.41}$$

By substituting these into the expression for $\tilde{S}_4(\mathbf{q})$ we obtain

$$\tilde{S}_4(\mathbf{q}) = \frac{[\mathbf{q}^T \mathbf{v}]^2}{\mathbf{q}^T C \mathbf{q}}$$

$$= \left[\frac{[\mathbf{q}^T \mathbf{v}]^2}{\mathbf{q}^T C \mathbf{q}}\right] \left[\frac{S_4(\mathbf{q}^\circ)}{\tilde{S}_4(\mathbf{q}^\circ)}\right]^2 \tag{2.42}$$

$$= \frac{[\mathbf{q}^T \mathbf{v} \mathbf{v}^T \mathbf{q}^\circ]^2}{\mathbf{q}^T C \mathbf{q} [\mathbf{v}^T \mathbf{q}^\circ]^2}.$$

20

Now by the Schwartz inequality, which for two vectors $\mathbf{x}$ and $\mathbf{z}$ implies that $(\mathbf{x}^T\mathbf{z})^2 \leq \mathbf{x}^T\mathbf{x}\mathbf{z}^T\mathbf{z}$. we obtain

$$
\begin{aligned}
\tilde{S}_4(\mathbf{q}) &\leq \frac{\mathbf{q}^T\mathbf{v}\mathbf{v}^T\mathbf{q}(\mathbf{q}^\circ)^T\mathbf{v}\mathbf{v}^T\mathbf{q}^\circ}{\mathbf{q}^T C\mathbf{q}\mathbf{v}^T\mathbf{q}^\circ\mathbf{v}^T\mathbf{q}^\circ} \\
&= \frac{\mathbf{q}^T\mathbf{v}\mathbf{v}^T\mathbf{q}(\mathbf{q}^\circ)^T\mathbf{v}\mathbf{v}^T\mathbf{q}^\circ}{\mathbf{q}^T C\mathbf{q}\mathbf{v}^T\mathbf{q}^\circ\mathbf{v}^T\mathbf{q}^\circ} \\
&= \frac{\mathbf{q}^T\mathbf{v}\mathbf{v}^T\mathbf{q}(\mathbf{q}^\circ)^T\mathbf{v}}{\mathbf{q}^T C\mathbf{q}(\mathbf{q}^\circ)^T\mathbf{v}} \\
&= \frac{\mathbf{q}^T\mathbf{v}\mathbf{v}^T\mathbf{q}(\mathbf{q}^\circ)^T\mathbf{v}}{\mathbf{q}^T C\mathbf{q}^\circ\mathbf{q}^T\mathbf{v}}.
\end{aligned}
\tag{2.43}
$$

Now by substituting the expression

$$
\mathbf{q}^\circ = \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1 - \nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\right]^{-1} [(\mathbf{\Delta F}_1) - (\mathbf{\Delta F}_0)] = C^{-1}\mathbf{v}
\tag{2.44}
$$

into the denominator of the above expression. we obtain

$$
\begin{aligned}
\tilde{S}_4(\mathbf{q}) &\leq \frac{\mathbf{q}^T\mathbf{v}\mathbf{v}^T\mathbf{q}(\mathbf{q}^\circ)^T\mathbf{v}}{\mathbf{q}^T CC^{-1}\mathbf{v}\mathbf{q}^T\mathbf{v}} \\
&= \frac{\mathbf{q}^T\mathbf{v}\mathbf{v}^T\mathbf{q}(\mathbf{q}^\circ)^T\mathbf{v}}{\mathbf{q}^T\mathbf{v}\mathbf{q}^T\mathbf{v}} \\
&= \frac{\mathbf{q}^T\mathbf{v}\mathbf{v}^T\mathbf{q}(\mathbf{q}^\circ)^T\mathbf{v}}{\mathbf{q}^T\mathbf{q}\mathbf{v}^T\mathbf{v}} \\
&= (\mathbf{q}^\circ)^T\mathbf{v} = \tilde{S}_4(\mathbf{q}^\circ).
\end{aligned}
\tag{2.45}
$$

Thus we have now shown that the expression for optimal quantizer satisfies the necessary and sufficient conditions for optimality.

2.6 Evaluation of Quantizer with Optimal Levels and Breakpoints

Here we derive the quantizer function with optimal levels and breakpoints that maximizes the performance measure $\tilde{S}_4(\mathbf{q})$. Specifically. we are maximizing the function $\tilde{S}_4(\mathbf{q}^\circ(t))$. We

need to evaluated the gradient of the performance measure with respect to the breakpoint vector,

$$\frac{\partial}{\partial t_k} \tilde{S}_4(\mathbf{q}^\circ(\mathbf{t})) =$$

$$= \frac{\partial}{\partial t_k} \left[[(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)]^T \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0] \right]^{-1} [(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)] \right]$$

$$= \left[\frac{\partial}{\partial t_k} [(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)]^T \right] \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0] \right]^{-1} [(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)]$$

$$+ [(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)]^T \left[\frac{\partial}{\partial t_k} \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0] \right]^{-1} \right] [(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)]$$

$$+ [(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)]^T \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0] \right]^{-1} \left[\frac{\partial}{\partial t_k} [(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)] \right]$$

$$(2.46)$$

We use the fact for invertible matrices that $\frac{\partial A^{-1}}{\partial x} = -A \frac{\partial A}{\partial x} A^{-1}$ and equation (2.46) to obtain

$$\left[\frac{\partial}{\partial t_k} [(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)]^T \right] \mathbf{q}^\circ$$

$$- (\mathbf{q}^\circ)^T \left[\frac{\partial}{\partial t_k} \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0] \right] \right] \mathbf{q}^\circ$$

$$+ (\mathbf{q}^\circ)^T \left[\frac{\partial}{\partial t_k} [(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)] \right]. \qquad (2.47)$$

This further reduces to

$$2(\mathbf{q}^\circ)^T \left[\frac{\partial}{\partial t_k} [(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)] \right]$$

$$- (\mathbf{q}^\circ)^T \left[\frac{\partial}{\partial t_k} \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0] \right] \right] \mathbf{q}^\circ$$

$$= (\mathbf{q}^\circ)^T \left[2\frac{\partial}{\partial t_k} [(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)] - \left[\frac{\partial}{\partial t_k} \left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0] \right] \right] \mathbf{q}^\circ \right].$$

$$(2.48)$$

So a necessary condition for the vector t to maximize the performance measure is

$$0 = (\mathbf{q}^\circ)^T \left[2 \left[\frac{\partial}{\partial t_k} [(\Delta\mathbf{F}_1) - (\Delta\mathbf{F}_0)] \right] \right]$$

22

$$-\left[\frac{\partial}{\partial t_k}\left[\nu[\hat{\mathbf{P}}_1 + \hat{\mathbf{F}}_1] + (1-\nu)[\hat{\mathbf{P}}_0 + \hat{\mathbf{F}}_0]\right]\right]\mathbf{q}^\circ\right], \quad \text{for } k = 1, 2, 3, \ldots, M-1. \quad (2.49)$$

This can be expanded into more detail:

$$0 = (\mathbf{q}^\circ)^T \left\{ 2 \begin{bmatrix} 0 \\ \vdots \\ 0 \\ f_1(t_k) - f_0(t_k) \\ -f_1(t_k) + f_0(t_k) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right.$$

$$-\begin{bmatrix} 0 & & & & & & & \\ & 0 & & & & & & \\ & & \ddots & & & & & \\ & & & 0 & & & & \\ & & & & (1-\nu)f_0(t_k) + \nu f_1(t_k) & & & \\ & & & & & -(1-\nu)f_0(t_k) - \nu f - 1(t_k) & & \\ & & & & & & 0 & \\ & & & & & & & \ddots \\ & & & & & & & & 0 \end{bmatrix}\mathbf{q}^\circ$$

$$\left. -\left[\frac{\partial}{\partial t_k}\left[\nu\hat{\mathbf{P}}_1 + (1-\nu)\hat{\mathbf{P}}_0\right]\right]\mathbf{q}^\circ\right\} \quad \text{for } k = 1, 2, 3, \ldots, M-1 \quad (2.50)$$

$$0 = 2\left[f_1(t_k) - f_0(t_k)\right]\left[q_k^\circ + q_{k+1}^\circ\right] - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ q_k\nu f_1(t_k) + q_k(1-\nu)f_0(t_k) \\ -q_{k+1}\nu f_1(t_k)q_{k+1}(1-\nu)f_0(t_k) \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \mathbf{q}^\circ$$

23

$$-(\mathbf{q}^\circ)^T \left[\frac{\partial}{\partial t_k} \left[\nu \hat{\mathbf{P}}_1 + (1 - \nu)\hat{\mathbf{P}}_0 \right] \right] \mathbf{q}^\circ \quad \text{for } k = 1, 2, 3, \ldots, M - 1 \qquad (2.51)$$

2.7 Numerical Results

In this section, we evaluate the performance of the memoryless quantizer discriminators via computer simulation. Although the optimal quantization functions may be computed for any m-dependent processes for which the marginal and bivariate distributions of the data are known under both hypotheses, we consider only the case typical to radar systems: ρ-mixing data from observations of the radar return envelope. ρ-mixing implies that $Cov\{Z_k, Z_{k+n}\} \leq \rho_n$, where $\rho_n \longrightarrow 0$ as $n \longrightarrow \infty$. We shall assume that the data are samples of the radar return envelope, which is either from a target (hypothesis H_1) or a decoy (hypothesis H_0.) Note that the radar has already detected the object (i.e. the target or decoy), but now must decide whether the return is from a target or decoy. Note that in our problems we neglect the possibility of detecting clutter or other objects. We define the *probability of false alarm* as the probability that the discriminator declares a decoy to be the target. The *probability of miss* is the probability that a target is declared a decoy. Thus the *probability of detection* is the probability that the discriminator declares a target a target.

We consider two discrimination cases (refer to Table 2.1). Under Case 1, the target's envelope samples have marginal pdfs which are lognormal, while the decoy's marginal pdfs are Rayleigh. The observations under each hypothesis have matched means and powers. For Case 2, both hypotheses have Rayleigh marginal pdfs. However, under Case 2, a 3dB (H_1 vs H_0) difference in power exists between the two hypotheses. The observations are assumed stationary and ρ-mixing. Appendix C is a summary of the necessary marginal and

24

Case	pdfs	Power Ratio	Mean Ratio	Decorrelation Times
	H_1 vs H_0	H_1 vs H_0	H_1 vs H_0	τ_1, τ_0
Case 1	Lognormal vs Rayleigh	0	0	0.130290, 0.013029
Case 2	Rayleigh vs Rayleigh	3	—	0.130290, 0.013029

Table 2.1: Discrimination Cases

bivariate pdfs for the lognormal and Rayleigh processes.

The Rayleigh processes are generated by underlying Gaussian processes (i.e. the inphase and quadrature components.) We denote the envelope observations as $\{Z_i\}_{i=1}^{\infty}$. The Rayleigh envelope process is generated by

$$Z_i = \sqrt{X_i^2 + Y_i^2}, \quad i = 1, 2, 3, \ldots \tag{2.52}$$

where $\{X_i\}_{i=1}^{\infty}$ and $\{Y_i\}_{i=1}^{\infty}$ are mutually independent Gaussian stationary ρ-mixing processes. This implies that $\{Z_i\}_{i=1}^{\infty}$ is also stationary and ρ-mixing. The underlying Gaussians are generated by

$$X_i = \rho X_{i-1} + \sqrt{1 - \rho^2} V_i$$

$$Y_i = \rho Y_{i-1} + \sqrt{1 - \rho^2} W_i, \quad \text{for } i = 2, 3, \ldots \tag{2.53}$$

with

$$X_1 = \sigma V_1$$

$$Y_1 = \sigma W_1 \tag{2.54}$$

where $\{V_i\}_{i=1}^{\infty}$ and $\{W_i\}_{i=1}^{\infty}$. are mutually independent sequences of i.i.d. (independent and identically distributed) zero mean/unit variance Gaussian random variables. σ is the

standard deviation of the underlying Gaussians, while ρ is the correlation coefficient for adjacent samples. Thus the underlying Gaussians are stationary ρ-mixing processes with correlation coefficient ρ for adjacent samples.

The correlation coefficient ρ is related to the decorrelation time τ (see Table 2.1) in the following manner: τ is defined to be the time it takes for the correlation coefficient between the first sample and another sample to decrease by a factor of e^{-1}. In our simulations, we assume that they are uncorrelated, when the correlation coefficient between the first and j-th radar sample drops below 0.1. Since the underlying processes are Gaussian, they will also be independent. Thus we can assume m-dependence and define m_0 and m_1 as the number of samples under H_1 and H_0 respectively it takes the correlation to drop to below 0.1, respectively.

For very large targets, the radar return envelope samples are often approximated by a *lognormal process*. Our lognormal process is simulated by exponentiating an underlying Gaussian process:

$$Z_i = \exp(X_i + \mu), \;\; i = 1, 2, 3, \ldots \tag{2.55}$$

where X_i is generated in the same manner as equations (2.53) and (2.54). Unlike the Rayleigh processes which have underlying Gaussians with zero mean, the underlying Gaussians for the lognormal process may have a mean μ.

To generate the quantizer functions, the marginal cdfs for each hypothesis are required. To compute the matrices $\hat{P}_1$ and $\hat{P}_0$, the sum of bivariate cdfs over the m-dependence interval must be computed for each hypothesis. Specifically, the sums $\sum_{j=1}^{m_i} F_i^{(1,j+1)}(x,y)$ must be computed for $i = 0, 1$ corresponding to H_1 and H_0, where $F_i^{(1,j+1)}(x,y)$ is the joint cdf for samples Z_1 and Z_{j+1} and m_i is the m-dependence length for hypothesis H_i. The decorrelation times listed in Table 2.1 imply that the m-dependence lengths are 300 and

26

30 for H_1 and H_0, respectively. The Rayleigh and lognormal marginal and sum of bivariate pdfs for both Case 1 and Case 2 are evaluated at a discrete grid of evenly spaced points. These points are chosen to lie in the support of the marginal density – that is the maximum and minimum sample values are computed so that the probability that a sample exceeds the maximum value of the support or falls below the minimum value of the support is 0.00005. 301 grid points are used over the support.

For each case, three classes of quantization functions are generated. All quantization functions are chosen to maximize the performance measure

$\bar{S}_3(Q) = [\mu_1 - \mu_0]^2/[\sigma_1^2 + \sigma_0^2]$. The first class of quantizers have uniform breakpoints and optimal levels. The second class of quantizers have optimal breakpoints and optimal levels. Finally, the third class of quantizers were obtained by quantizing a continuous nonlinearity. The continuous nonlinearity is quantized by

$$Q(x) \triangleq \begin{cases} g(t_0), & \text{if } x \leq t_0 \\ [g(t_i) + g(t_{i+1})]/2, & \text{if } t_i \leq x \leq t_{i+1}, \ i = 0, 1, ..., M-1 \\ g(t_M), & \text{if } x \geq t_M \end{cases} \qquad (2.56)$$

where t_i are the breakpoints and where $g(x)$ is the continuous nonlinearity which maximizes the performance measure $\bar{S}_3(Q)$ (see [6]). The quantizer functions with uniform breakpoints and optimal levels are computed via equation (2.28). The quantizer functions with optimal levels and optimal breakpoints are computed using equation (2.28) and a gradient search technique over varying breakpoints. Appendix A supplies the some of the required derivatives needed to compute the derivative of the performance measure for a gradient technique. However, in the actual computations, our simulations used a finite difference gradient technique.

Quantization functions from the various classes are computed for various number of levels. Tables 2.2 and 2.3 summarizes the quantization functions computed. For each

27

Quantization Levels	Quantized g0	Uniform Quantizer	Optimal Quantizer
2	0.0000716237	0.0000761859	0.0000814555
4	0.0000435368	0.0026132100	0.0003927096
8	0.0000047076	0.0004527053	0.0102431728
16	0.0008984112	0.0019421706	0.0110042309
32	0.0071742339	0.0092516430	0.0111641670
64	0.0101809436	0.0109332995	
128	0.0112417946	0.0113493744	

Table 2.2: Values of $\tilde{S}_3$ for Case 1 Quantizers

quantization function listed in Tables 2.2 and 2.3, the corresponding performance measure is also listed.

Figure 2.1 is a graph of the performance measure versus the number of quantization levels for each quantization class. As expected, as the number of levels increases the performance measure also increases. Also, the performance measure saturates as the number of quantization levels becomes very large. The results in Figure 2.1 are intuitively pleasing: for a fixed number of quantization levels the quantizer with optimal breakpoints and optimal levels has a greater performance measure than the quantizer with uniform breakpoint and optimal levels, which has a performance measure greater than the quantized continuous nonlinearity. This result is expected, since the quantized continuous nonlinearity with M-levels and uniform breakpoints is a subclass of the M-level quantizer functions with uniform

Quantization Levels	Quantized g0	Uniform Quantizer	Optimal Quantizer
2	0.0009132413	0.0009184310	0.0029278097
4	0.0028830939	0.0029171822	0.0029428345
8	0.0029869056	0.0029905122	0.0029954810
16	0.0029979991	0.0029982538	0.0029988189
32	0.0029994955	0.0029995113	0.0029996180
64	0.0029998175	0.0029998184	
128	0.0029998948	0.0029998948	

Table 2.3: Values of $\tilde{S}_3$ for Case 2 Quantizers

breakpoints and quantizers with M-levels and uniform breakpoints are subclass of quantizers with M-levels.

Typical quantization functions are shown in Figures 2.3 to 2.10. Figure 2.3 is the 128-level uniform quantizer for Case 1. The 8-level uniform quantizer and the 8-level quantized continuous nonlinearity do not have the abrupt changes for small and large z values that the 128-level quantizer has. But the optimal 8-level quantizer comes close to the shape of the 128-level quantizer. The differences for the quantizers for Case 2 are also similar. Note that the general shape of the quantized nonlinearity in Figure 2.10 seems to be different from the other quantizers for Case 2. But note the the general shape of the quantizer is the same - it differs only by a scalar constant. (Note a quantizer may be scaled without changing the performance.)

The discriminator structure is depicted in Figure 2.11. A maximum number of samples per test criterion was added to the sequential test for practicality. For Case 1 discriminators, the maximum number of samples permitted was 2000. For Case 2, the maximum number of samples permitted was 4000. Each discriminator was tested using random data sequences. Also, each case was evaluated with the desired error probabilities $\alpha = \beta = 10^{-2}$ and $\alpha = \beta = 10^{-3}$. When the discriminators were evaluated with $\alpha = \beta = 10^{-2}$ as the desired error probabilities, 1000 random sample paths from each hypothesis were utilized. For the discriminators designed for $\alpha = \beta = 10^{-3}$, 10000 random sample paths from each hypothesis were utilized.

Figures 2.12 through 2.15 are examples of simulated paths from each hypothesis. Tables 2.4 through 2.7 summarize the results from the simulations for the quantizer discriminators. Listed for each discriminator is the probability of miss, probability of detection, expected number of samples to make a decision, and the performance measure. Examining the results one can see that generally, as the number of quantization levels increase, the performance of the discriminator improves.

Examining the results from Case 1 we see that the minimum number of quantization levels for a uniform quantization function to result in good performance was 32. The quantization function with 32 levels designed for $P_f = P_m = 10^{-2}$ had P_f=0.003, P_d=0.991, and an average sample number of 516. The quantization functions with less levels had P_f≈1. The quantizer function with optimal breakpoints and levels required only 8 quantization levels to result in reasonable performance, the quantized continuous nonlinearity required 32 levels to yield reasonable performance. For the Case 1 quantizer discriminators with desired $P_f = P_m = 10^{-3}$, error probabilities were slightly less than those of the quantizer discriminators with $P_f = P_m = 10^{-2}$, but the average sample numbers increased; this is
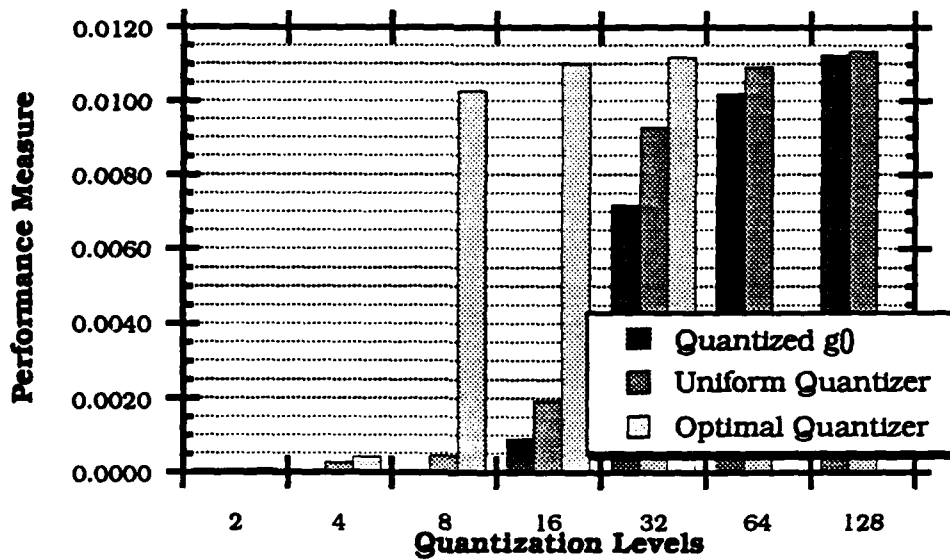
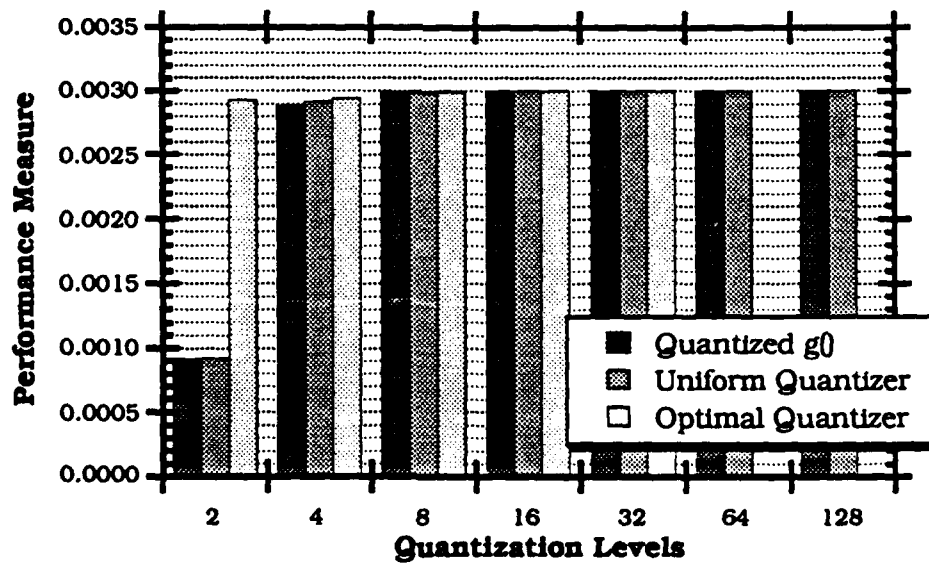**Figure 2.1: Performance Measures for Case 1 Quantizers**

Figure 2.2: Performance Measures for Case 2 Quantizers

expected since the decision thresholds move farther apart for smaller desired error proba-
bilities. For Case 2, the minimum number of quantization levels for good performance is 4
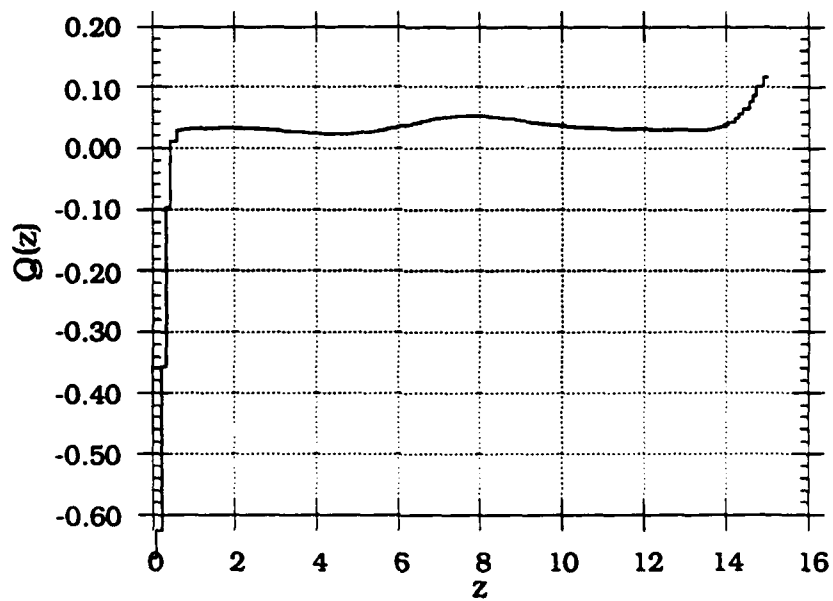for both optimal and uniform quantization functions.

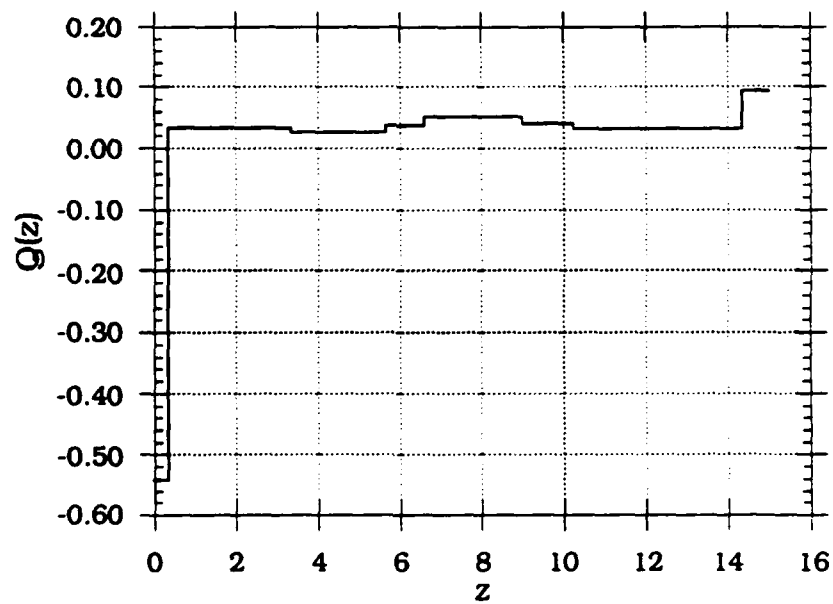**Figure 2.3: 128-Level Uniform Quantizer Function for Case 1**

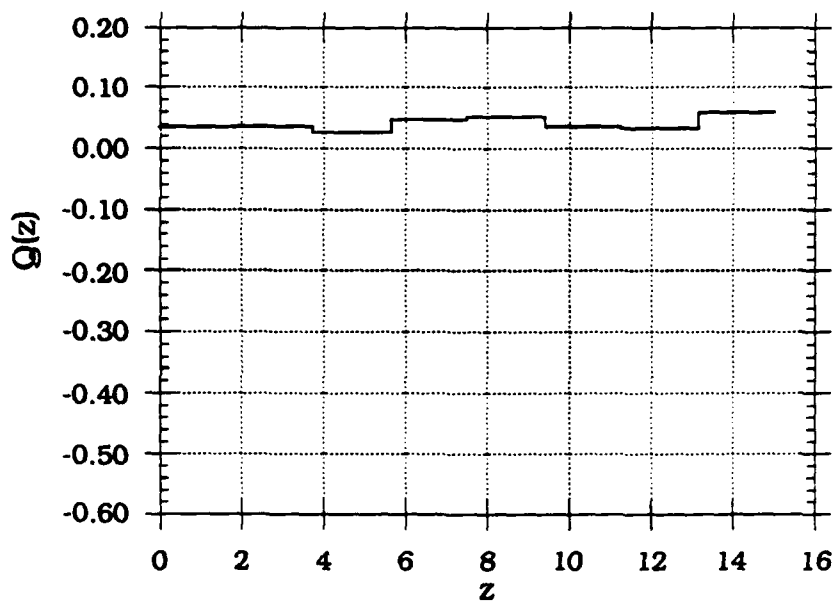

Figure 2.4: 8-Level Optimal Quantizer Function for Case 1

33

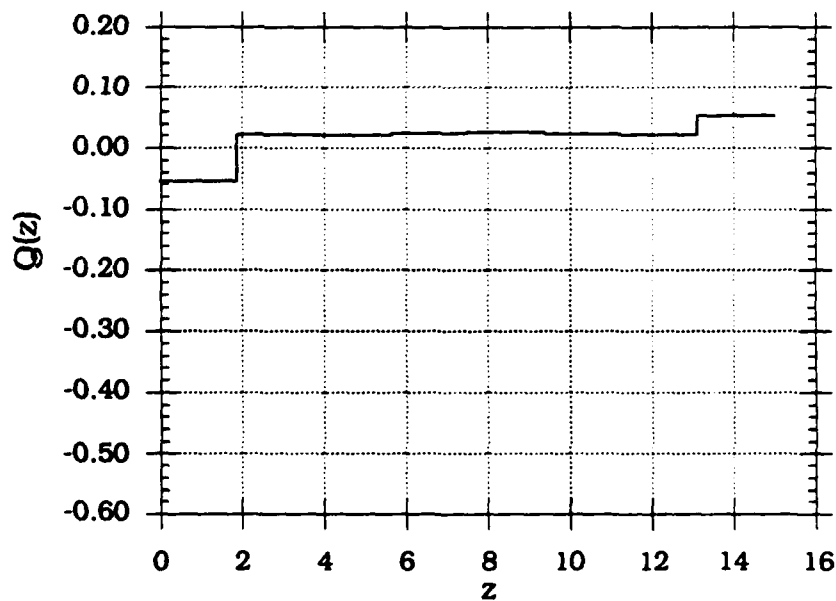**Figure 2.5: 8-Level Uniform Quantizer Function for Case 1**

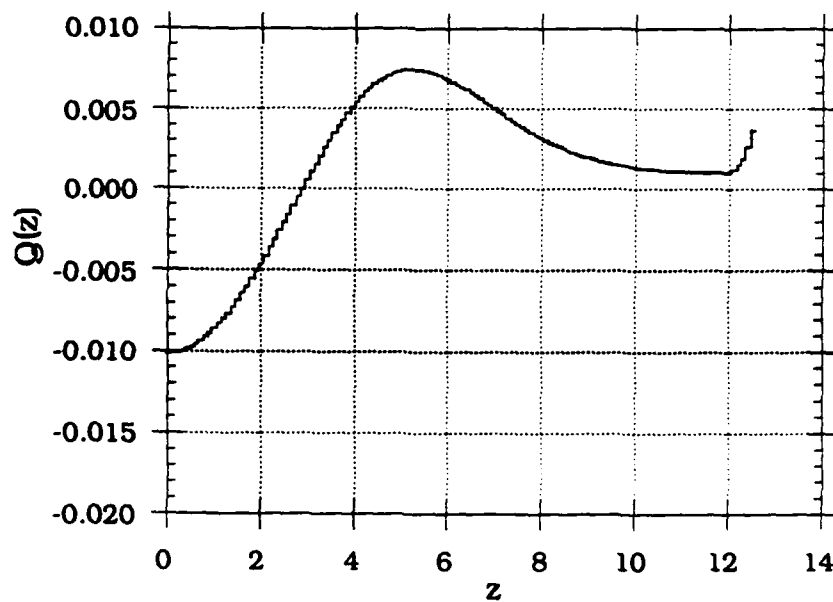**Figure 2.6: 8-Level Quantized g[] Function for Case 1**

Figure 2.7: 128-Level Uniform Quantizer Function for Case 2
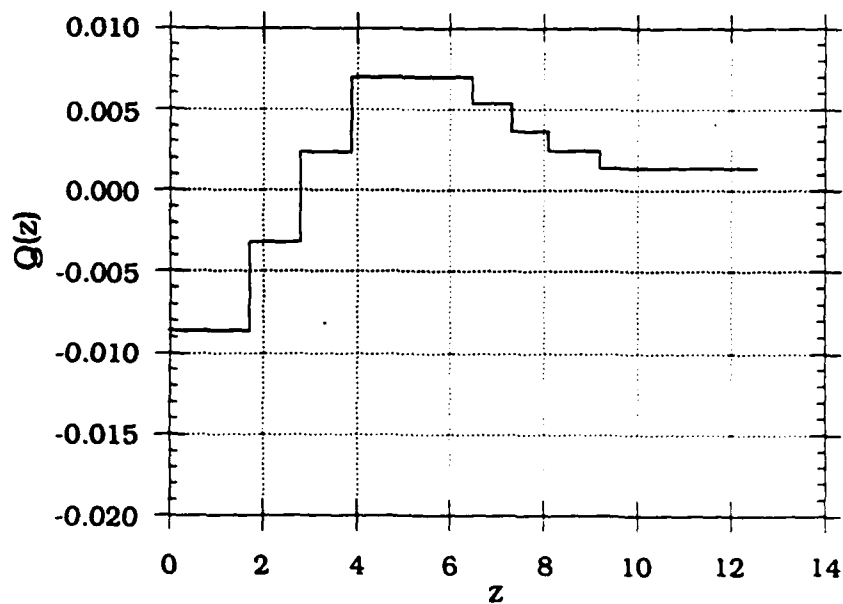

Figure 2.8: 8-Level Optimal Quantizer Function for Case 2
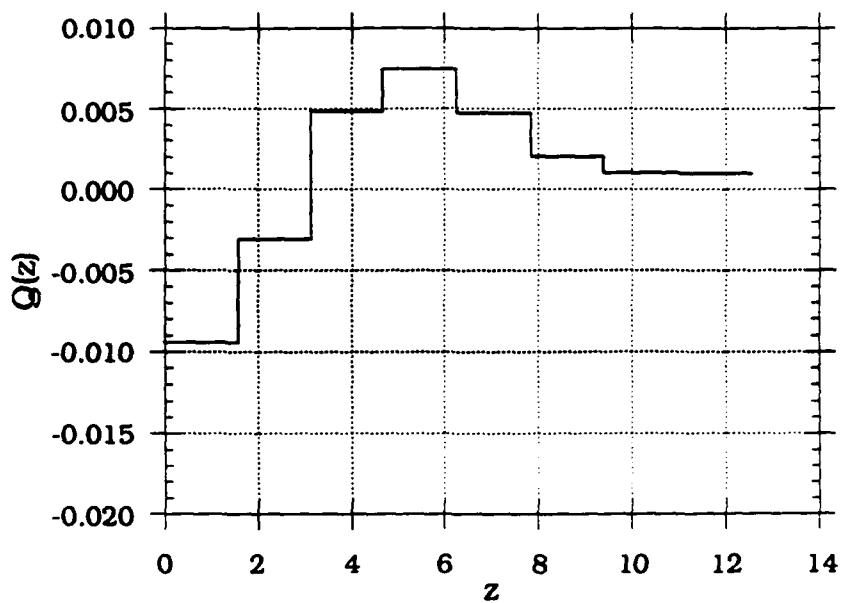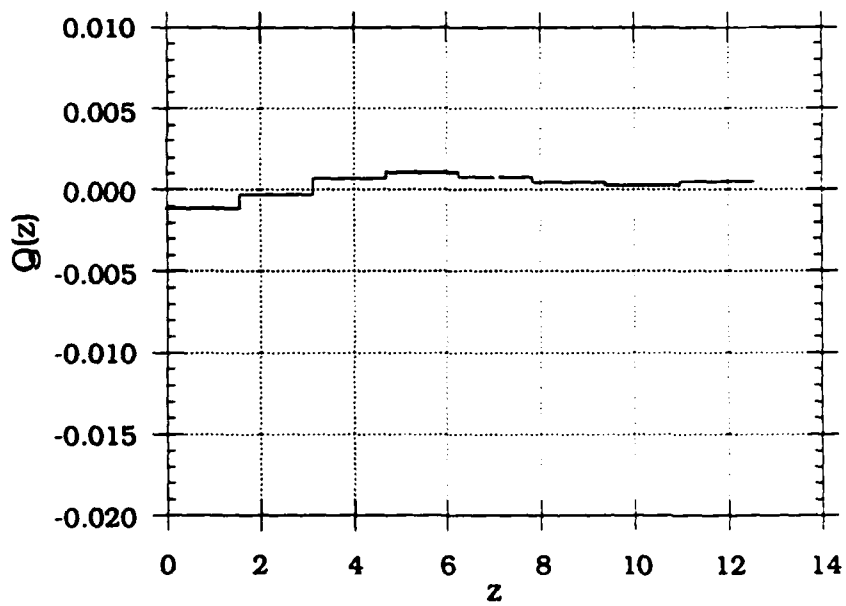
35

Figure 2.9: 8-Level Uniform Quantizer Function for Case 2
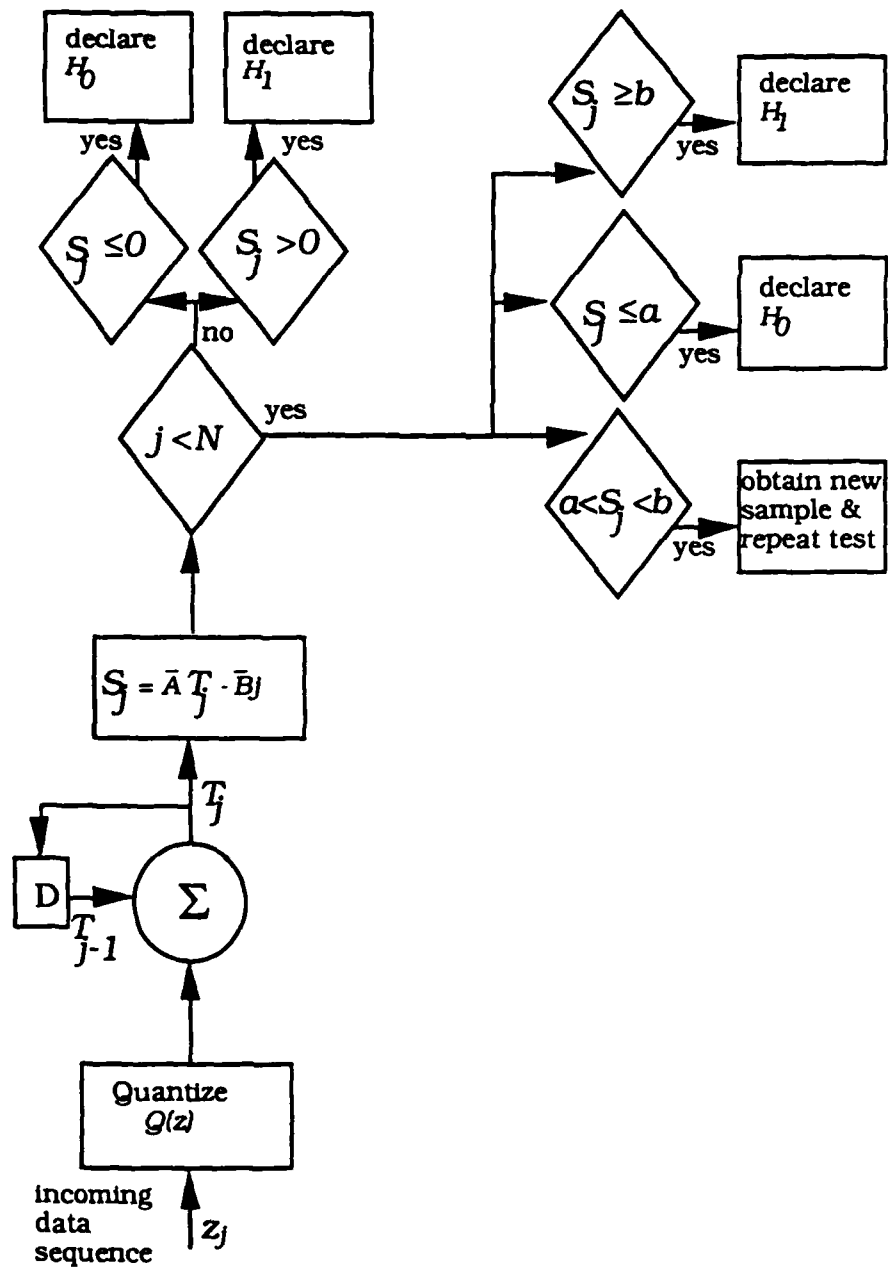

Figure 2.10: 8-Level Quantized g[] Function for Case 2

36

Figure 2.11: Memoryless Quantizer Discriminator Structure

Uniform Quantizer	P_f	P_d	$E[n]$	$\tilde{S}_3$
2	1.0	1.0	1997	0.0000761859
4	1.0	1.0	1998	0.0002613210
8	1.0	0.996	1986	0.0004527053
16	1.0	0.976	1573	0.0019421706
32	0.003	0.991	516	0.0092516428
64	0.0	0.985	527	0.0109332995
128	0.0	0.988	537	0.0113493742
Optimal Quantizer				
2	1	1	1997	0.0000814555
4	1	0.998	1983	0.0003927096
8	0	0.995	665	0.0102431725
16	0	0.984	541	0.0110042309
32	0	0.987	535	0.0111641665
Quantized g0				
2	0	0.007	1595	0.0000716237
4	0	0	2000	0.0000435368
8	0	0.191	2000	0.0000047076
16	1	0.965	1928	0.0008984112
32	0.025	0.972	617	0.0071742339
64	0	0.977	473	0.0101809439
128	0	0.983	488	0.0112417950

Table 2.4: Results for Case 1 Quantizers for desired $P_f = P_m = 10^{-2}$

Uniform Quantizer	P_f	P_d	$E[n]$	$\tilde{S}_3$
2	1.0	1.0	1999	0.0000761859
4	1.0	1.0	1997	0.0002613210
8	1.0	0.9999	1997	0.0004527053
16	1.0	0.9873	1790	0.0019421706
32	0.0143	0.9875	754	0.0092516428
64	0.0	0.9881	779	0.0109332995
128	0.0	0.9896	782	0.0113493742
Optimal Quantizer				
2	1	1	1999	0.0000814555
4	1	1	1998	0.0003927096
8	0	0.9946	963	0.0102431725
16	0.0001	0.9858	794	0.0110042309
32	0.0001	0.9895	783	0.0111641665
Quantized g0				
2	0	0.0003	1586	0.0000716237
4	0	0	2000	0.0000435368
8	0	0.1900	2000	0.0000047076
16	1	0.9902	1999	0.0008984112
32	0.0838	0.9773	884	0.0071742339
64	0.0016	0.9816	694	0.0101809439
128	0.0001	0.9862	719	0.0112417950

Table 2.5: Results for Case 1 Quantizers for desired $P_f = P_m = 10^{-3}$

Uniform Quantizer	P_f	P_d	$E[n]$	$\tilde{S}_3$
2	1	1	3314	0.0009184310
4	0.001	0.999	2387	0.0029171822
8	0.003	0.998	2344	0.0029905121
16	0.002	0.994	2332	0.0029982539
32	0.003	0.996	2332	0.0029995114
64	0.001	0.997	2330	0.0029998184
128	0	0.997	2329	0.0029998949
Optimal Quantizer				
2	0.738	0.999	1958	0.0029278097
4	0.015	0.996	2313	0.0029428344
8	0.003	0.997	2333	0.0029954809
16	0.002	0.997	2321	0.0029988189
32	0.001	0.995	2336	0.0029996179
Quantized g0				
2	0	0.762	3142	0.0009132413
4	0.001	0.999	2420	0.0028830940
8	0.002	0.999	2375	0.0029869056
16	0.003	0.991	2336	0.0029979990
32	0.001	0.996	2331	0.0029994954
64	0.002	0.998	2338	0.0029998174
128	0.003	0.994	2323	0.0029998948

Table 2.6: Results for Case 2 Quantizers for desired $P_f = P_m = 10^{-2}$

Uniform Quantizer	P_f	P_d	$E[n]$	$\tilde{S}_3$
2	1	1	3651	0.0009184310
4	0	0.9920	2685	0.0029171822
8	0	0.9921	2658	0.0029905121
16	0	0.9922	2668	0.0029982539
32	0.0001	0.9929	2663	0.0029995114
64	0	0.9917	2662	0.0029998184
128	0	0.9919	2660	0.0029998949
Optimal Quantizer				
2	0.9625	0.9996	2404	0.0029278097
4	0.0179	0.9980	2680	0.0029428344
8	0.0001	0.9916	2659	0.0029954809
16	0.0001	0.9908	2665	0.0029988189
32	0	0.9926	2665	0.0029996179
Quantized $g0$				
2	0	0.4924	3333	0.0009132413
4	0	0.9912	2692	0.0028830940
8	0	0.9923	2670	0.0029869056
16	0	0.9924	2659	0.0029979990
32	0	0.9938	2662	0.0029994954
64	0	0.9930	2666	0.0029998174
128	0	0.9924	2664	0.0029998948

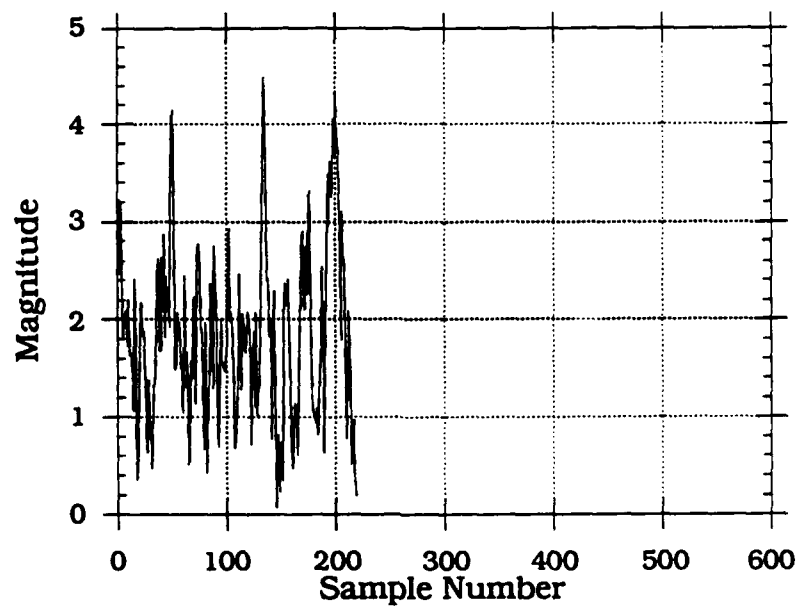Table 2.7: Results for Case 2 Quantizers for desired $P_f = P_m = 10^{-3}$
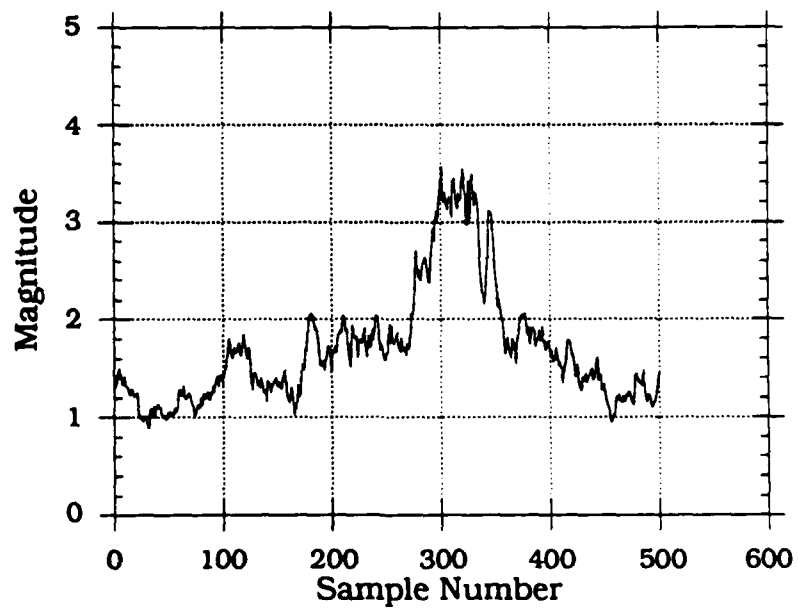
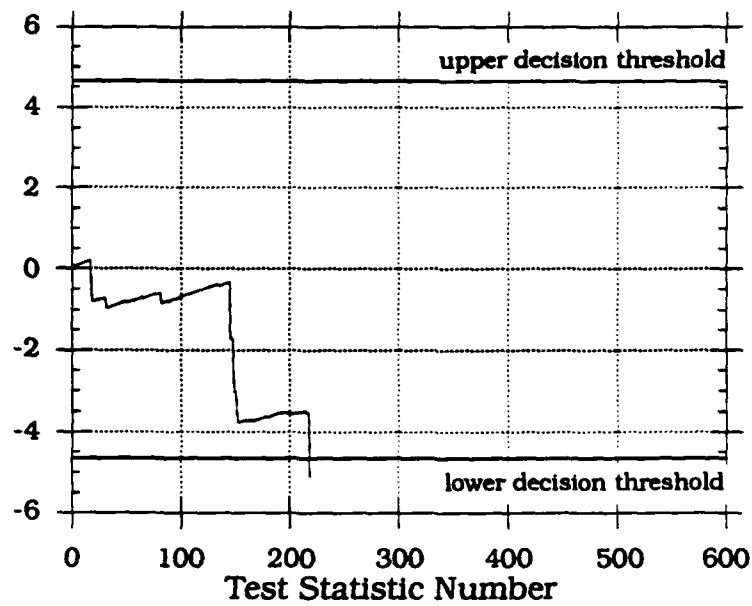**Figure 2.12: Sample Path from** H_0

Figure 2.13: Sample Path from H_1
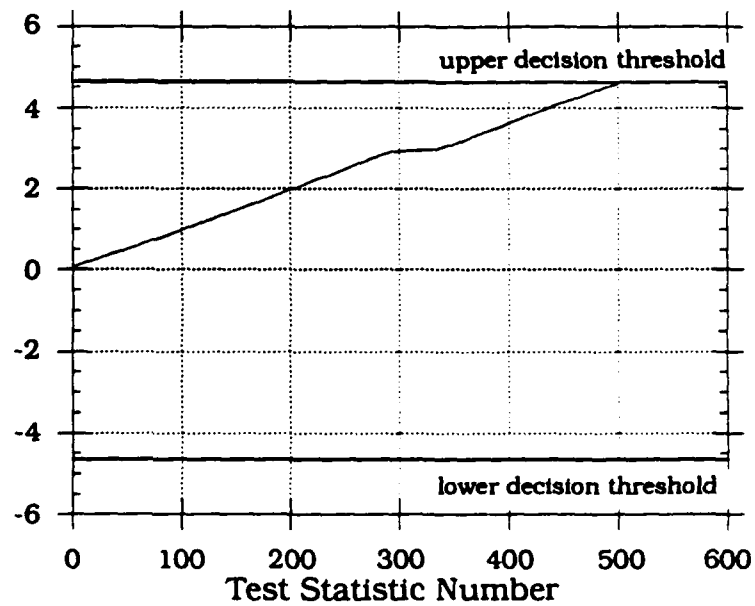
42

Figure 2.14: Test Statistic for H_0

Figure 2.15: Test Statistic for H_1

43

Chapter 3

Estimation and Discrimination

In the previous chapter we developed quantization functions which optimized the performance measure $\tilde{S}_4$ defined in equation (2.7). As mentioned in that chapter, the marginal cdfs of the data under each hypothesis were required to solve for the quantization functions. Also required were the sums $\sum_{j=1}^{m_i} F_i^{(1,j+1)}(x,y)$, where $F_i^{(1,j+1)}(x,y)$ was the joint cdf of the data for samples Z_1 and Z_{j+1} under hypothesis i, and m_i was the m-dependence length under H_i. For the continuous nonlinearities of [5][6], pdfs rather than cdfs were required.

The results presented in the previous chapter were obtained by using the actual cdfs of the various discrimination cases. However, in this chapter, it is assumed that the cdfs of the data are not known. This is a more realistic problem, since in many engineering problems the distributions of the data are not available. Therefore, in this chapter the pdfs of the data will be estimated from the training data introduced in Chapter 1, and via numerical integration techniques the cdfs will be obtained. Then quantization functions will be computed and implemented in simulated discriminators for evaluation. Thus the feasibility of estimation and discrimination techniques with memoryless quantizer discriminators is

addressed for the discrimination cases of the previous chapter.

3.1 Kernel Density Estimators

For the estimation of our pdfs we utilize kernel density estimators. The idea behind these estimators is that each observation X_k is replaced by a function of X_k. Then the functions are summed to yield the estimate of the density $\hat{f}(x)$. The kernel function produces a smoothing effect and, if the kernel satisfies certain constraints, the estimate will also have desirable properties. For our application, the main advantage of the kernel density estimator over a histogram method is the smoothing characteristic. The kernel density estimators are introduced in the following paragraph.

Given the data observations, $X_1, X_2, \ldots, X_n$, it is desired to estimate the marginal pdf of the data $f(x)$. It is assumed that the process $\{X\}_{k=1}^{\infty}$ is stationary. The kernel density estimate, denoted $\hat{f}(x; n)$, where n represents the number of observations used by the estimator, is defined as

$$\hat{f}(x; n) = \frac{1}{nh_n} \sum_{k=1}^{n} K\left(\frac{x - X_k}{h_n}\right).$$ (3.1)

The function $K(\cdot)$ is called the kernel function and h_n is usually referred to as the *window width* or *bandwidth* parameter.

Under certain conditions the kernel estimate has been shown to be asymptotically unbiased and strongly consistent. Asymptotically unbiased means that

$$\lim_{n \to \infty} E\left[\hat{f}(x; n)\right] = f(x)$$ (3.2)

45

and strongly consistent means that

$$\lim_{n \to \infty} \hat{f}(x; n) = f(x). \tag{3.3}$$

These two characteristics are desireable for an estimator since they imply that more observations improves the estimator's accuracy.

[8] shows that, if $x_1, x_2, \ldots, x_n$ are independent and identically distributed, and if

(1) $K(\cdot)$ is a density, that is $\int_{-\infty}^{\infty} K(x)dx = 1$ and $K(x) \geq 0, \forall x$.

(2) $\lim_{x \to \infty} |x| K(x) = 0$

(3) $\sup_x K(x) < \infty$

(4) $\lim_{n \to \infty} h_n = 0$

(5) $\lim_{n \to \infty} n h_n = \infty$

(6) $\sum_{n=1}^{\infty} \exp(-\alpha n h_n) < \infty, \forall \alpha > 0$

then

$$\lim_{n \to \infty} E\left[\hat{f}(x; n)\right] = f(x)$$

and

$$\lim_{n \to \infty} \hat{f}(x; n) = f(x). \tag{3.4}$$

For various conditions, the kernel density estimators have also been shown to be asymptotically unbiased and consistent in the quadratic mean sense for asymptotically independent/uncorrelated data (see [9]). Quadratic mean consistent means that

$$\lim_{n \to \infty} E\left(\hat{f}(x; n) - f(x)\right)^2 = 0. \tag{3.5}$$

One case of asymptotic independence used in [9] that is of interest to our problem is strong-mixing. Strong-mixing is now defined. Consider a continuous time random process $X(t)$. Let $\mathcal{F}_a^b = \sigma(X(t), a \leq t \leq b)$ denote the σ-algebra of events in $\mathcal{F}$ generated by the

46

random variables $\{X(t), \ a \leq t \leq b\}$, $-\infty \leq a \leq b \leq \infty$. The stationary process $X(t)$ is strong-mixing, if for $\tau \geq 0$.

$$\sup_{A \in \mathcal{F}^0_{-\infty}, \ B \in \mathcal{F}^\infty_\tau} |P[AB] - P[A]P[B]| = \alpha(\tau)$$

where

$$\lim_{\tau \to \infty} \alpha(\tau) = 0. \tag{3.6}$$

$\alpha(\tau)$ characterizes the mixing rate and is referred to as the mixing coefficient. The above definition basically states that two non-intersecting events A and B, which are asymptotically separated, are asymptotically independent. We assume that the data $X_1, X_2, \ldots, X_n$ are observations of the process $X(t)$ obtained by uniform sampling.

[9] shows that, if

(1) $K(\cdot)$ is a density, that is $\int_{-\infty}^{\infty} K(x)dx = 1$ and $K(x) \geq 0, \forall x$

(2) $\lim_{x \to \infty} K(x) = 0$

(3) $\sup_x K(x) < \infty$

(4) $\lim_{n \to \infty} h_n = 0$

(5) $\lim_{n \to \infty} nh_n = \infty$

(6) $\int_0^\infty [\alpha(\tau)]^q d\tau < \infty$, for $0 < q < \frac{1}{2}$

then

$$\lim_{n \to \infty} E\left[\hat{f}(x; n)\right] = f(x)$$

and

$$\lim_{n \to \infty} E\left(\hat{f}(x; n) - f(x)\right)^2 = 0. \tag{3.7}$$

This result is useful to our problem in Chapter 2, where we assumed that our process was m-dependent (i.e. X_k and X_l have known correlation for $|k - l| \leq m$, while X_k and X_l

are independent for $|k - l| > m$.) Since an m-dependent process satisfies (3.6), it is also a strong-mixing process. Therefore the results of [9] are useful for our problem.

For the bivariate kernel density estimators, vector observations of the form $X = (X^1, X^2)^T$ are required. Given the observations, $x_1, x_2, \ldots, x_n$, the kernel density estimate of the bivariate pdf $f(x)$ is obtained by

$$\hat{f}(\mathbf{x}; n) = \frac{1}{nh_n^2} \sum_{k=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{X_k}}{h_n}\right). \tag{3.8}$$

For independent identically distributed vectors, the estimator of (3.8) is also unbiased and strongly consistent [8]. That is, if

(1) $K(\cdot)$ is a density on $\mathbf{R}^2$

(2) $\lim_{||\mathbf{x}|| \to \infty} ||\mathbf{x}||^2 K(\mathbf{x}) = 0$

(3) $\sup_{\mathbf{x} \in \mathbf{R}^2} K(\mathbf{x}) < \infty$

(4) $\lim_{n \to \infty} h_n = 0$

(5) $\lim_{n \to \infty} nh_n = \infty$

(6) $\sum_{n=1}^{\infty} \exp(-\alpha nh_n) < \infty, \forall \alpha > 0$

$$\lim_{n \to \infty} E\left[\hat{f}(\mathbf{x}; n)\right] = f(\mathbf{x})$$

and

$$\lim_{n \to \infty} \hat{f}(\mathbf{x}; n) = f(\mathbf{x}). \tag{3.9}$$

3.2 Implementation of Kernel Density Estimators

To estimate our pdfs, we utilize the training data. Denote the estimates of the marginals as $\hat{f}_1(x; n)$ and $\hat{f}_0(x; n)$, for H_1 and H_0, respectively. Also denote the estimates of the

48

bivariate pdfs for samples X_1 and X_{j+1} by $\hat{f}_1^{(1,j+1)}(x,y;n)$ and $\hat{f}_0^{(1,j+1)}(x,y;n)$, for H_1 and H_0, respectively. The kernel density estimates of the marginals are obtained from the training data by

$$\hat{f}_1(x;N) = \frac{1}{NMh_N} \sum_{l=0}^{M-1} \sum_{k=0}^{N-1} K_a\left(\frac{x - \zeta_{l,k}^1}{h_N}\right)$$

and

$$\hat{f}_0(x;N) = \frac{1}{NMh_N} \sum_{l=0}^{M-1} \sum_{k=0}^{N-1} K_a\left(\frac{x - \zeta_{l,k}^0}{h_N}\right). \tag{3.10}$$

The bivariates are obtained by

$$\hat{f}_1^{(1,j+1)}(x,y;N-j) = \frac{1}{(N-j)Mh_{(N-j)}^2} \sum_{l=0}^{M-1} \sum_{k=0}^{N-j-1} K_b\left(\frac{x - \zeta_{l,k}^1}{h_{(N-j)}}, \frac{y - \zeta_{l,(k+j)}^1}{h_{(N-j)}}\right)$$

$$\hat{f}_0^{(1,j+1)}(x,y;N-j) = \frac{1}{(N-j)Mh_{(N-j)}^2} \sum_{l=0}^{M-1} \sum_{k=0}^{N-j-1} K_b\left(\frac{x - \zeta_{l,k}^0}{h_{(N-j)}}, \frac{y - \zeta_{l,(k+j)}^0}{h_{(N-j)}}\right). \tag{3.11}$$

We choose the kernel functions to be Gaussian:

$$K_a(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \tag{3.12}$$

and

$$K_b(x,y) = \frac{1}{2\pi} e^{-\frac{1}{2}(x^2+y^2)}. \tag{3.13}$$

The window width is set as

$$h_n = n^{-c}, \quad c \in (0,1). \tag{3.14}$$

Note that, in equations (3.10) and (3.11), we average over the M independent sample paths of the training data. For the bivariates, we utilize pairs of observations, $\zeta_{l,k}^i$ and $\zeta_{l,k+j}^i$, which are j samples apart, to estimate $\hat{f}_i^{(1,j+1)}(x,y;n)$ under hypothesis H_i.

Since the estimators cannot practically be implemented to estimate continuous functions, we implement them to estimate the pdfs at a discrete set of gridpoints. Denote these

49

gridpoints as $x_0, x_1, \ldots, x_{G-1}$, where G is the total number of points. So, using (3.10) and (3.11) we form the set of estimates

$$\hat{f}_1(x_i; N), \quad \hat{f}_0(x_i; N), \quad \text{for } i = 0, 1, \ldots, G-1 \qquad (3.15)$$

and

$$\hat{f}_1^{(1,j+1)}(x_i, x_l; N-j), \quad \hat{f}_0^{(1,j+1)}(x_i, x_l; N-j), \quad \text{for } i, l = 0, 1, \ldots, G-1. \qquad (3.16)$$

Using equations (3.15) and (3.16) the estimators can easily be implemented in a digital computer simulation. Some computers are now available with vector processing capabilities, which greatly decreases processing time. Equations (3.15) and (3.16) can be easily vectorized as

$$\begin{bmatrix} \hat{f}_i(x_0; N) \\ \hat{f}_i(x_1; N) \\ \vdots \\ \hat{f}_i(x_{(G-1)}; N) \end{bmatrix} = \frac{1}{\sqrt{2\pi} N M h_N} \sum_{j=0}^{M-1} \left(\left[\vartheta_{0,0}^i + \vartheta_{0,1}^i + \ldots + \vartheta_{0,(N-1)}^i \right] \right.$$

$$+ \left[\vartheta_{1,0}^i + \vartheta_{1,1}^i + \ldots + \vartheta_{1,(N-1)}^i \right] + \ldots$$

$$+ \left. \left[\vartheta_{(M-1),0}^i + \vartheta_{(M-1),1}^i + \ldots + \vartheta_{(M-1),(N-1)}^i \right] \right) \qquad (3.17)$$

50

and

$$
\begin{bmatrix}
\hat{f}_i^{(1,j+1)}(x_0, x_1; N-j) \\
\hat{f}_i^{(1,j+1)}(x_0, x_2; N-j) \\
\vdots \\
\hat{f}_i^{(1,j+1)}(x_0, x_{(G-1)}; N-j) \\
\hat{f}_i^{(1,j+1)}(x_1, x_1; N-j) \\
\hat{f}_i^{(1,j+1)}(x_1, x_2; N-j) \\
\vdots \\
\hat{f}_i^{(1,j+1)}(x_1, x_{(G-1)}; N-j) \\
\vdots \\
\hat{f}_i^{(1,j+1)}(x_{(G-1)}, x_1; N-j) \\
\hat{f}_i^{(1,j+1)}(x_{(G-1)}, x_2; N-j) \\
\vdots \\
\hat{f}_i^{(1,j+1)}(x_{(G-1)}, x_{(G-1)}; N-j)
\end{bmatrix}
=
$$

$$
\frac{1}{\sqrt{2\pi}(N-j)Mh_{(N-j)}^2} \sum_{j=0}^{M-1} \left(\left[\chi_{0,0}^i + \chi_{0,1}^i + \ldots + \chi_{0,(N-j-1)}^i \right] \right.
$$

$$
+ \left[\chi_{1,0}^i + \chi_{1,1}^i + \ldots + \chi_{1,(N-j-1)}^i \right] + \ldots
$$

$$
\left. + \left[\chi_{(M-1),0}^i + \chi_{(M-1),1}^i + \ldots + \chi_{(M-1),(N-j-1)}^i \right] \right) \tag{3.18}
$$

where we have defined

$$
\vartheta_{m,l}^i =
\begin{bmatrix}
\exp\left\{ -\frac{1}{2h_N^2}\left(x_0 - \zeta_{m,l}^i \right)^2 \right\} \\
\exp\left\{ -\frac{1}{2h_N^2}\left(x_1 - \zeta_{m,l}^i \right)^2 \right\} \\
\vdots \\
\exp\left\{ -\frac{1}{2h_N^2}\left(x_{(G-1)} - \zeta_{m,l}^i \right)^2 \right\}
\end{bmatrix}
\tag{3.19}
$$

51

and

$$\chi^i_{m,l} = \begin{bmatrix} \exp\left\{-\frac{1}{2h^2_{(N-j)}}\left(\left(x_0 - \zeta^i_{m,l}\right)^2 + \left(x_0 - \zeta^i_{m,(l+j)}\right)^2\right)\right\} \\ \exp\left\{-\frac{1}{2h^2_{(N-j)}}\left(\left(x_0 - \zeta^i_{m,l}\right)^2 + \left(x_1 - \zeta^i_{m,(l+j)}\right)^2\right)\right\} \\ \vdots \\ \exp\left\{-\frac{1}{2h^2_{(N-j)}}\left(\left(x_0 - \zeta^i_{m,l}\right)^2 + \left(x_{(G-1)} - \zeta^i_{m,(l+j)}\right)^2\right)\right\} \\ \exp\left\{-\frac{1}{2h^2_{(N-j)}}\left(\left(x_1 - \zeta^i_{m,l}\right)^2 + \left(x_0 - \zeta^i_{m,(l+j)}\right)^2\right)\right\} \\ \exp\left\{-\frac{1}{2h^2_{(N-j)}}\left(\left(x_1 - \zeta^i_{m,l}\right)^2 + \left(x_1 - \zeta^i_{m,(l+j)}\right)^2\right)\right\} \\ \vdots \\ \exp\left\{-\frac{1}{2h^2_{(N-j)}}\left(\left(x_1 - \zeta^i_{m,l}\right)^2 + \left(x_{(G-1)} - \zeta^i_{m,(l+j)}\right)^2\right)\right\} \\ \vdots \\ \exp\left\{-\frac{1}{2h^2_{(N-j)}}\left(\left(x_{(G-1)} - \zeta^i_{m,l}\right)^2 + \left(x_0 - \zeta^i_{m,(l+j)}\right)^2\right)\right\} \\ \exp\left\{-\frac{1}{2h^2_{(N-j)}}\left(\left(x_{(G-1)} - \zeta^i_{m,l}\right)^2 + \left(x_1 - \zeta^i_{m,(l+j)}\right)^2\right)\right\} \\ \vdots \\ \exp\left\{-\frac{1}{2h^2_{(N-j)}}\left(\left(x_{(G-1)} - \zeta^i_{m,l}\right)^2 + \left(x_{(G-1)} - \zeta^i_{m,(l+j)}\right)^2\right)\right\} \end{bmatrix} . \tag{3.20}$$

3.3 Numerical Results

In this section, the performance of memoryless quantizer discriminators based upon estimated pdfs was evaluated via computer simulation. Equations (3.17) through (3.20) were implemented in a Convex 210 mini-super computer capable of vector processing. The training data introduced in Chapter 1 was fed into the simulations to obtain estimates of the necessary pdfs. These pdfs were then integrated via a Simpson's integration to result in the cdfs required to derive the quantization functions. Next we consider the consistency of the estimators.
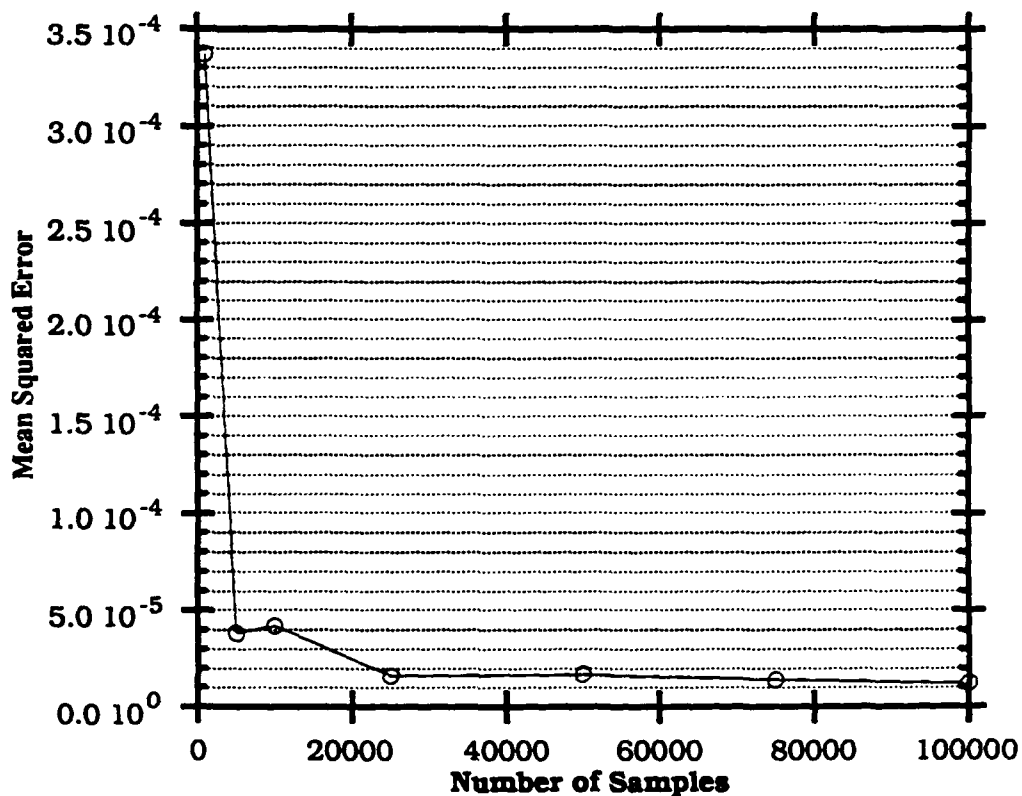
52

Figure 3.1: Mean Squared Error for an Estimated Marginal Density Function

Using the estimator simulations, the consistency of the marginal estimators were checked for a Rayleigh density. Equations (2.52) through (2.54) were utilized to generate correlated data sequences with a Rayleigh marginal density. These data were then fed into estimators. The densities were evaluated at 65 gridpoints over the interval (0.02,12). The lower limit, 0.02, was placed just below the minimum observed sample, and the upper limit, 12, was set just above the maximum observed sample. The constant c in equation (3.14) was set to 0.1. Figure 3.1 shows the mean squared error as a function of the number of
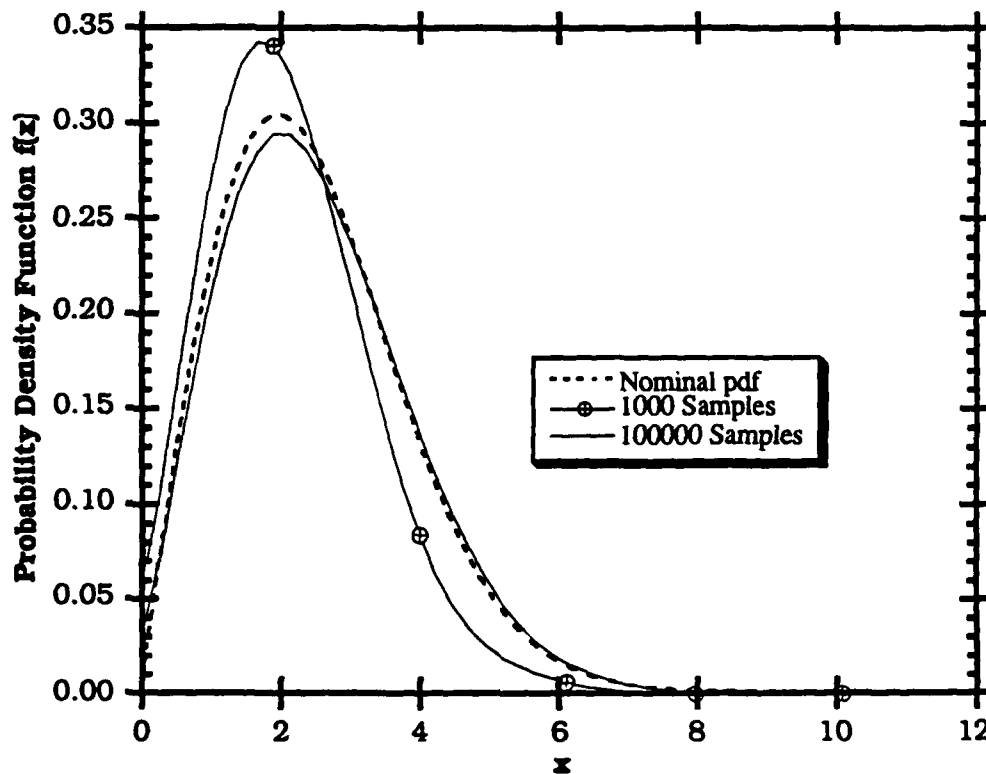
53

Figure 3.2: Nominal and Estimated Marginal Probability Density Functions

samples used by the estimator. Notice that, as the number of samples increases, the mean squared error decreases (apparently exponentially towards zero). This result supports the notion of quadratic mean consistency of the marginal density estimator for correlated data. Due to computer processing limitations, the consistency of bivariate pdf estimates could not be checked. Figure 3.2 depicts the nominal density, the estimated density for 1,000 samples, and the estimated density for 100,000 samples. Clearly the estimate for 100,000 samples is closer to the nominal density than the estimate for 1,000 samples.

With some confidence that the estimators produce reasonable estimates of the pdfs, we now consider our discrimination cases. Using the training data introduced in Chapter 1 and the estimator simulations, estimates of the marginal and bivariate pfds for each hypothesis of Case 1 and Case 2 were formed. These were computed over the interval (0.02,12) at 33 gridpoints for each case. The interval was chosen in the same manner as described in the preceding paragraph. Figures 3.3 and 3.4 depict the nominal and estimated marginal pdfs for each hypothesis for Case 1 and Case 2, respectively. For both the marginal and bivariate estimators, the constant c in (3.14) was set to 0.1. The bivariate pdfs in equation (3.18) were computed for $j = 1, 2, \ldots, 30$, for H_0, and $j = 1, 2, \ldots, 150$, for H_1. The choices of the maximum j were due to computation restrictions. A better method of choosing j would have been to estimate the decorrelation time under each hypothesis and use those values for the maximum choice of j.

After the marginal pdf estimates were obtained, cdfs were computed via Simpson's integration. These bivariate pdfs were summed over j for each hypothesis and then integrated in two dimensions (also using a Simpson's integration) to yield the necessary sums of joint cdfs required for the optimum quantization function.

Figures 3.5 and 3.6 show the quantization functions computed for Case 1 and Case 2, respectively using the expressions given in Chapter 2. Comparing these to the 128-level uniform quantizers from Chapter 2 (see Figures 2.3 and 2.7), some similarities can be noted. For the quantizer of Case 1 derived from estimated pdfs, note that the *drop* for small values of x is still present. The sharp incline for large values of x is still present for values of x between 10.5 and 11.5. The function is relatively flat for values of x between 1 and 10.5. However, note the drop for values of x for the last two quantization levels. This drop may be attributed to the inaccuracies of the estimates of the pdfs in the *tails* of the densities. The
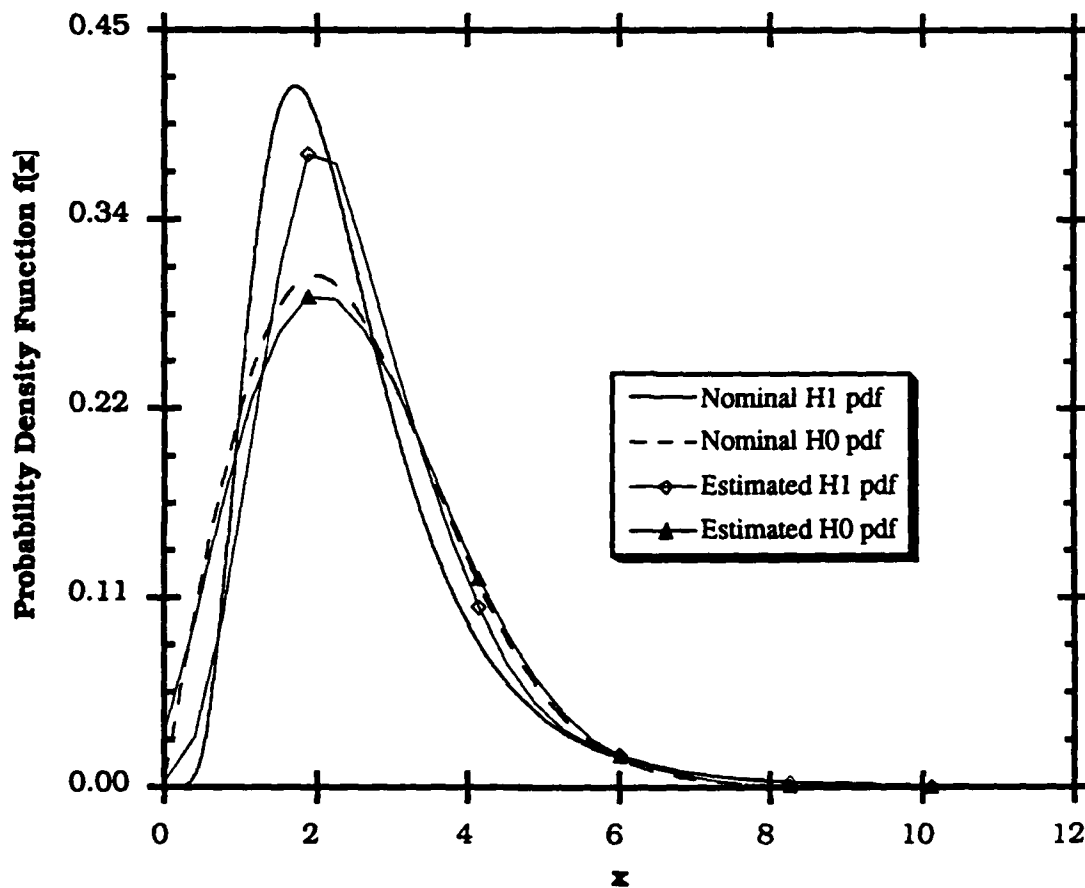
55

Figure 3.3: Nominal and Estimated Marginal Probability Density Functions for Case 1

Case 2 quantizer also has inaccuracies out at its *tails*. Table 3.1 lists the performance of the memoryless quantizer discriminators using the functions of Figures 3.5 and 3.6. The thresholds a and b were set for desired probabilities of error of 10^{-3}. Despite having low probabilities of error, these discriminators performed poorly when the average sample size was considered. The Case 1 discriminator required an average of 3400 samples to make
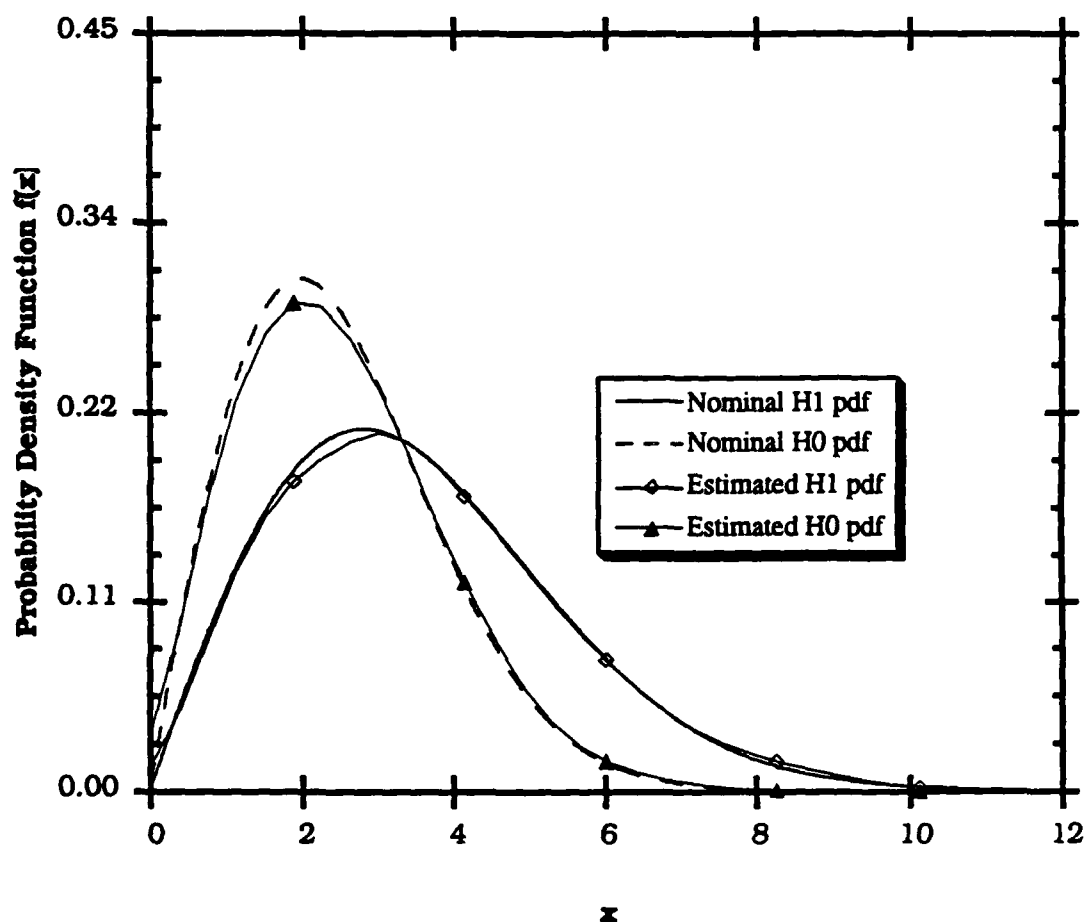
Figure 3.4: Nominal and Estimated Marginal Probability Density Functions for Case 2

a decision, while the discriminator for Case 2 required an average of 4270 samples for a decision. The quantizers from Chapter 2, which were derived from nominal pdfs, required an average number of samples of 782 and 2660 for Case 1 and Case 2, respectively, for comparable probabilities of error.

57
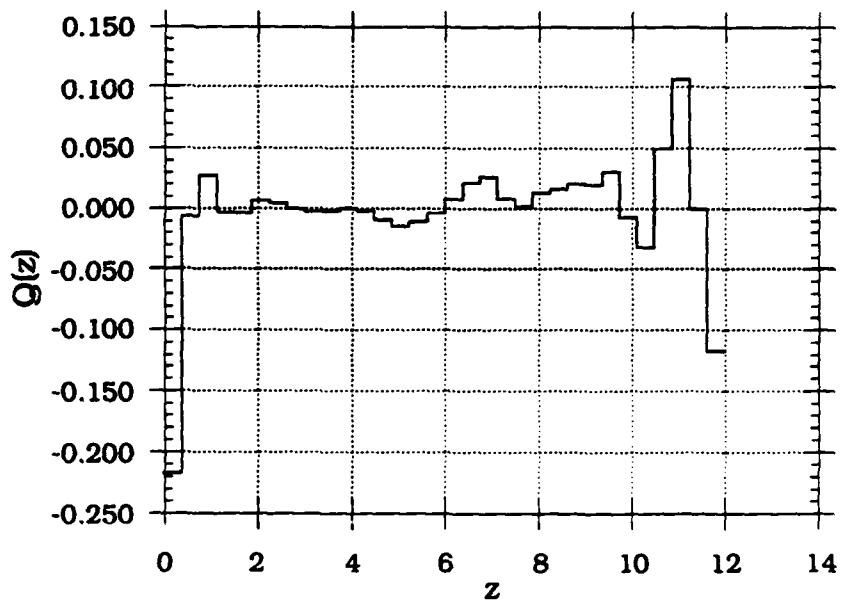
Figure 3.5: 32-Level Uniform Quantizer Function for Case 1
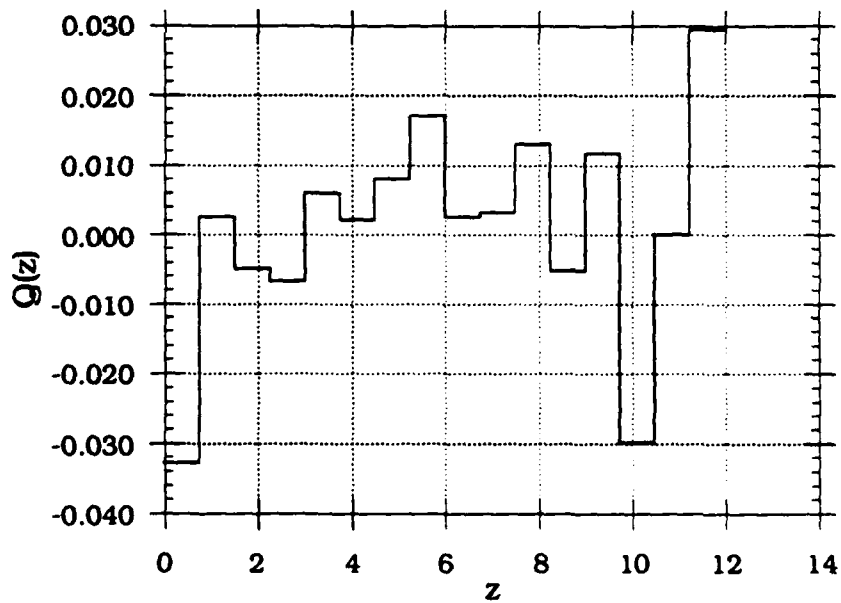
Figure 3.6: 16-Level Uniform Quantizer Quantizer Function for Case 2

58

Case	Quantization Levels	P_f	P_d	$E[n]$
Case 1	32	0.039	0.995	3400
Case 2	16	0.002	1	4270

Table 3.1: Results for Case 1 and Case 2 Quantizers for Desired $P_f = P_m = 10^{-3}$

Chapter 4

Neural Network Discriminators

The memoryless discriminators derived in the preceding sections can be easily implemented in practice because they only require estimating first and second order probability density functions of the observed process under H_1 and H_0. These memoryless discriminators use nonlinear functions of one variable, with the form $Q(x)$, which are chosen to maximize a performance measure and are derived from first and second order probability density functions. The nonlinearities are used in the test statistic of the discriminators as follows:

$$T_n = \sum_{j=1}^{n} Q(Z_j).$$ (4.1)

Similar nonlinear functions, which have memory and have the form

$$\gamma(x_1, x_2, \ldots, x_K),$$

could be derived to optimize the same performance measures. Test statistics of the form

$$T_n = \sum_{j=1}^{n} \gamma(Z_{K-1+j}, Z_{K-2+j}, \ldots, Z_j)$$ (4.2)

could be computed using nonlinear functions of K variables. These functions, however, would require higher-order probability density functions to be estimated(see [10]). In practice, only first and second order probability density functions can be easily obtained with reasonable accuracy for a small amount of training data.

In this section, we restrict the class of nonlinearities, $\gamma(x_1, x_2, \ldots, x_K)$, to have a maximum absolute value - not an unreal limitation in a real system. Then we use a perceptron neural network to form our nonlinearity and the back-propagation to minimize our performance measure.

4.1 Perceptron Neural Networks

Perceptron neural networks are interconnected layers of simple processing units called perceptrons. A perceptron is illustrated in Figure 4.1. The perceptron takes an input vector $\mathbf{x} = (x_0, x_1, \ldots, x_{K-1})^T$ and a weighting vector $\mathbf{w} = (w_0, w_1, \ldots, w_{K-1})^T$ and forms a dot product

$$\sum_{i=0}^{K-1} x_i w_i = \mathbf{x}\mathbf{w}^T. \tag{4.3}$$

From the dot product, an offset value θ is subtracted to get the result $y = \mathbf{x}\mathbf{w}^T - \theta$; y is then passed through a sigmoidal nonlinearity of the form

$$f(y) = \frac{1}{1 + e^{-y}}. \tag{4.4}$$

The sigmoidal function is shown in Figure 4.2. Note that, throughout this thesis, we use the term perceptron and node interchangeably. We also refer to the offset value θ as the node offset value.
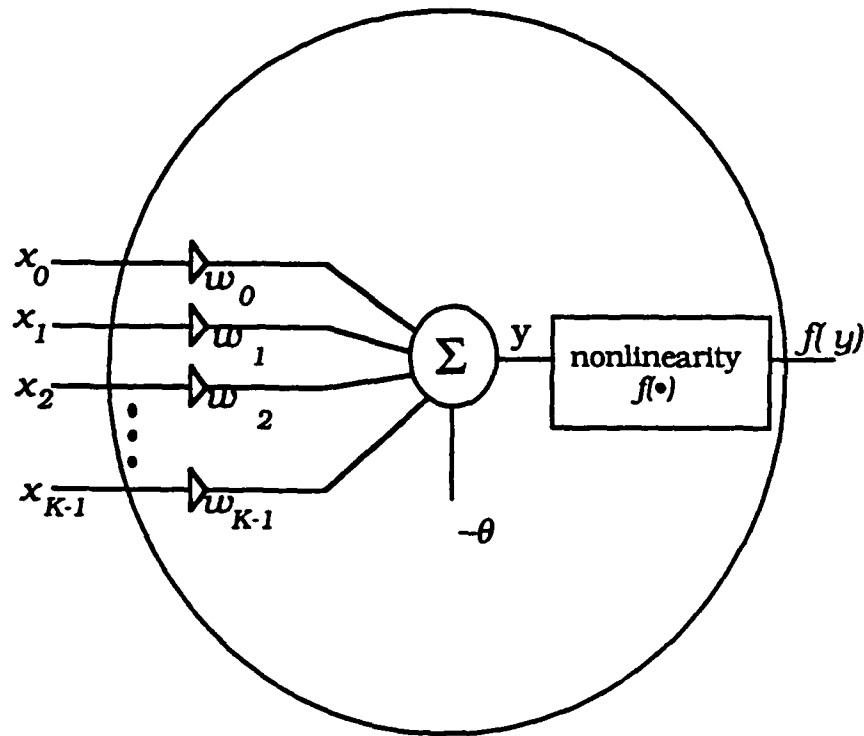
Figure 4.1: A Perceptron

To gain an understanding of what a perceptron does, consider a perceptron with two inputs, x_0 and x_1. This implies that the perceptron has two weights, w_0 and w_1. To simplify the analysis, replace the sigmoidal curve with a hard quantizer

$$q(x) = \begin{cases} 1, & \text{if } x \geq 0; \\ 0, & \text{otherwise.} \end{cases} \qquad (4.5)$$

So the output of the perceptron is either a 0 or 1. Figure 4.3 shows the x_0, x_1 plane. The perceptron with a hard quantizer actually forms two decision regions separated by the line:

$$x_1 = \frac{-w_0}{w_1} x_0 + \frac{\theta}{w_1}. \qquad (4.6)$$

(x_0, x_1) pairs on one side of the line result in a perceptron output of 1, while pairs on the other side of the line result in an output of 0. If the perceptron had K inputs, the

62
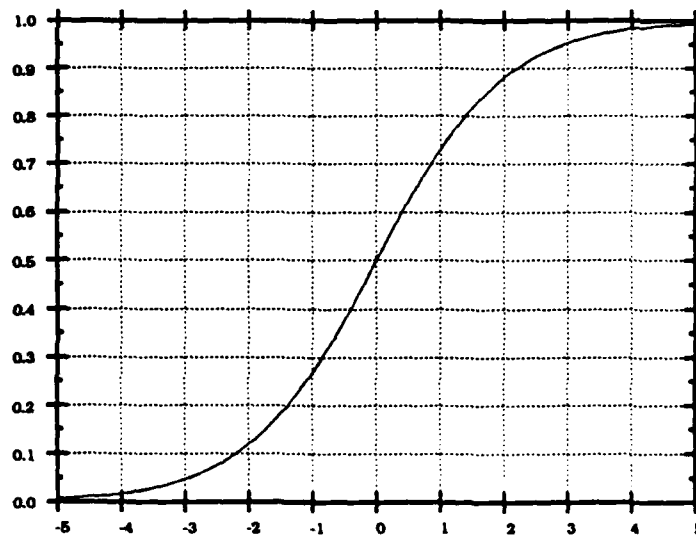
Figure 4.2: A Sigmoid Nonlinearity

decision region would become a hyperplane in $\mathbf{R}^K$. Note that the location of the hyperplane separating the decision regions is determined only by the weights w and the offset value θ.

If the hard limiter above is replaced by the sigmoidal nonlinearity, then the decision regions become *soft*. That is, input vectors near the hyperplane have outputs that are near $\frac{1}{2}$. Input vectors taken farther away from the hyperplane have outputs that approach 0 or 1, depending on which side of the hyperplane they lie.

More complex decision regions can be formed by utilizing multiple hyperplanes. Decision regions can be formed by using a perceptron to form each hyperplane of a complex region. The output of each perceptron can then be fed into an AND gate — or, better yet, another perceptron with weights and an offset appropriately set to simulate an AND function. This leads to the concept of multi-layer perceptron neural networks.

63
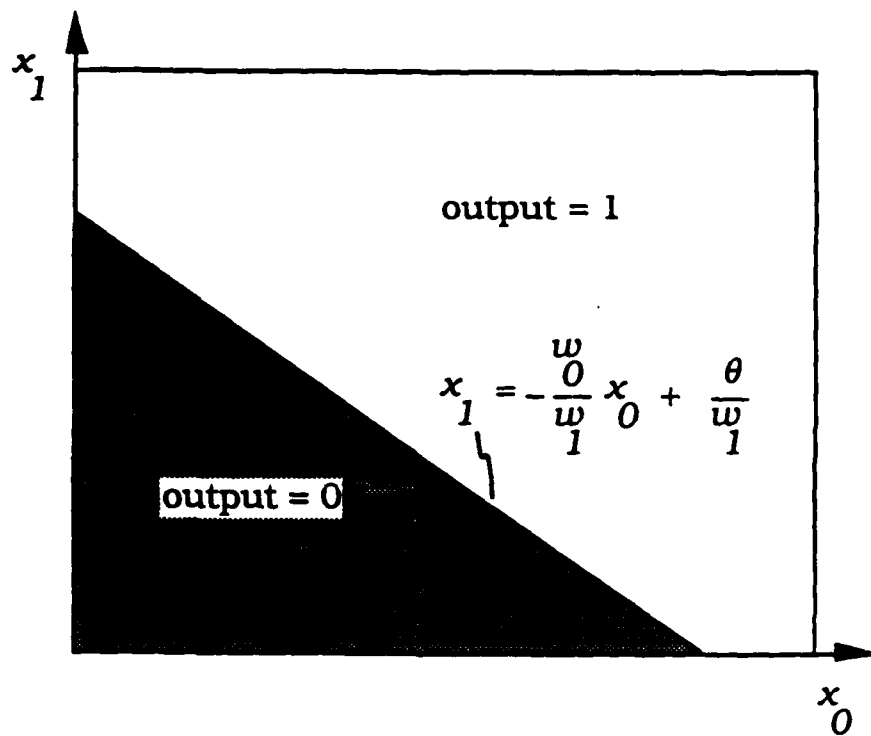
Figure 4.3: Decision Space for a Perceptron

Multiple-layer perceptron neural networks take the outputs of the perceptrons on a layer and use them as inputs to the next higher level of perceptrons (see Figure 4.4.) Networks of this type are usually called feed-forward neural networks. As demonstrated in the above discussion. a single perceptron can only divide the decision space with a hyperplane. But it has been shown that a two-layer perceptron neural network can form any convex decision region [11]. A convex region is a region from which any two points can be connected by a line which lies entirely within the region. A third layer of nodes can allow the network to form any arbitrary decision region [11] (assuming enough nodes are allocated to the correct layers.)

To form a desired decision region. the weights and node offset values for each node in each layer of a neural network must be specified. This would be a difficult task even if the decision region were known. But. for many problems. the decision region is not known because

64
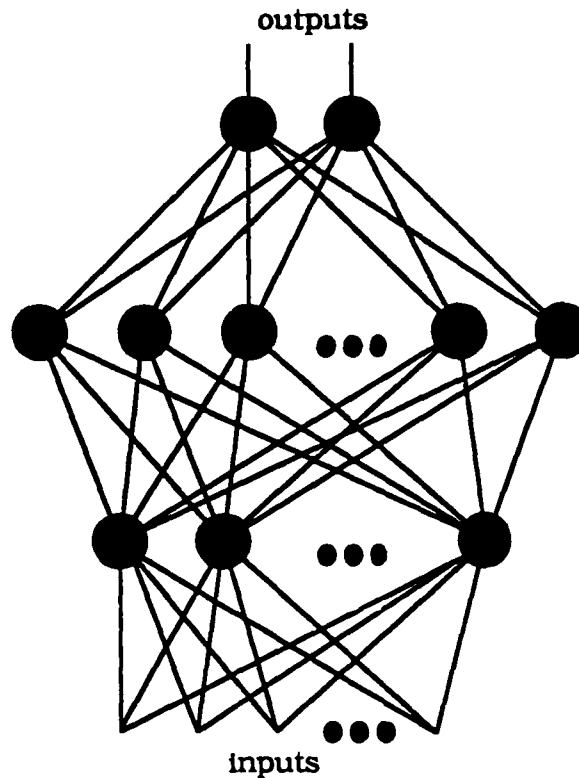
Figure 4.4: A Multiple-Layer Perceptron

the statistical models of the data are not known. Training algorithms to form appropriate decision regions exist for perceptron neural networks. These algorithms typically present the training data to the network along with a desired response and the network weight values and node offset values are adjusted to force the actual network response towards the desired response. One such algorithm is the back-propagation algorithm. The back-propagation algorithm is a gradient search method (searching over w and θ), which minimizes the square error of the neural network outputs [12]. Note that the back-propagation algorithm requires the nodes to have sigmoidal nonlinearities. (See Appendix B for a description of the back-propagation algorithm.)

4.2 The Neural Network Sequential Discriminator

As mentioned in the introduction to this chapter, optimal nonlinearities $\gamma(x_1, x_2, \ldots, x_K)$ could be derived for use in a discriminator using the test statistic $T_n = \sum_{j=K}^n \gamma(Z_{K-1+j}, Z_{K-2+j}, \ldots, Z_j)$. [10] derived one step memory nonlinearities for use in a block discrimination scheme to form the test statistic $T_n = \sum_{j=2}^n g(Z_{j-1}, Z_j)$. However, in general these nonlinearities require knowledge (or estimation) of the pdfs of the data of degrees higher than two. Nonlinearities in [10] require pdfs of the data of the fourth degree under each hypothesis.

We now consider a suboptimal approach that utilizes perceptron neural networks and yields excellent performance. We start by defining the structure of our sequential discriminator. Our discriminator utilizes a test statistic of the form

$$T_n = \sum_{j=K}^n \gamma(Z_{K-1+j}, Z_{K-2+j}, \ldots, Z_j).$$

A two threshold test is implemented, using the constants $\tilde{a}$ and $\tilde{b}$. So, upon obtaining a new data sample, Z_n, the discriminator computes the test statistic T_n. If T_n reaches $\tilde{b}$, then the discriminator chooses H_1 and terminates the test. If T_n drops to $\tilde{a}$, then the test terminates and the discriminator chooses H_0. If T_n lies between $\tilde{a}$ and $\tilde{b}$, then another sample Z_{n+1} is obtained, T_{n+1} is computed, and the entire test is repeated. This process continues until either a decision is made, or the N-th sample is reached. Upon obtaining the N-th sample, T_N is computed and a one-threshold test is performed. Obviously T_{K-1} is initialized to a value in the interval $(\tilde{a}, \tilde{b})$.

We now restrict the class of nonlinearities of the form $\gamma(x_1, x_2, \ldots, x_K)$ to have a

range with maximum absolute value of r. That is, we require

$$|\gamma(x_1, x_2, \ldots, x_K)| \leq r \text{ for all possible values of}$$

$$\text{the } K - \text{tuple } (x_1, x_2, \ldots, x_K). \tag{4.7}$$

This restriction leads to a suboptimal discriminator, but allows us to obtain a solution.

Now assuming that r, $\tilde{a}$, and $\tilde{b}$ are all specified constants, the structure of our test allows us to scale r, $\tilde{a}$, and $\tilde{b}$ to get a test with a nonlinearity with a maximum absolute value of 1. The newly scaled thresholds shall be denoted as a and b. This rescaling of r to 1 allows us to utilize a perceptron neural network with a sigmoid nonlinearity on its nodes in the following paragraphs.

To find the optimal nonlinearity within our class, we first consider the optimal paths that the test statistic T_n can take under each hypothesis. By optimal path we mean the path that T_n should take to minimize the number of samples needed to cross the correct threshold under the appropriate hypothesis. Obviously the quickest path to reach a threshold is when the discriminator takes a step of magnitude 1 in the appropriate direction upon obtaining each new data sample. That is, for each new data sample, the test statistic under H_1 is incremented by $+1$, while the test statistic under H_0 is incremented by -1. If the data sequence $\{Z\}_{i=1}^{\infty}$ is obtained by sampling some continuous process with a uniform sampling period T, then the optimal path for T_n would lie on a straight line with slope $+\frac{1}{T}$ for H_1 and slope $-\frac{1}{T}$ for H_0. Figure 4.5 depicts these paths. Thus, for an ideal discriminator, (that is a discriminator which never makes mistakes and always uses the minimum number of samples possible), the statistics of the nonlinearity should be

$$E_1[\gamma(Z_1, Z_2, \ldots, Z_K)] = 1$$

67
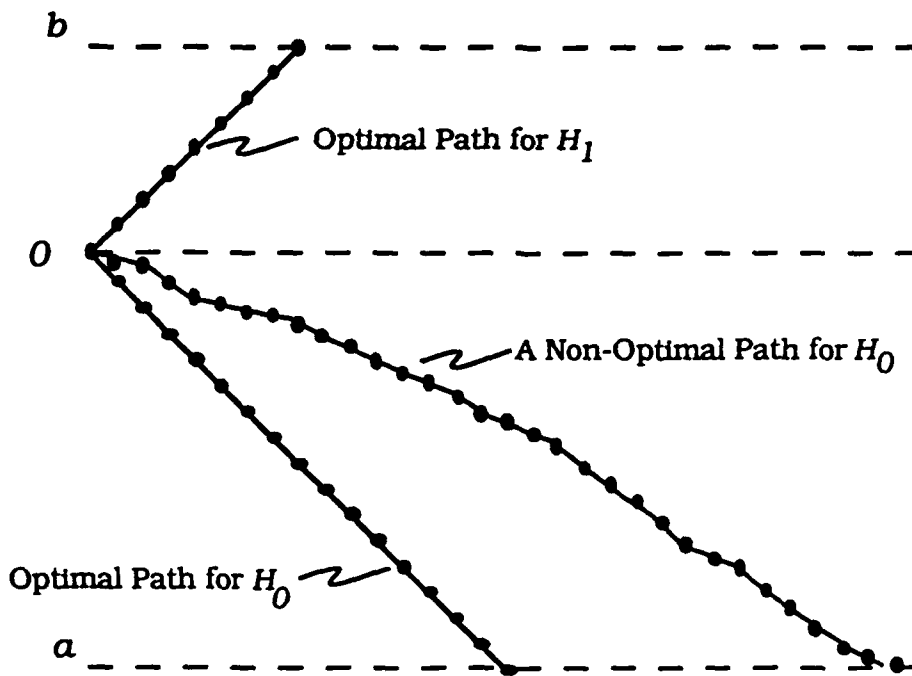
Figure 4.5: Optimal Paths for T_n under Constraint of Maximum Slope

$$E_0[\gamma(Z_1, Z_2, \ldots, Z_K)] = -1$$

$$Var_1[\gamma(Z_1, Z_2, \ldots, Z_K)] = 0$$

$$Var_0[\gamma(Z_1, Z_2, \ldots, Z_K)] = 0 \qquad (4.8)$$

We cannot expect a real discriminator to achieve the statistics of the above equations. However, we can choose the nonlinearity to minimize some performance measure, such as a mean squared error criterion of γ about its desired values. We show that the back-propagation algorithm can be used to minimize a related mean squared error criterion.

We form a nonlinearity by constructing a perceptron neural network with K inputs $x_1, x_2, \ldots, x_K$ and two outputs which are functions of the inputs (and the weights/offsets for each perceptron in the network), $o^1(x_1, x_2, \ldots, x_K)$ and $o^0(x_1, x_2, \ldots, x_K)$. To simplify

68

the notation we denote the output nodes as o^1 and o^0. During training (see Section 4.3), the desired values of the output nodes are $(1\,0)$ for inputs from H_0 and $(0\,1)$ for inputs from H_1. Our notation $(x\,y)$ implies that $o^0 = x$ and $o^1 = y$. The nonlinearity, $\gamma(x_1, x_2, \ldots, x_K)$, is formed by

$$\gamma(x_1, x_2, \ldots, x_K) = o^1(x_1, x_2, \ldots, x_K) - o^0(x_1, x_2, \ldots, x_K),$$

or with simplified notation,

$$\gamma = o^1 - o^0. \tag{4.9}$$

We wish the nonlinearity to be such that γ is close to values of 1 for inputs from H_1 and -1 for inputs from H_0. We choose a performance measure which involves the mean squared error of o^1 and o^0 about their desired values for each hypothesis:

$$\tilde{S}_5 = E_0\left[(1 - o^0)^2 + (0 - o^1)^2\right] + E_1\left[(0 - o^0)^2 + (1 - o^1)^2\right]. \tag{4.10}$$

We would like the weight and node offset values of each perceptron in our neural network to have values which minimize equation (4.10).

We now try to relate this performance measure via an intuitive argument to performance measures from the previous chapters. Comparing (4.10) to our performance measure $\tilde{S}_3$ from Chapter 2, we notice that they are similar. Recall that

$$\tilde{S}_3 = \frac{[\mu_1 - \mu_0]^2}{[\sigma_1^2 + \sigma_0^2]}. \tag{4.11}$$

Effectively, by maximizing equation (4.11), the expected values of the nonlinearity are separated, while the variances about the expected values are minimized. Minimizing our performance measure $\tilde{S}_5$ fixes the difference of the desired values, and minimizes a second order moment. Both performance measures try to keep the expected values separated while minimizing a second order moment about (or near) the expected value.

69

Recall that the back-propagation algorithm [12] is a gradient descent algorithm which minimizes the performance measure

$$\tilde{E} = \frac{1}{2} \sum_p \sum_j (t_p^j - o_p^j)^2 \tag{4.12}$$

where t_p^j is the desired output for node j associated with input pattern p, and o_p^j is the actual value of the output node j associated with input pattern p. Suppose we have P K-tuples from each hypothesis available for training the neural network. We also have $j = 0, 1$ for the two output nodes o^1 and o^0, respectively. We can rewrite (4.12) as

$$\tilde{E} = \frac{1}{2} \sum_{p=0}^{P-1} \{(1 - o^0)^2 + (0 - o^1)^2\} + \frac{1}{2} \sum_{p=P}^{2P-1} \{(0 - o^0)^2 + (1 - o^1)^2\} \tag{4.13}$$

where the first sum is over the H_0 training patterns and the second sum is over the H_1 training patterns. The problem of minimizing $\tilde{E}$ is equivalent to minimizing $\tilde{E}$ scaled by a constant. Thus minimizing (4.13) is equivalent to minimizing

$$\frac{2}{P}\tilde{E} = \frac{1}{P} \sum_{p=0}^{P-1} \{(1 - o^0)^2 + (0 - o^1)^2\} + \frac{1}{P} \sum_{p=P}^{2P-1} \{(0 - o^0)^2 + (1 - o^1)^2\}. \tag{4.14}$$

Now as $P \longrightarrow \infty$ we have

$$\frac{2}{P}\tilde{E} \longrightarrow E_0 \left[(1 - o^0)^2 + (0 - o^1)^2\right] + E_1 \left[(0 - o^0)^2 + (1 - o^1)^2\right] = \tilde{S}_s, \tag{4.15}$$

which is our desired performance measure. Consequently, the back-propagation algorithm is a reasonable algorithm to be utilized for our perceptron neural network nonlinearity.
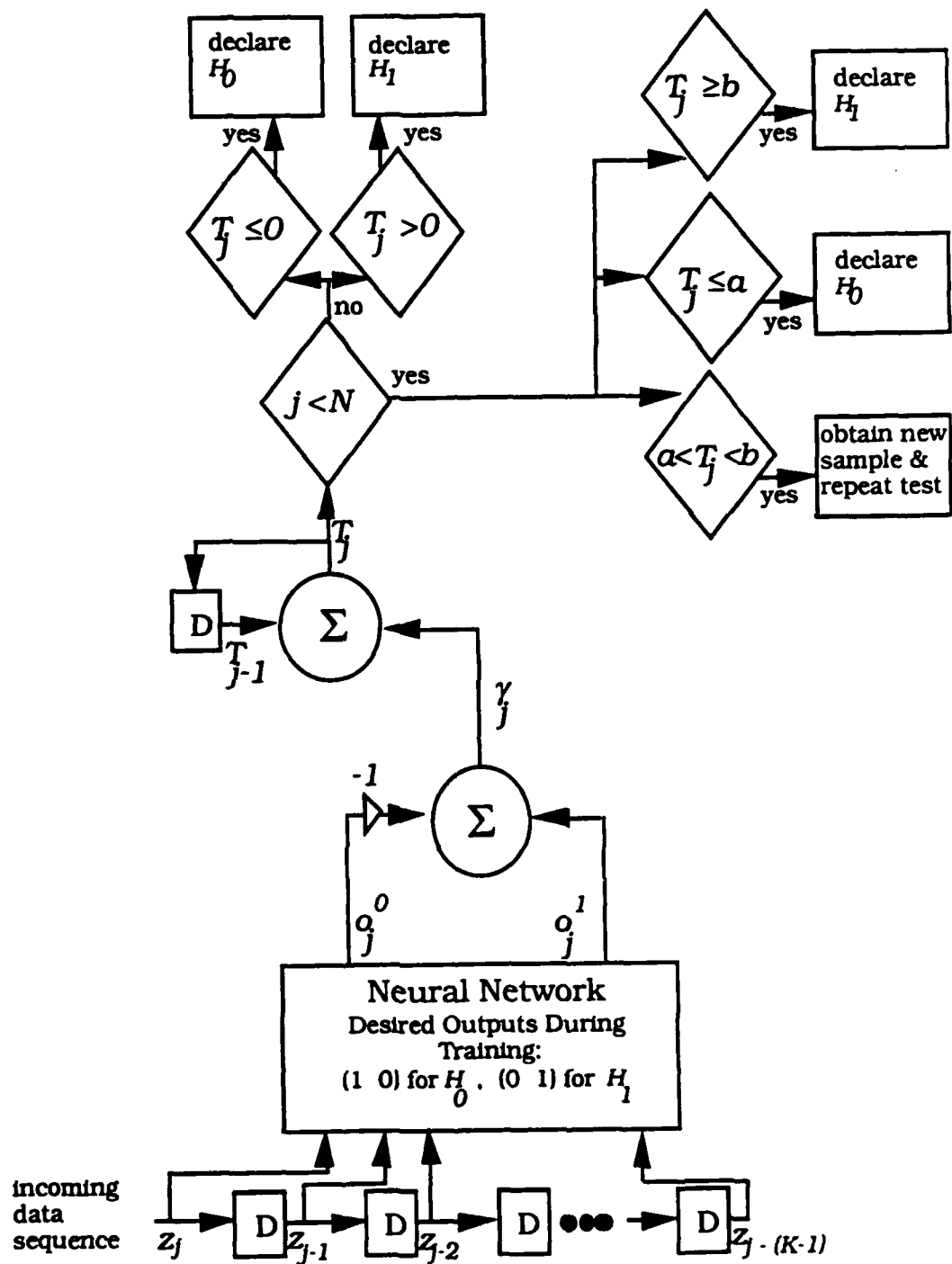
Figure 4.6: Sequential Neural Network Discriminator

Using this nonlinearity we can form the test statistic

$$T_n = \sum_{j=1}^{n} \gamma(Z_{K-1+j}, Z_{K-2+j}, \ldots, Z_j). \qquad (4.16)$$

Figure 4.6 shows the implementation of our test. The incoming data samples are passed through a tapped delay line. The K taps are the inputs to the perceptron neural network. The difference of the outputs of the neural network is formed and added to the test statistic T_j. The notation subscripts j correspond to the values associated with the jth data sample. The sample number j is compared to N. If j reaches N, then a one threshold test is performed (in this figure the threshold is 0.) If j is less than N, then a two threshold test is performed.

4.3 Neural Network Training Phase

The neural network used in our sequential discrimination scheme operates on K-tuples $(Z_{K-1+j}, Z_{K-2+j}, \ldots, Z_j)$, which are formed from the incoming data sequence $\{Z_j\}_{j=1}^{\infty}$ on which the discriminator must make a decision of H_1 or H_0. The neural network may have two or three layers of nodes, but it will always have two output nodes on the output layer. Figures 4.7 and 4.8 depict the two possible forms of the neural network considered in this thesis.

The neural network is trained using the back-propagation algorithm and the training data set. The training data set consists of M sample paths of length N from each hypothesis. These training data are defined as $\zeta_{m,j}^i$, where $i = 0, 1$ denotes the hypothesis (H_1 or H_0), $m = 0, 1, \ldots, M - 1$ denotes the sample path number, and $j = 0, 1, \ldots, N - 1$ denotes the
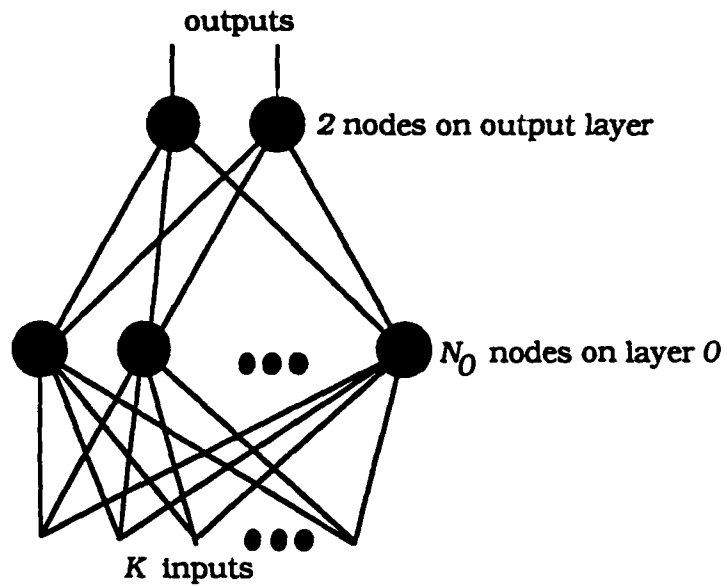
72

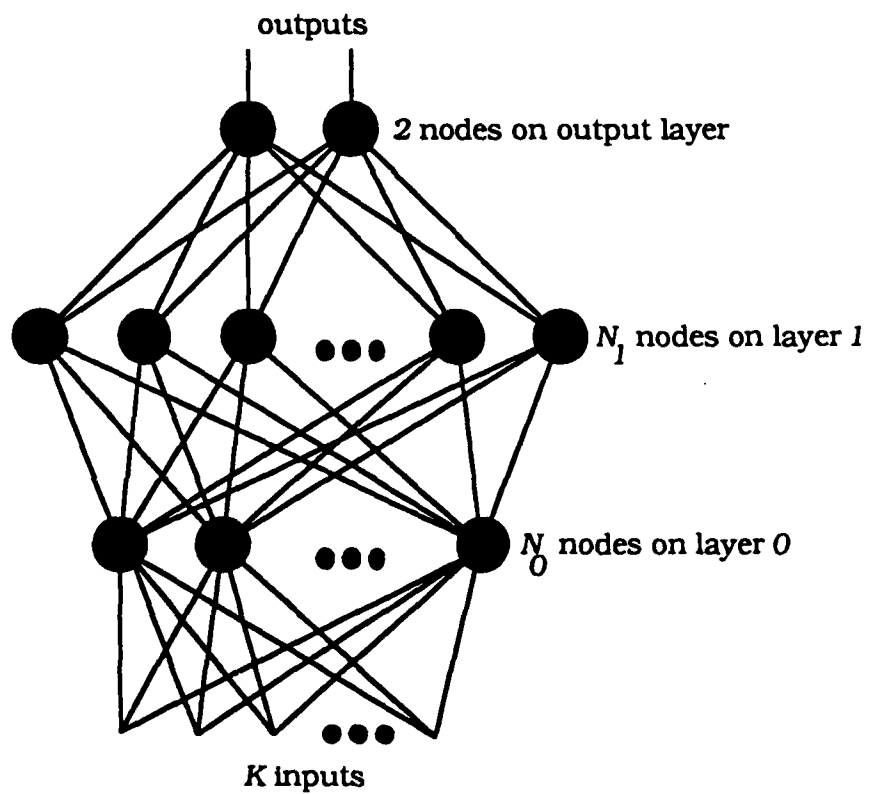**Figure 4.7: A Two Layer Perceptron Neural Network**

Figure 4.8: A Three Layer Perceptron Neural Network

sample number. The desired responses for the neural network are $(1\ 0)$ for H_0 and $(0\ 1)$ for H_1. Our notation $(a\ b)$ implies that the output node 0 outputs a and the output node 1 outputs b.

The training process proceeds as follows: The first K-tuple from the first sample path from H_0, $(\zeta_{0,0}^0, \zeta_{0,1}^0, \ldots, \zeta_{0,K-1}^0)$, is presented to the neural network inputs. The back-propagation algorithm is performed using $(1\ 0)$ as the desired output. Then the first K-tuple from the first sample path from H_1, $(\zeta_{0,0}^1, \zeta_{0,1}^1, \ldots, \zeta_{0,K-1}^1)$, is presented to the neural network inputs. Back-propagation is performed with the desired response of $(0\ 1)$. Then the second K-tuple from the first H_0 sample path, $(\zeta_{0,1}^0, \zeta_{0,2}^0, \ldots, \zeta_{0,K}^0)$, is presented to the network for back-propagation. Then the second K-tuple from the first H_1 sample path, $(\zeta_{0,1}^1, \zeta_{0,2}^1, \ldots, \zeta_{0,K}^1)$, is presented to the network for back-propagation. When all the K-tuples (of ordered adjacent samples) from the first sample path for H_0 and H_1 have been exhausted, the process is repeated for the remaining until they have all been exhausted. Then the entire process is repeated until all sample paths have been presented to the network L times.

4.4 Determination of Thresholds a and b

The discriminators in Section 4.3 are *trained* to minimize the squared error of the desired outputs, o^1 and o^0, under each hypothesis. In effect, the average slope of the path of the test statistic is forced towards $+1$ for H_1 and -1 for H_0. In this section we suggest a scheme for determining practical values of the thresholds a and b. Intuitively, as the thresholds are moved farther away from zero, the probabilities of error decrease, while the average sample

74

size increases. Therefore, by constraining

$$b = -a > 0 \qquad (4.17)$$

to correspond to the largest of the desired α and β probably will not affect the performance of the discriminator; this assumes that the desired values of α and β are small. We also impose the following constraint on the maximum value of b:

$$b < B. \qquad (4.18)$$

We force our test to begin with $T_{K-1} = 0$. By utilizing the training data $\zeta_{m,j}^i$ and the neural network discriminator, we generate the output data sequences

$$T_{m,j}^i \qquad (4.19)$$

where $i = 0, 1$ denotes the hypothesis, $m = 0, 1, \ldots, M-1$ denotes the sample path number, and where $j = K, K+1, \ldots, N-1$ denotes the test statistic number. Define a new set of functions $e_m^i(b)$ by

$$e_m^i(b) = \begin{cases} 0, & \text{if discriminator with thresholds } -b \text{ and } b \text{ chooses } H_i \text{ for path } T_m^i \\ 1, & \text{otherwise.} \end{cases} \qquad (4.20)$$

Now using the functions $e_0^1(b), e_1^1(b), \ldots, e_{M-1}^1(b)$ and $e_0^0(b), e_1^0(b), \ldots, e_{M-1}^0(b)$ define

$$\hat{e}^1(b) = \frac{1}{M} \sum_{m=0}^{M-1} e_m^1(b)$$

$$\hat{e}^0(b) = \frac{1}{M} \sum_{m=0}^{M-1} e_m^0(b). \qquad (4.21)$$

Thus $\hat{e}^i(b)$ is the average number of errors for thresholds $-b$ and b under hypothesis H_i. So, as $M \longrightarrow \infty$ we have

$$\hat{e}^0(b) \longrightarrow \alpha(b)$$

75

$$\hat{e}^1(b) \longrightarrow \beta(b) \tag{4.22}$$

where $\alpha(b)$ and $\beta(b)$ are the probabilities of false alarm and miss respectively. Notice that they are functions of the threshold b.

Since it is not possible to generate a continuous function on the computer, we can simulate equations (4.20) and (4.21) with discrete bins or intervals. In this manner, reasonable values of b (and $-b$) can be chosen to get desirable values of α and β. Using equation (4.21), we choose the value of b that satisfies the constraints

$$\beta \leq \hat{e}^1(x) \text{ for all } x \geq b$$

$$\alpha \leq \hat{e}^0(x) \text{ for all } x \geq b. \tag{4.23}$$

4.5 A Scheme for Multiple Hypothesis Discrimination

Generalizing the binary hypothesis neural network discriminator to a multiple hypothesis discriminator can be achieved without much effort. Instead of two neural network outputs o^0 and o^1, the neural network shall have R outputs, $o^0, o^1, \ldots, o^{R-1}$, corresponding to the R hypotheses $H_0, H_1, \ldots, H_{R-1}$. The test statistic is now the vector $\mathbf{T}_n = (T_n^0, T_n^1, \ldots, T_n^{R-1})^T$, where

$$\mathbf{T}_n = \sum_{j=K}^{n} \Gamma(Z_{K-1+j}, Z_{K-2+j}, \ldots, Z_j)$$

$$= \sum_{j=0}^{n} \Gamma_j. \tag{4.24}$$

The nonlinearity $\Gamma_j = (\gamma_j^0, \gamma_j^1, \ldots, \gamma_j^{R-1})^T$ is formed by setting

$$\gamma_j^i = o_j^i, \quad \text{for} i = 0, 1, \ldots, R - 1. \tag{4.25}$$

Thus, each component of Γ_j has a maximum value of 1. Instead of a two-threshold test, the multiple hypothesis sequential test utilizes R thresholds $a^0, a^1, \ldots, a^{R-1}$. The test proceeds as follows: Obtain a new data sample Z_n. Form the new test statistic $\mathbf{T}_n$. If T_n^i exceeds the other components of $\mathbf{T}_n$ by a margin of a^i, then stop the test and declare H_i. If no decisions are made for the sample Z_n, the next sample, Z_{n+1} is obtained, $\mathbf{T}_n$ is computed, and the test is repeated. Once again, after the maximum number of samples, N, has been reached, a block test is performed. The block test is performed by choosing the hypothesis which satisfies

$$\arg \min_{0 \leq i \leq R-1} \left\{ a^i - \left(T_N^i - \sum_{\substack{0 \leq j \leq R-1 \\ i \neq j}} T_N^j \right) \right\}. \tag{4.26}$$

Figure 4.9 depicts the structure of the multiple hypothesis sequential test.

For $0 \leq l \leq R\text{-}1$,
check if any
$$a^l < T_j^{\,l} - T_j^{\,k} \text{ for all } l \neq k$$

If so, choose $H^{\,l}$
and terminate test.
Otherwise obtain new
sample z_{j+1}

and repeat test.

Terminate test and choose
hypothesis which satisfies:

$$\arg\min_{0 \leq l \leq R\text{-}1} \left\{ a^l - \left(T_N^{\,l} - \sum_{\substack{l \neq k \\ 0 \leq k \leq R\text{-}1}} T_N^{\,k} \right) \right\}$$

$j < N$

yes   no

$T_j^{\,0}$ $T_j^{\,1}$ $T_j^{\,R\text{-}1}$

Σ D Σ D   ● ● ● Σ D

$q_j^{\,0}$ $q_j^{\,1}$ ● ● ● $q_j^{\,R\text{-}1}$

Neural Network

incoming
data
sequence z_j → D $z_{j\text{-}1}$ → D $z_{j\text{-}2}$ → D ● ● ● → D $z_{j\,\text{-}\,(K\text{-}1)}$
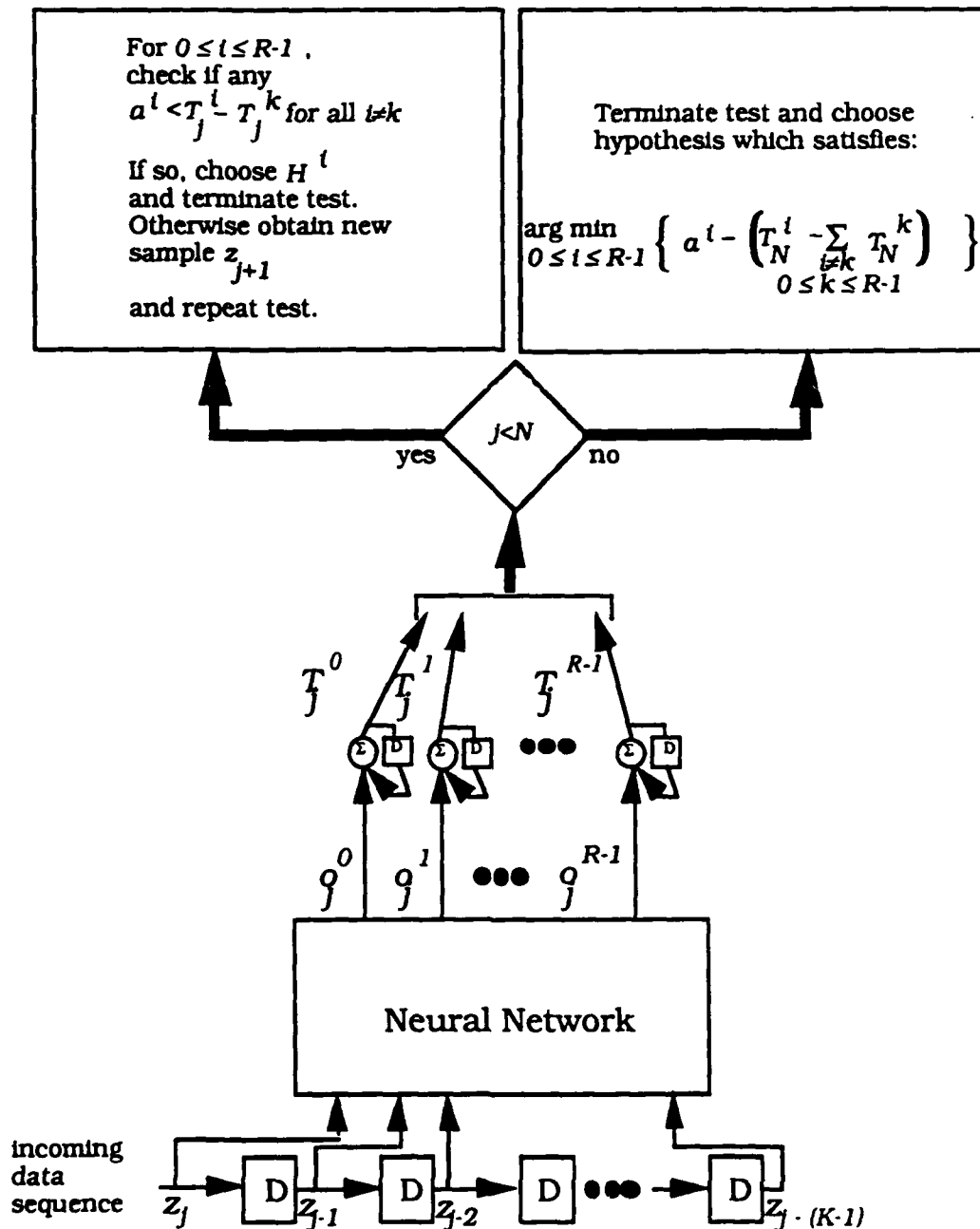
Figure 4.9: Sequential Neural Network Discriminator for Multiple Hypotheses

78

Case	pdfs H_1 vs H_0	Power Ratio H_1 vs H_0	Mean Ratio H_1 vs H_0	Decorrelation Times τ_1, τ_0
Case 1	Lognormal vs Rayleigh	0	0	0.130290, 0.013029
Case 2	Rayleigh vs Rayleigh	3	———	0.130290, 0.013029
Case 3	Rayleigh vs Rayleigh	0	———	0.130290, 0.013029

Table 4.1: Discrimination Cases

4.6 Numerical Results

In this section, the performance characteristics of the neural network discriminators are evaluated. As in Section 2.7, the data used for evaluation of the neural network discriminators is simulated radar data. Table 4.1 summarizes the three data cases considered in this section. Case 1 and Case 2 are identical to Case 1 and Case 2 from Section 2.7. A new case, Case 3, is also considered. Case 3 has Rayleigh pdfs under both H_1 and H_0 with matched means and powers of the marginals. The decorrelation time constants are identical to those of Case 1 and Case 2. Radar envelope samples $\{Z_i\}_{i=0}^{\infty}$ are generated via computer simulation by equations (2.52) through (2.55), just as in Section 2.7.

Tables 4.2 through 4.4 summarize the neural networks simulated and trained to operate in the discriminator structure of Figure 4.6. The first column contains the designated net name. The second column lists the number of inputs (i.e. K), while columns three and four list N_0 and N_1 respectively. Recall from Figures 4.7 and 4.8 that N_0 and N_1, are the number of nodes on layers 0 and 1. All of the neural networks listed in Tables 4.2 through

79

4.4 have two outputs. The final column in Tables 4.2 through 4.4 contains the performance measure, $\bar{S}_5$, which was estimated using the training data after completion of the training phase.

All neural networks were trained using the back propagation algorithm and the training data with the method detailed in Section 4.3. The training data were also generated by simulation using equations (2.52) through (2.55). The sample paths generated so that the number of samples in each path, N, was 1000. The number of sample paths from each hypothesis, M, was set to 50. The constants for the back propagation algorithm were chosen by experimentation to get acceptable convergence rates. The gain was set to 0.001, while the momentum was set to 0. Each sample path of the training data was presented to the network 100 times, (that is, using the notation of Section 4.3, $L=100$.) Since nets 4, 8, and 12 have three layers of nodes, we set $L = 200$ to allow for the expected slower convergence rates associated with the additional layer.

Tables 4.5 through 4.7 summarize the results of the neural network discriminators. The first column of each table contains the name of the neural network. The next two columns contain the probability of false alarm and the probability of detection, respectively. The next column contains the expected number of samples needed to make a decision. Each discriminator was evaluated by simulating 10,000 sample paths from each hypothesis. The probabilities of false alarm and detection were computed by dividing the number of false alarms and correct detections, respectively, by 10,000. The expected number of samples, or average sample number, was computed by averaging the number of samples needed to make a decision for our test sample paths. The thresholds a and b were chosen by experimentation, not by the method of Section 4.4. Our second choice of the thresholds, $a = -20$ and $b = 20$, were used in the simulations presented in this section.

Net	Inputs	N_0	N_1	$\tilde{S}_5$
net 1	2	8	—	1.014 E-06
net 2	4	16	—	3.259 E-06
net 3	8	32	—	2.721 E-07
net 4	4	16	64	7.566 E-08

Table 4.2: Case 1 Neural Networks

Net	Inputs	N_0	N_1	$\tilde{S}_5$
net 5	2	8	—	8.652 E-06
net 6	4	16	—	4.075 E-06
net 7	8	32	—	3.982 E-07
net 8	4	16	64	3.020 E-06

Table 4.3: Case 2 Neural Networks

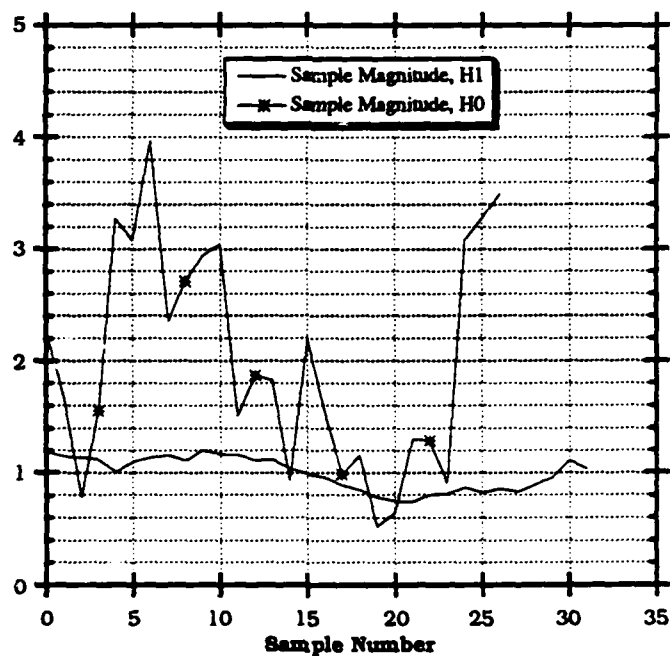| Net	Inputs	N_0	N_1	$\tilde{S}_5$
net 9	2	8	—	1.993 E-05
net 10	4	16	—	4.012 E-06
net 11	8	32	—	3.752 E-07
net 12	4	16	64	1.893 E-05

Table 4.4: Case 3 Neural Networks

Figure 4.10: Sample Paths from H_1 and H_0

Examining the results for Case 1, (see Table 4.5), we see that the discriminators performe well. All discriminators correctly classify all 20,000 sample paths. As the performance measure $\bar{S}_5$ decreases from 1.014E-05 to 7.566E-08, the average sample number decreases from 49 to 28. Figures 4.10 depicts a typical sample path under hypotheses H_1 and H_0, respectively. Figure 4.11 is the corresponding test statistic for the neural network discriminator. Recall from Section 2.8 that the 128 level uniform quantizer designed using the nominal (i.e. known, not estimated) cdfs had a measured probability of false alarm of 0, a probability of detection of 0.9896, and an average sample number of 782. Clearly, the neural network scheme works significantly better than the memoryless discriminator schemes.
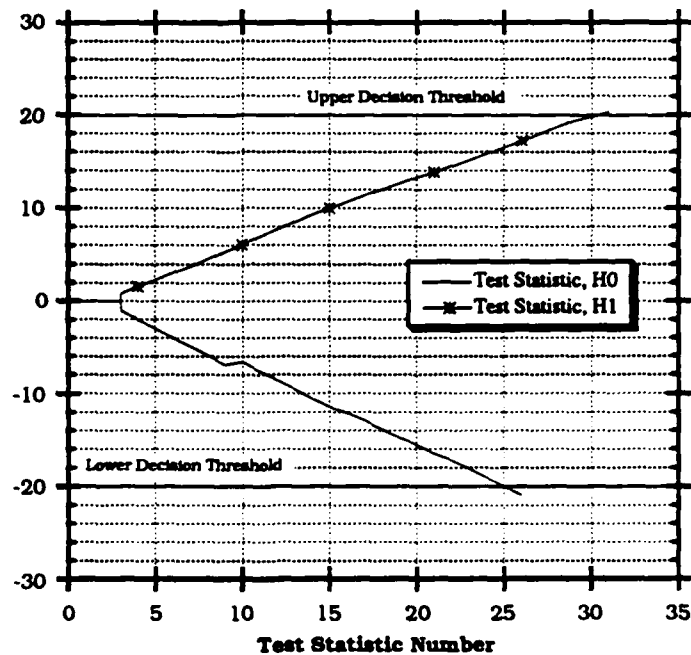
Figure 4.11: Test Statistics from H_1 and H_0

Case 2 results are tabulated in Table 4.6. Most of the discriminators for this case also

performed well. The discriminator using net 5 as its nonlinearity, however, had a probability

of false alarm as high as 0.0997. One can see that the performance measure $\tilde{S}_5$ for net 5 was

slightly higher than the performance measure for the other case 2 nets. The Case 2 data also

implies that smaller values of the performance measure $\tilde{S}_5$ results in better performance

(in probabilities of error and/or average sample number.) Recall from Table 2.7 that the

performance for the optimal 128 level uniform quantizer discriminator had a probability of

false alarm of 0, a probability of detection of 0.9919, and an average sample number of 2660.

Comparing this with the average sample number of 43, and the perfect classification of the

net 7 discriminator, we can conclude that the neural network discriminators outperformed

Net	P_f	P_d	$E[n]$	$\tilde{S}_5$
net 1	0	1	49	1.014 E-06
net 2	0	1	32	3.259 E-06
net 3	0	1	32	2.721 E-07
net 4	0	1	28	7.566 E-08

Table 4.5: Performance of Case 1 Neural Network Discriminators

Net	P_f	P_d	$E[n]$	$\tilde{S}_5$
net 5	0.0997	0.9999	132	8.652 E-06
net 6	0	1	55	4.075 E-06
net 7	0	1	43	3.982 E-07
net 8	0.0001	1	59	3.020 E-06

Table 4.6: Performance of Case 2 Neural Network Discriminators

Net	P_f	P_d	$E[n]$	$\tilde{S}_5$
net 9	0.9904	0.9999	1000	1.993 E-05
net 10	0	1	41	4.012 E-06
net 11	0	1	36	3.752 E-07
net 12	0.4619	0.9986	59	1.893 E-05

Table 4.7: Performance of Case 3 Neural Network Discriminators

the quantizer discriminators for Case 2.

Case 3 results are given in Table 4.7. Recall that Case 3 was Rayleigh vs Rayleigh case with matched means and powers. Note that only the decorrelation times differed. Also note that the quantizer discriminators from Chapter 2 could not be generated for this case because the performance measure $\tilde{S}_3$ is always zero. The discriminator with two inputs, net 9, performed very poorly with its large probability of false alarm of 0.9904 and its large average sample number of 1,000. However, we observe that this network had a large performance measure, $\tilde{S}_4 = 1.993$ E-05. On the other hand, the discriminators using nets 10 and 11, with their perfect classifications and relatively small average sample numbers of 41 and 36, respectively, performed very well. The discriminator using net 12 did not perform well, since its performance measure of 1.893 E-05 was too large. Net 12 probably required more time to converge during its training phase.

A neural network was also trained for the multi-hypothesis discrimination scheme. The number of hypotheses, R, for this experiment was four. Table 4.8 summarizes the four hypotheses. Hypothesis H_0 had a Rayleigh pdf with decorrelation time of 0.013029 seconds. H_1 was lognormal with decorrelation time 0.13029 seconds. H_1 had a 0dB mean ratio and a 0dB power ratio (H_1 vs H_0). H_2 was Rayleigh with the same decorrelation time as H_1, and had a 0dB power ratio (H_2 vs H_0). H_3 is Rician with the same decorreation time as H_1, a mean ratio of 0dB and a 6dB power ratio (H_3 vs H_0). Training data was again generated using equations (2.52) through (2.55) in a computer simulation. The Rician data was created in a manner identical to Rayleigh data, except that the underlying Gaussian processes had a nonzero mean. The number of sample paths, M, was set to 50 for this experiment, while the maximum number of samples, N, was 2000. Training was performed with the gain constant set to 0.001 and the momentum constant set to 0. The number of

presentations, L, was 300.

The multiple hypothesis discriminator was also evaluated via computer simulation. Table 4.9 summarizes the performance of the multi-hypothesis experiment. Each row lists the results for the 10,000 simulated sample paths from each hypothesis. The first column lists the hypothesis number, the second column the number of decisions in favor of H_0, the third the number of choices for H_1, the fourth the number of choices for H_2, the fifth the number of choices for H_3, and the sixth column gives the average sample number for the hypothesis listed in column one. The discriminator acheived over 94 percent correct decisions under each hypothesis and an average sample number (averaged over all hypotheses) of 266.

We have seen networks with various numbers of inputs, layers, and nodes perform very well for our discrimination cases. We now consider the performance of a network with fixed number of inputs and varying nodes. This will help to quantify our *intuitive* belief that more nodes in a neural network will allow a finer tuning of its decision regions. Table 4.10 lists each neural network and the associated number of nodes on each level. Figure 4.12 is a graph of the performance measure $\bar{S}_5$ for each neural network in Table 4.10. Each network was trained with the Case 1 training data. For the back propagation algorithm, the gain was 0.001 while the momentum was 0. Each sample path was presented during training 100 times (*i.e.*, $L = 100$.) The results shown in Figure 4.12 are intuitively pleasing since, as the number of nodes increases (or number of levels for net f), we see that the performance measure $\bar{S}_5$ decreases. This result is expected, as more nodes and levels will allow more hyperplanes to be constructed in the decision space.

To see how the number of presentations, L, affects the performance measure $\bar{S}_5$, an experiment was performed with a two layer network with N_0 set to 16. The network had two

inputs, and was trained to operate on discrimination Case 1. The gain term in the back-propagation algorithm was set to 0.001, while the momentum term was 0. Figure 4.13 shows the performance measure $\tilde{S}_5$ as a function of L, the training cycle number. One can see that the curve is approximately a decaying exponential. At first there is poor performance (large values of $\tilde{S}_5$.) As L increases, the performance improves until it reaches a steady-state minimum. This is expected since, as the number training cycles increases, the decision region should converge to the *optimal* decision region (optimal in the mean squared error sense.)

Hypothesis	pdf	Power Ratio ($\bullet$ vs H_0)	Mean Ratio ($\bullet$ vs H_0)	Decorrelation Time
H_0	Rayleigh	——	——	0.013029
H_1	Lognormal	0	0	0.130290
H_2	Rayleigh	0	——	0.130290
H_3	Rician	6	0	0.130290

Table 4.8: Hypotheses for a Multiple Hypothesis Discrimination Problem

Hypothesis	H_0 choices	H_1 choices	H_2 choices	H_3 choices	E[n]
H_0 true	10000	0	0	0	107
H_1 true	0	9435	0	565	374
H_2 true	1	0	9436	563	370
H_3 true	1	0	316	9683	213

Table 4.9 Results for Multiple Hypothesis Discrimination Problem

Net	Inputs	N_0	N_1
net a	2	2	————
net b	2	4	————
net c	2	8	————
net d	2	16	————
net e	2	32	————
net f	2	4	16

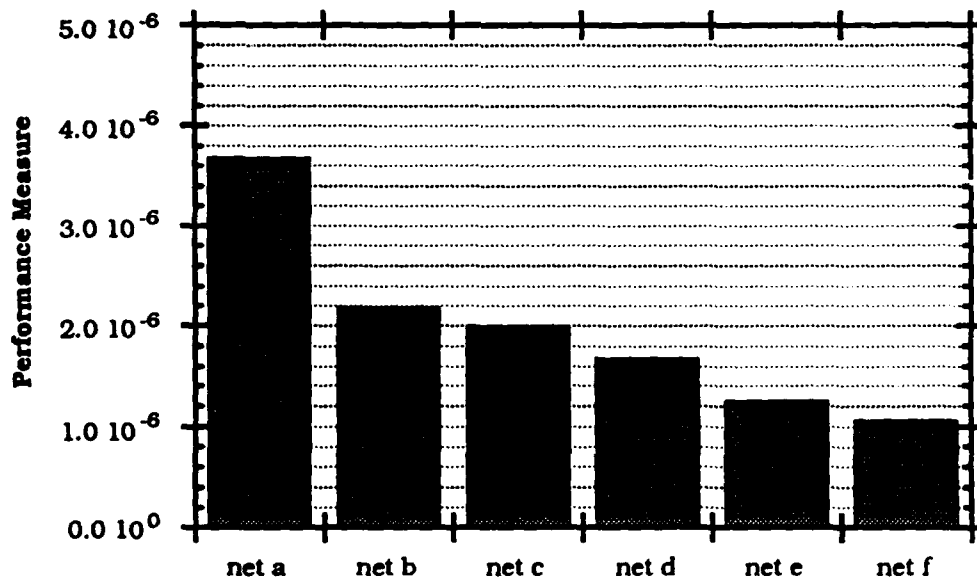Table 4.10: Node Distribution of Networks

Figure 4.12: Performance Measure $\tilde{S}_5$ for Varying Node Distribution
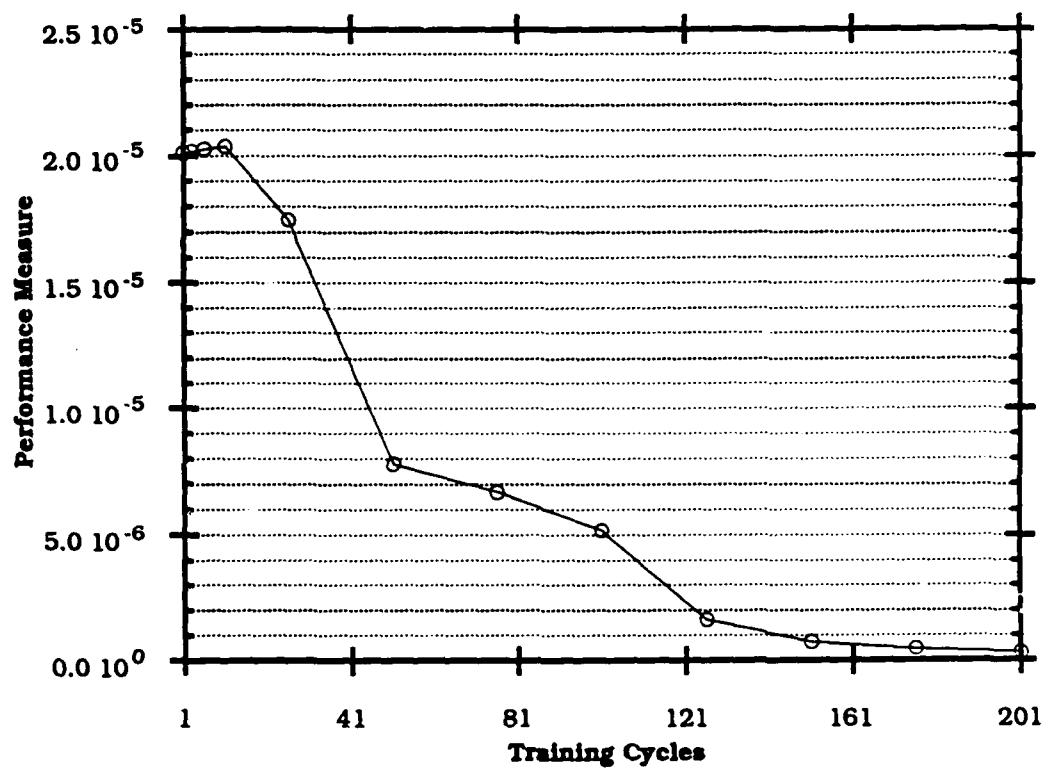
89

Figure 4.13: Performance Measure as a Function of Number of Training Cycles

90

Chapter 5

Mismatch Performance Results

The discriminators in the previous chapter were constructed with *a priori* information involving either known (or assumed) pdfs or training data. If training data were available, the pdfs were either estimated to construct memoryless quantizer discriminators, or the training data was used by a neural network and the back-propagation training algorithm. In many real situations the discriminators are presented with data whose statistics are different from those on which the discriminator was designed to operate. This could be the result of making invalid assumptions about the statistics or of obtaining a non-representative set of training data which results in less accurate estimates of the pdfs. Therefore, the discriminator which is chosen by the designer should be *robust*, that is, the discriminator should not be overly sensitive to changes of the statistics of the data.

In this chapter the performance of our discriminators are evaluated under mismatch conditions (*mismatch* meaning that the data have different statistics from the data for which the discriminators were originally designed). Since there is an infinite number of possibilities for the statistics of the testing data, we can only present some representative mismatch

test	pdfs	Power Ratio	Mean Ratio	Decorrelation Times	
	H_1 vs H_0	H_1 vs H_0	H_1 vs H_0	τ_1, τ_0	σ_0^2
test1.a	Lognormal vs Rayleigh	0	0	0.013029, 0.130290	4
test1.b	Lognormal vs Rayleigh	0	0	0.130290, 0.130290	4
test1.c	Lognormal vs Rayleigh	0	0	0.013029, 0.013029	4
test1.d	Lognormal vs Rayleigh	0	0	0.060000, 0.060000	4

Figure 5.1: Tests for Mismatch of Decorrelation Times

conditions. This mismatch study is certainly not a comprehensive study; the cases considered, however, show some interesting characteristics of the performance of our different discriminators under mismatch conditions.

5.1 Mismatch of Decorrelation Times

In this section we consider the performance of the discriminators under mismatch of the decorrelation times τ_0 and τ_1. Since τ_0 and τ_1 correspond to the correlation coefficients ρ_0 and ρ_1, this is effectively a mismatch of the higher order pdfs. The marginal pdfs for these tests remain unchanged. Table 5.1 lists the discrimination tests for mismatch of the decorrelation times τ_0 and τ_1. The mismatch data for all four tests, (test1.a, test1.b, test1.c and test1.d), are lognormal versus Rayleigh with matched means and powers. The underlying Gaussians for H_0 have variance equal to 4. The fourth column of Table 5.1 lists the decorrelation times.

The discriminators considered for these tests are the 128 level uniform quantizer listed in Table 2.5, the 32 level uniform quantizer listed in Table 3.1, and the neural network discriminator referred to as net 2 in Table 4.2. All of these discriminators were designed for lognormal versus Rayleigh with match means and powers and $(\tau_1, \tau_0) = (0.13029, 0.013029)$.

We see that test1.a just reverses the decorrelation times for which the discriminators were designed. In test1.b, both decorrelation times are set to 0.13029. Both decorrelation times for test1.c were 0.013029, while for test1.d both decorrelation times were 0.06.

100 sample paths from each hypothesis were generated accor⁻¹ing to the appropriate distributions listed in Table 5.1 and presented to the discriminators. The sample paths were generated in the same fashion as in previous chapters. Table 5.2 lists the computer simulation results for test1.a, test1.b, test1.c and test1.d for the various discriminators. Columns labeled P_f contain the measure probability of false alarm and columns labelled P_d the probability of detection. Columns labelled with $E[n]$ contain the average number of samples required to make a decision.

The results for the quantizer discriminator from Chapter 2 (*the memoryless quantizer discriminator from known pdfs*), performed well for all four tests. For all tests, the measured probability of false alarm was less than 2 percent, while the probability of detection was greater than 99 percent. The average sample size varied between 802 and 953. This is still reasonable compared to the results from Chapter 2, namely average sample sizes of about 780 and similar probabilities of error.

The quantizer discriminator derived from the estimated pdfs did not perform well for test1.a and test1.b; the probability of false alarm for test1.a and test1.b were 0.40. The quantizer discriminator from estimated pdfs had low error probabilities for test1.c, but a large average sample number of 3437. For test1.d, the quantizer discriminator from estimated

	Memoryless Quantizer Discriminator from Known pdfs			Memoryless Quantizer Discriminator from Estimated pdfs			Neural Network Discriminator		
	P_f	P_d	$E[n]$	P_f	P_d	$E[n]$	P_f	P_d	$E[n]$
test1.a	0.02	0.99	953	0.40	1	1527	0.94	0	167
test1.b	0.02	1	930	0.40	1	1304	0.92	1	98
test1.c	0	1	802	0.01	1	3437	0	0	32
test1.d	0	0.99	864	0.29	1	2347	0.23	1	85

Table 5.2: Results for Mismatch of Decorrelation Times

pdfs had a probability of false alarm of 0.29, a probability of detection of 1, and a average sample number of 2347.

The neural network discriminator worked marginally well for test1.d, but it performed poorly for the other tests. For test1.a, its probability of false alarm (i.e., the error probability under H_0) was 0.94; this corresponded with τ_0 being mismatched. For test1.b, τ_0 was also very different from its nominal value, and the discriminator had a very high probability of false alarm. For test1.c, τ_1 was very different from its nominal value, and for this test the probability of detection was 0 (i.e., the probability of error under H_1 was 1.) For test1.d, the decorrelation times τ_0 and τ_1 were both at values midway between their nominal values; the discriminator performed only marginally well with a low error probability under H_1 and a probability of false alarm of 0.29.

The results in this section imply that the memoryless quantizer discriminators derived from known pdfs tend to discriminate using the marginal pdfs more *heavily* than the bivariate pdfs. This discriminator worked well for all four tests. The lognormal vs Rayleigh marginal pfds of Case 1 produce the sharp increase in the quantizer function for large values of the observed data sample (see Figure 2.3); this corresponds to the tails of the lognormal density being larger than the tails of the Rayleigh density. For small values of the observed data samples, the Rayleigh density values are much larger than the lognormal density value; this produces a sharp drop in the quantization function for small values of the observed data samples.

Apparently the memoryless quantizer discriminator derived from estimated pfds is more dependent upon the bivariate pdfs than the memoryless quantizer discriminator derived from known pdfs. This could be attributed to poor estimation accuracy of the marginal and bivariates.

The neural network discriminator, however, performed poor for most of the tests. This implies that the neural network discriminator places more emphasis on the higher order pdfs than the memoryless quantizer discriminators. Since the memoryless quantizer discriminators use only one observed data sample at a time when forming their test statistic and since the data from these tests are correlated, one could expect the neural network scheme with memory to perform better than the memoryless quantizer discriminators.

	pdfs	Power Ratio	Mean Ratio	Decorrelation Times	
test	H_1 vs H_0	H_1 vs H_0	H_1 vs H_0	τ_1, τ_0	σ_0^2
test2.a	Rayleigh vs Rayleigh	3	——	0.130290, 0.013029	4
test2.b	Rayleigh vs Rayleigh	0	——	0.130290, 0.013029	4
test2.c	Rayleigh vs Rayleigh	9.0309	——	0.130290, 0.013029	1
test2.d	Rayleigh vs Rayleigh	0	——	0.130290, 0.013029	1
test2.e	Lognormal vs Rayleigh	10.4391	10.4391	0.130290, 0.013029	4
test2.f	Rayleigh vs Rayleigh	0	——	0.130290, 0.013029	10

Table 5.3: Tests for Mismatch of Marginal pdfs

5.2 Mismatch of Marginal pdfs

In this section, the values of τ_0 and τ_1 remain unchanged, but the marginal pdfs are varied. This implies that the bivariate pdfs are changed in shape, but the correlation between samples is unchanged. The same discriminators used in Section 5.1 are used for the results presented in this section.

Table 5.3 lists the six tests used in this section. Figures 5.1 through 5.2 illustrate the nominal and mismatch marginal pdfs used for each test. Table 5.4 contains the corresponding discrimination results, which were obtained by simulating 100 sample paths under each hypothesis and were generated in the same manner as previously.

The experiment test2.a used data from Case 2 to evaluate our discriminators (which

	Memoryless Quantizer Discriminator from Known pdfs			Memoryless Quantizer Discriminator from Estimated pdfs			Neural Network Discriminator		
	P_f	P_d	$E[n]$	P_f	P_d	$E[n]$	P_f	P_d	$E[n]$
test2.a	0	0.29	866	0.04	0.91	4263	0	0.66	67
test2.b	0	0.01	1076	0.06	0.43	4049	0	0.94	55
test2.c	0	1	484	0	0.82	1505	0	0.70	67
test2.d	0	0.05	458	0	0.17	658	0	1	50
test2.e	0	1	411	0.01	0.28	3224	0	1	38
test2.f	0.35	0.46	1878	0	0.93	2084	0	0.67	67

Table 5.4: Results for Mismatch of Marginal pdfs

were designed for Case 1). Examining Figure 5.1, we see that the actual pdf for H_0 was unchanged from the nominal one, but that the pdf for H_1 had larger variance and a peak moved to larger values of x. The results (see Table 5.4) show that the memoryless quantizer discriminator derived from the known pdfs performed very poorly. The memoryless quantizer discriminator designed from estimates of the Case 1 pdfs had reasonable error probabilities ($P_f = 0.04$ and $P_m = 0.09$) but a very large average sample number of 4263. The neural network discriminator had a probability of false alarm of 0, a probability of miss of 0.34, and an average sample number of 67.
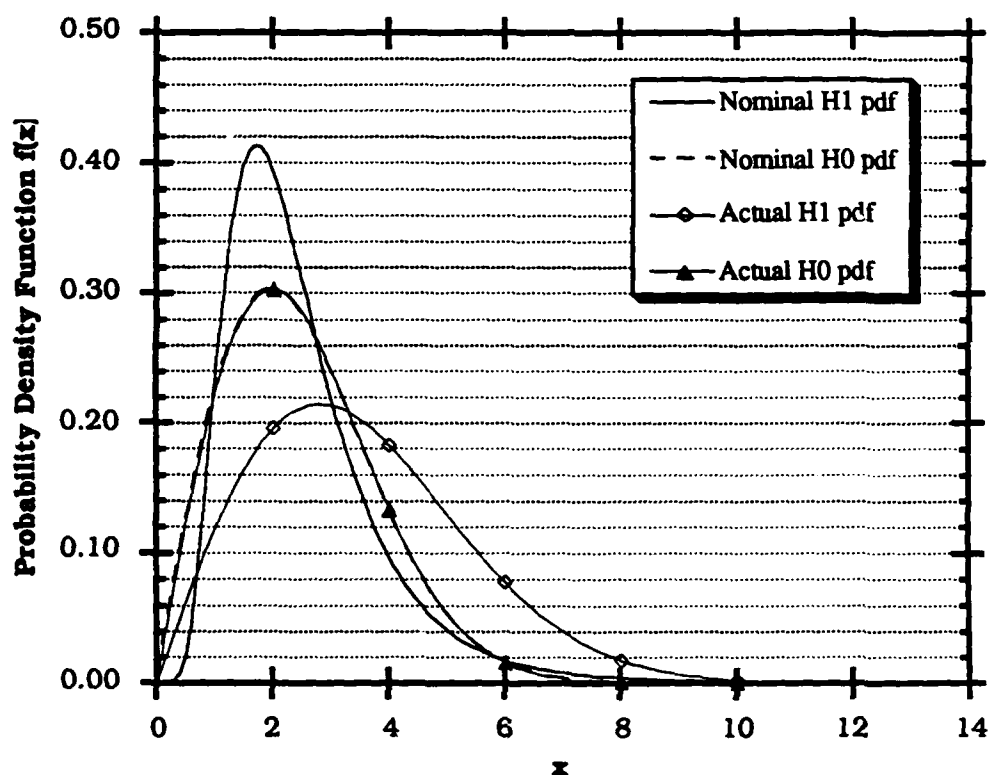
Figure 5.1: Probability Density Functions for test2.a

Experiment test2.b used data from Case 3 (see Chapter 4). Here the actual data from both hypotheses had marginal pdfs matched to the nominal H_0 marginal pdf. Both memory-less quantizer discriminators performed very poorly, while the neural network discriminator had small probabilities of error and a small average sample number.

Experiment test2.c was Rayleigh vs Rayleigh with a power ratio of 9.0309dB (H_1 versus H_0) and an underlying Gaussian variance for H_0 of 1. The marginal pdf for H_1 effectively was changed so that its peak occurred at a larger value of r than the nominal H_0 marginal pdf. The marginal pdf for H_0 had its peak at smaller values of x than the nominal
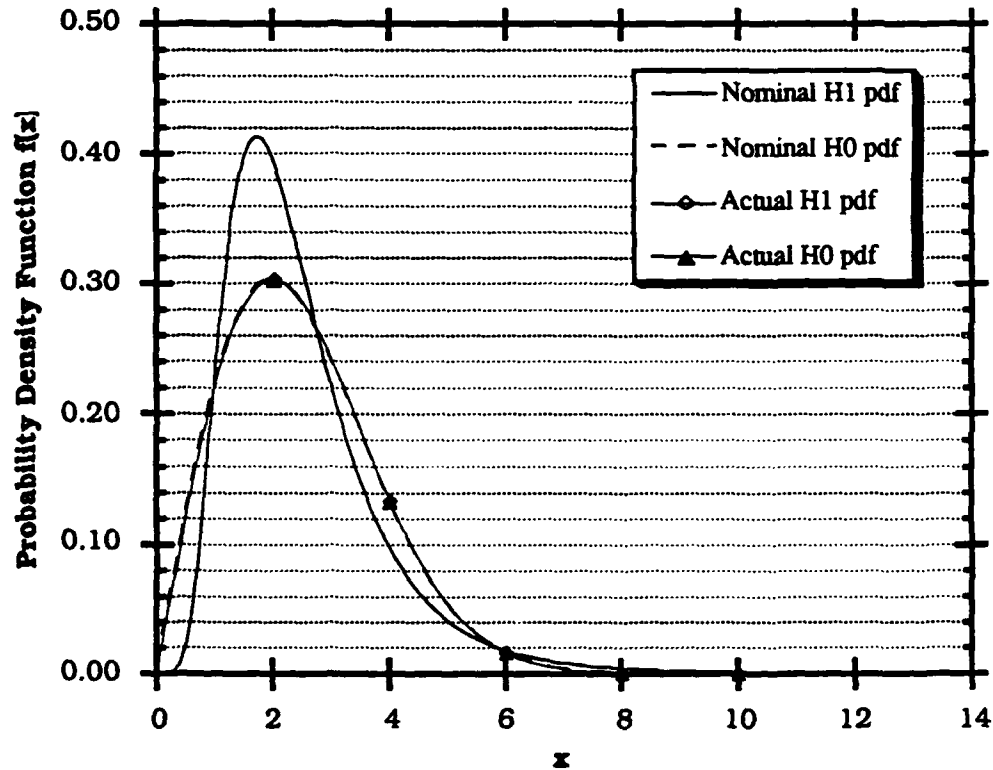
Figure 5.2: Probability Density Functions for test2.b

marginal pdf for H_0. For this experiment, the memoryless quantizer discriminator derived from known pdfs classified all sample paths correctly. The average sample number was 484. We attribute this performance to the probability under each hypothesis being shifted towards the large jumps in the quantization function (see Figure 2.3), where discrimination *power* exists. For this experiment, the memoryless quantizer discriminator derived from estimated pdfs and the neural network discriminator performed marginally well.
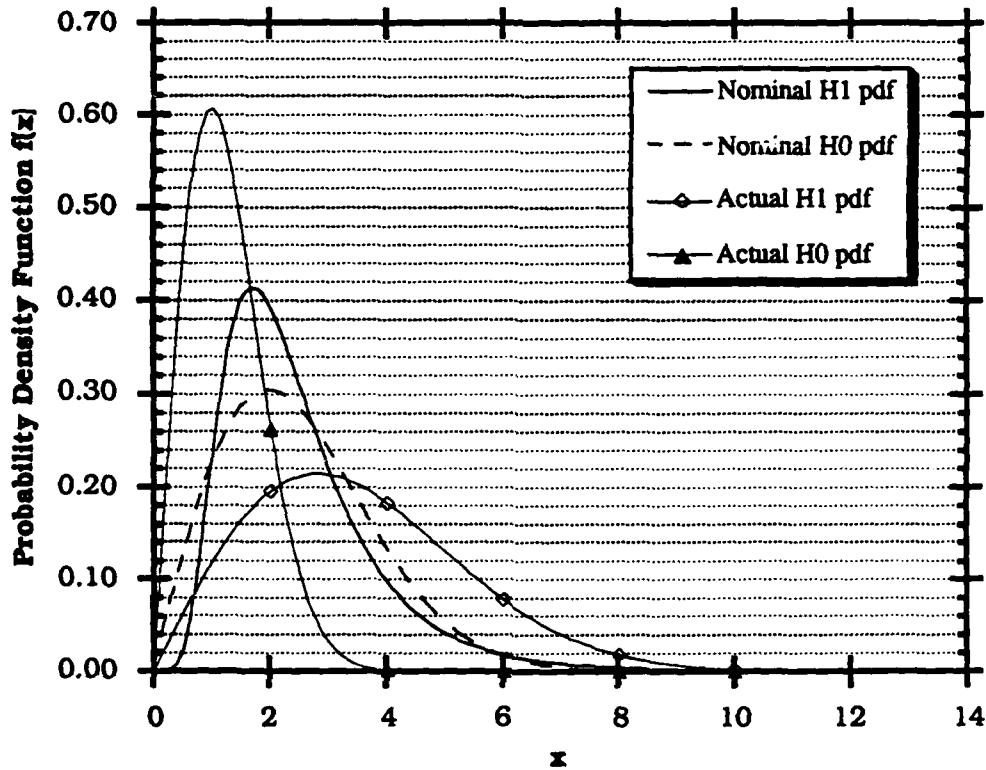
Figure 5.3: Probability Density Functions for test2.c

For test2.d the H_0 marginal pdf was the same as test2.c. However, the H_1 marginal pdf was matched to the H_0 marginal pdf. In this experiment, the neural network discriminator performed very well with perfect classifications and an average sample number of 50. However, the quantizer discriminators performed poorly: note that their error probabilities under H_1 were very large. The large H_1 error probabilities can be attributed to the shift in probability to the negative jump in quantization function (see Figure 2.3); this changes the test statistic T_n in favor of H_0.
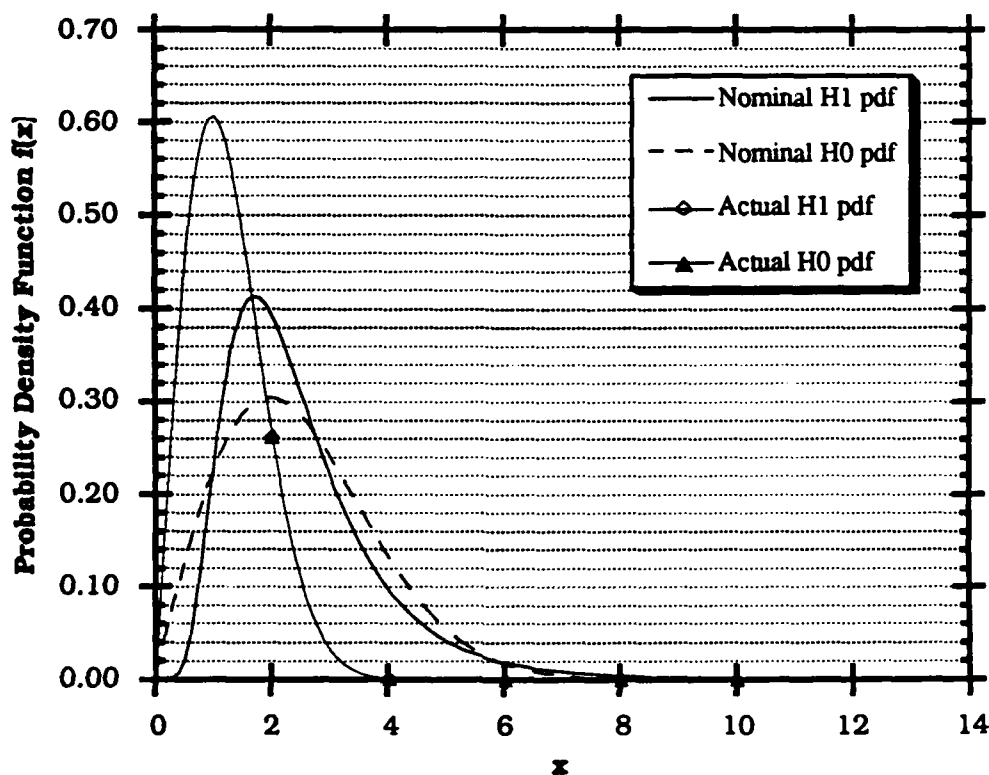
Figure 5.4: Probability Density Functions for test2.d

For test2.e, the H_0 marginal pdf was matched to the nominal H_0 marginal pdf. The H_1 marginal pdf was lognormal with a 10.4391dB mean and power ratio over H_0. With the shift in mass to much greater values of x, the memoryless quantizer discriminators derived from known pdfs performed very well. However, the quantizer discriminators derived from estimated pdfs performed poorly. Note the drop-off in the quantizer function of Figure 3.5, which was attributed to inaccuracies of the pdf estimates at the *tails*. The shift of the H_1 pdf in this mismatch condition probably caused many of the H_1 data samples to be mapped into negative values and to appear to be from H_0. The neural network discriminator classified
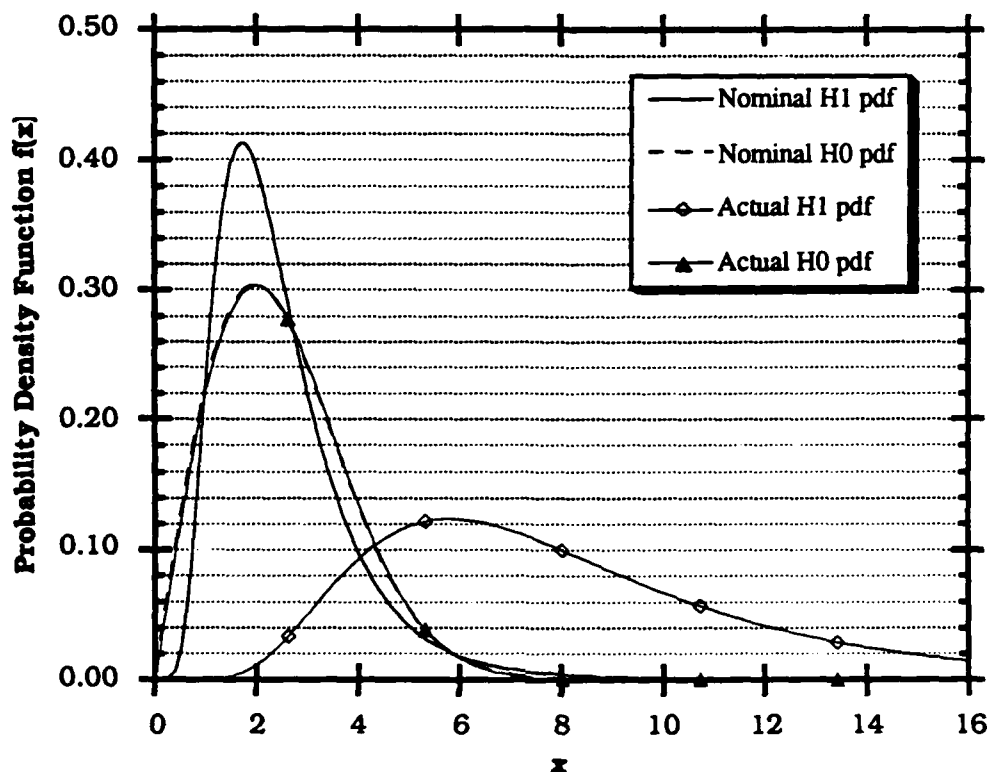
Figure 5.5: Probability Density Functions for test2.e

all sample paths correcly and had an average sample number of 38.

The final experiment, test2.f, was Rayleigh vs Rayleigh with a power difference of 0dB and a variance of the underlying Gaussians for H_0 of 10. Figure 5.6 shows that the peaks occur at larger values of x than both nominal H_1 and H_0 marginal pdfs. Here, however, the Rayleigh process tails were not *heavy* enough to cause the processes to be classified consistently as H_1 for the quantizer discriminators. The quantizer from known pdfs performed poorly with large probabilities of error. The quantizer from estimated pdfs classified all sample paths from H_0 correctly and 93 percent of the H_1 sample paths correctly. The neural
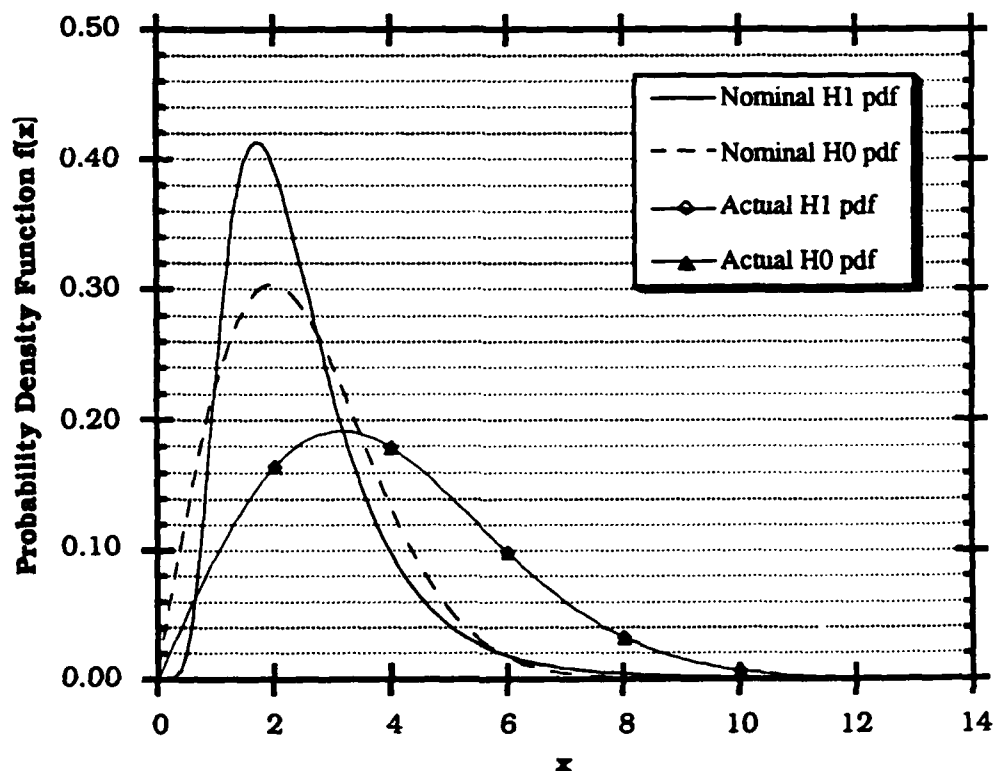
Figure 5.6: Probability Density Functions for test2.f

network discriminator classified all H_0 sample paths correctly and classified 67 percent of the H_1 sample paths correctly.

Although the neural network discriminator did not perform better than the quantizer discriminators in all cases, it showed itself to be less sensitive to changes in the marginal pdfs. The quantizer discriminators worked very well for some these experiments but performed very poorly for others. These results indicate that memoryless quantizer discriminators rely more heavily on marginal pdfs, while the neural network discriminators, which have memory, rely on higher order pdfs - or correlation. These results are to be expected.

Chapter 6

Conclusion

In the previous chapters various schemes for discrimination were considered. Quantization functions were derived that maximize performance measures shown to be useful in both block and sequential discrimination schemes. In Chapter 2, by assuming the probability densities of the data under each hypothesis, quantization functions were constructed for use in discriminators; we consider this a parametric scheme, since pdfs were assumed. In Chapter 3, non-parametric estimates of the marginal and bivariate pdfs were obtained from the *training* data by use of kernel density estimators. These pdfs were input to the expressions for the optimal quantization functions in Chapter 2. The resulting quantization functions were implemented in discriminators; we refer to these memoryless quantizer discriminators as non-parametric. In Chapter 4, another non-parametric scheme was considered. Multilayer perceptron neural networks were utilized to form the nonlinearities used in the test statistic of discriminators. This scheme allowed for the design of discriminators with memory without the requirements of knowledge or estimation of high order pdfs. The neural network scheme utilized training data and the back-propagation training algorithm to form a mean squared optimal non-parametric nonlinearity.

104

The memoryless quantizer discriminators in Chapter 2 performed reasonably well. Their error probabilities were small (for enough quantization levels,) but their average sample numbers were high. These results indicate that more quantization levels give better performance. Quantization functions with optimal breakpoints produced the same discrimination performance as quantization functions with more quantization levels but uniform breakpoints.

The kernel density estimators in Chapter 3 supported the consistency theory; results of experiments that estimated marginal densities showed that larger sets of data resulted in more accurate estimates. Since no theory was available for consistency of higher order pdf estimates for correlated observations, the consistency of bivariates was not checked. The bivariate consistency was also not checked due to processing limitations. Estimation of the sums of the bivariates described in Section 3.3 required as much as 3 days of cpu time on a Convex-210 mini-super computer. If the grid that the estimates were computed over were made denser to result in more accurate quantization functions, much more processing time would be required. Memoryless quantizer discriminators designed using the estimated pdfs had reasonably low error probabilities but extremely high average sample numbers.

The discriminators with memory constructed using multi-layer perceptron neural networks and the back-propagation algorithm performed very well. With training times on the order of a few hours these neural network discriminators, for most experiments, had probabilities of error which could not be measured (with 10,000 simulated sample paths) and average sample numbers at least an order of magnitude smaller than the memoryless quantizer discriminators. Experiments with the number of training cycles using the back-propagation algorithm pleased our intuition; more training decreased the mean squared error of the neural network outputs. Experiments with the number of nodes and layers were also

105

pleasing intuitively; more nodes on a layer decreased the mean squared error of the neural network outputs. The addition of a third layer on the neural network further allowed the back-propagation algorithm to *fine tune* the nonlinearity - thus reducing the mean squared error.

The nonlinearity constructed by the neural network was generalized to operate in a multiple hypothesis classification scheme. Simulation showed that the scheme could classify four hypotheses with error probabilities less than 6 percent and an average sample number of 266.

The use of neural networks as nonlinearities used in forming a test statistic certainly merits further study. Topics that were not addressed in this thesis but might be explored are how to set the training constants and how allocate the number of nodes and layers of the perceptron network. Comparisons could be made between the neural network nonlinearities and the optimal nonlinearities formed with knowledge of the high-order pdfs.

The mismatch results indicate that the memoryless quantizer discriminators are sensitive to changes in the marginal pdfs. The neural network schemes were less sensitive to changes in the marginal pdfs but more sensitive to changes in the higher order pdfs and correlation. The addition of memory to the discriminator apparently explains this phenomena. The robustness of neural network discriminators clearly deserves further study.

Acknowledgement

I would like to thank my advisor, Dr. Evaggelos Geraniotis, for his encouragement and advice. I'd also like to thank Dr. Joe Lawrence & U.S. Naval Research Laboratory for supporting me with a generous fellowship, and the Systems Research Center at the University of Maryland for administering it. Special thanks go to Tom Calomiris and Doug Sauder for their numerous discussions on the topic of discrimination. The numerical results in this thesis would have been very difficult to obtain without the many answers to my computer programming questions provided by Bob Futato, Brook Susman, and Dr. Naba Barkakati. Brook Susman and Allen Goldberg's prompt solutions (*rescue operations*) to computer system problems at NRL are also appreciated. Doug Sauder's advice on T$_{\text{E}}$X was very helpful in preparing this text. Roula and Kate certainly deserve thanks for proofreading this thesis and correcting the spelling and grammatical mistakes. Most of all I'd like to thank my parents for their never ending encouragement and support.

Appendix A

Gradient Evaluation

The evaluation of the matrix $\frac{\partial \hat{P}}{\partial t_k}$ is necessary for a gradient search technique to maximize the performance measure with respect to the breakpoints, unless a finite difference gradient computation is used. If more accuracy than that of a finite difference gradient method is desired, such as when it is expected that the $\hat{P}$ does not change slowly with varying breakpoints t, then the gradient must be explicitly calculated. This appendix contains the necessary equations for computation of the $\frac{\partial \hat{P}}{\partial t_k}$ matrix.

To compute $\frac{\partial \hat{P}}{\partial t_k}$ we employ Leibnitz's rule. For a joint cumulative distribution function $F_{XY}(a, b)$ we have

$$F_{XY}(a,b) = \int_{-\infty}^{a} \int_{-\infty}^{b} f_{XY}(x,y)dxdy \qquad (A.1)$$

where $f_{XY}(x,y)$ is the probability density function associated with $F_{XY}(a,b)$. For our problem of finding optimal breakpoints and levels, we need to evaluate derivatives of the form $\frac{\partial}{\partial a}F_{XY}(a,b)$, $\frac{\partial}{\partial b}F_{XY}(a,b)$, and $\frac{\partial}{\partial c}F_{XY}(c,c)$.

Using Leibnitz's rule [13] on $\frac{\partial}{\partial a} F_{XY}(a,b)$ we get

$$\frac{\partial}{\partial a} F_{XY}(a,b) = \frac{\partial}{\partial a} \int_{-\infty}^{a} \int_{-\infty}^{b} f_{XY}(x,y)dxdy$$

$$= \frac{\partial}{\partial a} \int_{-\infty}^{a} g(b,x)dx$$

$$= \int_{-\infty}^{a} \frac{\partial}{\partial a} g(b,x)dx + g(b,a)\frac{\partial a}{\partial a} - g(b,-\infty)\frac{\partial(-\infty)}{\partial a} \qquad (A.2)$$

$$= g(b,a) = \int_{-\infty}^{b} f_{XY}(a,y)dy$$

where $g(b,x)$ is defined as

$$g(b,x) = \int_{-\infty}^{b} f_{XY}(x,y)dy. \qquad (A.3)$$

In a similar manner it can be shown that

$$\frac{\partial}{\partial b} F_{XY}(a,b) = \int_{-\infty}^{a} f_{XY}(x,b)dx \qquad (A.4)$$

and

$$\frac{\partial}{\partial c} F_{XY}(c,c) = \int_{-\infty}^{c} f_{XY}(x,c)dx + \int_{-\infty}^{c} f_{XY}(c,y)dy. \qquad (A.5)$$

Now consider the nth column and lth row of the matrix $\hat{P}$.

$$\left(\hat{P}_i\right)_{n,\ell} = 2\sum_{j=1}^{m_i} Pr_i\{X_1 \in (\,t_{n-1},t_n\,] \ \ \text{AND} \ \ X_{j+1} \in (\,t_{\ell-1},t_\ell\,]\} \qquad (A.6)$$

$$- (2m_i + 1)[F_i(t_n) - F_i(t_{n-1})][F_i(t_\ell) - F_i(t_{\ell-1})].$$

We know that

$$Pr_i\{\,X_i \in (\,t_{n-1},t_n\,] \ \ \text{AND} \ \ X_{j+1} \in (\,t_{\ell-1},t_\ell\,]\,\}$$

$$= F_i^{X_1,X_{j+1}}(t_n,t_\ell) + F_I^{X_1,X_{j+1}}(t_{n-1},t_{\ell-1}) \qquad (A.7)$$

$$- F_i^{X_1,X_{j+1}}(t_{n-1},t_\ell).$$

110

This yields

$$
\left(\hat{P}\right)_{n,\ell} = 2 \sum_{j=1}^{m_i} \left\{ F_i^{X_1,X_{j+1}}(t_n,t_\ell) + F_i^{X_1,X_{j+1}}(t_{n-1},t_{\ell-1}) \right.
$$

$$
\left. - F_i^{X_1,X_{j+1}}(t_{n-1},t_\ell) - F_i^{X_1,X_{j+1}}(t_n,t_{\ell-1}) \right\} \qquad (A.8)
$$

$$
- (2m_i + 1)\left[F_i(t_n) - F_i(t_{n-1})\right]\left[F_i(t_\ell) - F_i(t_{\ell-1})\right].
$$

So, applying equations $(A.2)$ through $(A.5)$ to the above expression for $\left(\hat{P}\right)_{n,l}$ for the various values of n and l, we get the following expressions:

Case 1: $n, \ell \neq k$ and $n, \ell \neq k+1$

$$
\left(\frac{\partial \hat{P}_i}{\partial t_k}\right)_{n,\ell} = 0 \qquad (A.9)
$$

Case 2: $n = k, \ell \neq k, \ell \neq k+1$

$$
\left(\frac{\partial \hat{P}_i}{\partial t_k}\right)_{k,\ell} = \frac{\partial}{\partial t_k}\left[2 \sum_{j=}^{m_i} \left\{ F_i^{X_1,X_{j+1}}(t_k,t_\ell) + F_i^{X_1,X_{j+1}}(t_{k-1},t_{\ell-1}) \right.\right.
$$

$$
\left. - F_i^{X_1,X_{j+1}}(t_k,t_{\ell-1}) - F_i^{X_1,X_{j+1}}(t_{k-1},t_\ell) \right\}
$$

$$
\left. - (2m_i - 1)\left[F_i(t_k) - F_i(t_{k-1})\right]\left[F_i(t_\ell) - F_i(t_{\ell-1})\right]\right]
$$

$$
= 2 \sum_{j=1}^{m_i} \left\{ \int_0^{t_\ell} f_i^{X_1,X_{j+1}}(tk_1 y)dy + 0 - \int_0^{t_{\ell-1}} f_i^{X_1,X_{j+1}}(t_k,y)dy - 0 \right\} \qquad (A.10)
$$

$$
- (2m_i + 1)\left[F_i(t_\ell) - F_i(t_{\ell-1})\right]F_i(t_k)
$$

$$
= 2 \sum_{j=1}^{m_i} \int_{t_{\ell-1}}^{t_\ell} f_i^{X_1,X_{j+1}}(t_k,y)dy - (2m_i + 1)\left[F_i(t_\ell) - F_i(t_{\ell-1})\right]f_k(t_k)
$$

111

Case 3: $n = k+1, \ell \neq k, \ell \neq k+1$

$$\left(\frac{2\hat{P}_i}{\partial t_k}\right)_{k+1,\ell} = \frac{\partial}{\partial t_k}\left[2\sum_{j=1}^{m_i}\left\{F_i^{X_1,X_{j+1}}(t_{k+1},t_\ell) + F_i^{X_1,X_{j+1}}(t_k,t_{\ell-1})\right.\right.$$

$$\left. - F_i^{X_1,X_{j+1}}(t_{k-1},t_{\ell-1}) - F_i^{X_1,X_{j+1}}(t_k,t_\ell)\right\}$$

$$\left. - (2m_i+1)[F_i(t_{k+1}) - F_i(t_k)][F_i(t_\ell) - F_i(t_{\ell-1})]\right]$$

$$= 2\sum_{j=1}^{m_i}\left\{0 + \int_0^{t_{\ell-1}} f_i^{X_1,X_{j+1}}(t_k,y)dy - 0 - \int_0^{t_\ell} f_i^{X_1,X_{j+1}}(t_k,y)dy\right\}$$

$$- (2m_i+1)[F_i(t_\ell) - F_i(t_{\ell-1})][-f_i(t_k)]$$

$$[F_i(t_\ell) - F_i(t_{\ell-1})]f_k(t_k)$$

$$(A.11)$$

Case 4: $n \neq k, n \neq k+1, \ell = k$

$$\left(\frac{\partial\hat{P}_i}{\partial t_k}\right)_{n,k} = \frac{\partial}{\partial t_k}\left[2\sum_{j=1}^{m_i}\left\{F_i^{X_1,X_{j+1}}(t_n,t_k) + F_i^{X_1,X_{j+1}}(t_{n-1},t_{k-1}) + F_i^{X_1,X_{j+1}}(t_n,t_{k-1})\right.\right.$$

$$\left. + F_i^{X_1,X_{j+1}}(t_{n-1},t_k)\right\}$$

$$\left. - (2m_i+1)[F_i(t_n) - F_i(t_{n-1})][F_i(t_k) - F_i(t_{k-1})]\right]$$

$$= 2\sum_{j=1}^{m_i}\left\{\int_{t_{n-1}}^{t_n} f_i^{X_1,X_{j+1}}(x,t_k)dx + 0 - 0 - \int_0^{t_{n-1}} f_i^{X_1,X_{j+1}}(x,t_k)dx\right\}$$

$$- (2m_i+1)[F_i(t_n) - F_i(t_{n-1})]f_i(t_k)$$

$$= 2\sum_{j=1}^{m_i}\int_{t_{n-1}}^{t_n} f_i^{X_1,X_{j+1}}(x,t_k)dx - (2m_i+1)[F_i(t_n) - F_i(t_{n-1})]f_i(t_k)$$

$$(A.12)$$

112

Case 5: $n \neq k, n \neq k+1, \ell = k+1$

$$\left(\frac{\partial \hat{P}_i}{\partial t_k}\right)_{n,k+1} = \frac{\partial}{\partial t_k}\left[2\sum_{j=1}^{m_i}\left\{F_i^{X_1,X_{j+1}}(t_n,t_{k+1}) + F_i^{X_1,X_{j+1}}(t_{n-1},t_k)\right.\right.$$

$$\left. - F_i^{X_1,X_{j+1}}(t_n,t_k) - F_i^{X_1,X_{j+1}}(t_{n-1},t_{k+1})\right\}$$

$$\left. - (2m_j+1)\left[F_i(t_n) - F_i(t_{n-1})\right]\left[F_i(t_{k+1}) - F_i(t_k)\right]\right]$$

$$= 2\sum_{j+1}^{m_i}\left\{0 + \int_o^{t_{n-1}} f_i^{X_1,X_{j+1}}(x,t_k)dx - \int_0^{t_n} f_i^{X_1,X_{j+1}}(x,t_k)dx - 0\right\}$$

$$- (2m_i+1)\left[F_i(t_n) - F_i(t_{n-1})\right]\left[-f_i(t_k)\right]$$

$$= -2\sum_{j=1}^{m_i}\int_{t_{n-1}}^{t_n} f_i^{X_1,X_{j+1}}(x,t_k)dx + (2m_i+1)\left[F_i(t_n) - F_i(t_{n-1})\right]f_i(t_k)$$

$$(A.13)$$

Case 6: $n = k, \ell = k$

$$\left(\frac{\partial \hat{P}_i}{\partial t_k}\right)_{k,k} = \frac{\partial}{\partial t_k}\left[2\sum_{j=1}^{m_i}\left\{F_i^{X_1,X_{j+1}}(t_k,t_k) + F_i^{X_1,X_{j+1}}(t_{k-1},t_{k-1})\right.\right.$$

$$\left. - F_i^{X_1,X_{j+1}}(t_k,t_{k-1}) - F_i^{X_1,X_{j+1}}(t_{k-1},t_k)\right\}$$

$$\left. - (2m_i+1)\left[F_i(t_k) - F_i(t_{k-1})\right]\left[F_i(t_k) - F_i(t_{k-1})\right]\right]$$

$$= 2\sum_{j=1}^{m_i}\left\{\int_0^{t_k} f_i^{X_1,X_{j+1}}(x,t_k)dx + \int_0^{t_k} f_i^{X_1,X_{j+1}}(t_k,y)dy \right. \qquad (A.14)$$

$$\left. - \int_0^{t_{k-1}} f_i^{X_1,X_{j+1}}(t_k,y)dy - \int_0^{t_{k-1}} (x,t_k)dx\right\}$$

$$- 2(2m_i+1)f_i(t_k)\left[F_i(t_k) - F_i(t_{k-1})\right]$$

$$= 2\sum_{j=1}^{m_i}\left\{\int_{t_{k-1}}^{t_k} f_i^{X_1,X_{j+1}}(X,t_k)dx + \int_0^{t_k} f_i^{X_1,X_{j+1}}(t_k,y)dy\right\}$$

$$- 2(2m_i+1)f_i(t_k)\left[F_i(t_k) - F_i(t_{k-1})\right]$$

113

Case 7: $n = k + 1, \ell = k + 1$

$$\left(\frac{\partial \hat{P}_i}{\partial t_k} \right)_{k+1, k+1} = \frac{\partial}{\partial t_k} \left[2 \sum_{j=1}^{m_i} \left\{ F_i^{X_1, X_{j+1}}(t_{k+1}, t_{k+1}) + F_i^{X_1, X_{j+1}}(t_k, t_k) \right. \right.$$

$$\left. - F_i^{X_1, X_{j+1}}(t_{k+1}, t_k) - F_i^{X_1, X_{j+1}}(t_k, t_{k+1}) \right\}$$

$$\left. - (2m_i + 1) \left[F_i(t_{k+1}) - F_i(t_k) \right] \left[F_i(t_{k+1}) - F_i(t_k) \right] \right]$$

$$= 2 \sum_{j=1}^{m_i} \left\{ \int_0^{t_k} f_i^{X_1, X_{j+1}}(x, t_k) dx + \int_0^{t_k} f_i^{X_1, X_{j+1}}(t_k, y) dy \right. \qquad (A.15)$$

$$\left. - \int_0^{t_{k+1}} f_i^{X_1, X_{j+1}}(X, t_k) dx - \int_0^{t_{k-1}} (x, t_k) dx \right\}$$

$$+ 2(2m_i + 1) f_i(t_k) \left[F_i(t_{k+1}) - F_i(t_k) \right]$$

$$= -2 \sum_{j=1}^{m_i} \left\{ \int_{t_k}^{t_{k+1}} f_i^{X_1, X_{j+1}}(X, t_k) dx + \int_{t_k}^{t_{k+1}} f_i^{X, X_{j+1}}(t_k, y) dy \right\}$$

$$+ 2(2m_i + 1) f_i(t_k) \left[F_i(t_{k+1}) - F_i(t_k) \right]$$

114

Case 8: $n = k, \ell = k+1$

$$\left(\frac{\partial \hat{P}_i}{\partial t_k}\right)_{k,k+1} = \frac{\partial}{\partial t_k}\left[2\sum_{j=1}^{m_i}\left\{F_i^{X_1,X_{j+1}}(t_k, t_{k+1}) + F_i^{X_1,X_{j+1}}(t_{k-1}, t_k)\right.\right.$$

$$\left. - F_i^{X_1,X_{j+1}}(t_k, t_k) - F_i^{X_1,X_{j+1}}(t_{k-1}, t_{k+1})\right\}$$

$$\left. - (2m_i + 1)\left[F_i(t_k) - F_i(t_{k-1})\right]\left[F_i(t_{k+1}) - F_i(t_k)\right]\right]$$

$$= 2\sum_{j=1}^{m_i}\left\{\int_0^{t_{k+1}} f_i^{X_1,X_{j+1}}(t_k, y)dy + \int_0^{t_{k-1}} f_i^{X_1,X_{j+1}}(x, t_k)dx\right.$$

$$\left. - \int_0^{t_k} f_i^{(t_k,y)}dy - \int_0^{t_k} f_i^{X_1,X_{j+1}}(X, t_k)dx\right\}$$

$$\left. - (2m_i + 1)\left\{f_i(t_k)\left[F_i(t_{k+1}) - F_i(t_k)\right] - f_i(t_k)\left[F_i(t_k) - F_i(t_{k-1})\right]\right\}\right.$$

$$= 2\sum_{j=1}^{m_i}\left\{\int_{t_k}^{t_{k+1}} f_i^{X_1,X_{j+1}}(t_k, y)dy - \int_{t_{k-1}}^{t_k} f_i^{X_1,X_{j+1}}(X, t_k)dx\right\}$$

$$- (2m_i + 1)\left[2F_i(t_k) + F_i(t_{k+1}) + F_i(t_{k-1})\right]$$

$$(A.16)$$

Case 9: $n = k+1, \ell = k$

$$\left(\frac{\partial \hat{P}_i}{\partial t_k}\right)_{k+1,k} = \frac{\partial}{\partial t_k}\left[2\sum_{j=1}^{m_i}\left\{F_i^{X_1,X_{j+1}}(t_{k+1},t_k) + F_i^{X_1,X_{j+1}}(t_k,t_{k-1})\right.\right.$$

$$\left. - F_i^{X_1,X_{j+1}}(t_{k+1},t_{k-1}) - F_i^{X_1,X_{j+1}}(t_k,t_k)\right\}$$

$$\left. - (2m_i+1)\left[F_i(t_{k+1}) - F_i(t_k)\right]\left[F_i(t_k) - F_i(t_{k+1})\right]\right]$$

$$= 2\sum_{j=1}^{m_i}\left\{\int_0^{t_{k+1}} f_i^{X_1,X_{j+1}}(x,t_k)dx + \int_0^{t_{k-1}} f_i^{X_1,X_{j+1}}(t_k,y)dy\right. \qquad (A.17)$$

$$\left. - \int_0^{t_k} f_i^{X_1,X_{j+1}}(x,t_k)dx - \int_0^{t_k} f_i^{X_1,X_{j+1}}(t_k,y)dy\right\}$$

$$- (2m_i+1)f_i(t_k)\left[2F_i(t_k) + F_i(t_{k+1}) + F_i(t_{k-1})\right]$$

$$= 2\sum_{j=1}^{m_i}\left\{\int_{t_k}^{t_{k-1}} f_i^{X_1,X_{j+1}}(x,t_k)dx - \int_{t_{k-1}}^{t_k} F_i^{X_1,X_{j+1}}(t_k,y)dy\right\}$$

$$- (2m_i+1)f_k(t_k)\left[2F_i(t_k) + F_i(t_{k+1}) + F_i(t_{k-1})\right]$$

Appendix B

Back-Propagation Algorithm

The back-propagation is a training algorithm designed to minimize the mean square error between the output of the perceptron neural network and the desired output of the network for a given input vector. This is achieved via a gradient descent algorithm. One requirement is that the nonlinearity is continuously differentiable. One commonly used continuously differentiable nonlinearity is the sigmoid $f(y) = \frac{1}{1+e^{-y}}$. The back-propogation algorithm given below assumes a sigmoidal nonlinearity.

Step 1:

The weights and node offset values for all perceptrons in the network are initialized to small random values.

Step 2:

The input vector from the training data, $z = (z_0, z_1, \ldots, z_{K-1})^T$, is presented as an input to the perceptron neural network. The desired output of the neural network, $d = (d^0, d^1, \ldots, d^{M-1})^T$, is also specified at this stage.

117

Step 3:

The actual output of the network. $\mathbf{o} = (o^0, o^1, \ldots, o^{M-1})^T$, is computed by the network in a feed forward manner.

Step 4:

Now the weights are adjusted. Starting at the output nodes and working down towards the the first layer of nodes, the weights are adjusted by

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x^i + \alpha(w_{ij}(t) - w_{ij}(t-1)).$$

$w_{ij}(t)$ is the weight at time t from node i (or input i) to node j. x^i is the output of node i (or is input i.) η is a gain term such that $\eta \in (0,1)$. α is a momentum term such that $\alpha \in (0,1)$. And, δ_j is an error term for node j. For an output node j,

$$\delta_j = o^j(1 - o^j)(d^j - o^j).$$

For an internal node j,

$$\delta_j = x^i(1 - x^i) \sum_k \delta_k w_{jk}$$

where k is over all nodes in the layers above node j. The node offset values are adapted in a similar manner by assuming they are weights from constant valued inputs.

Step 5:

Return to Step 2 and repeat the process for another training vector.

Appendix C

Probability Density Functions

The numerical results of Chapter 2 required kn‿ ledge of the marginal and bivariate cdfs under each hypothesis. This appendix lists the expressions for the Rayleigh and lognormal marginal pdfs and cdfs. The bivariate pdfs are listed. but the bivariate cdfs are not. Bivariate cdfs were obtained via a Simpson's integration of the bivariate pdfs.

The Rayleigh marginal pdf is given by

$$f(x) = \frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right).$$ $$(C.1)$$

The constant σ^2 is the variance of the underlying Gaussian process. The Rayleigh marginal cdf is obtained by integrating $(C.1)$ and is given by

$$F(x) = 1 - \exp\left(-\frac{x^2}{2\sigma^2}\right).$$ $$(C.2)$$

The Rayleigh bivariate pdf has the form

$$f(z, w) = \frac{zw}{\sigma^4 (1 - \rho^2)} \exp\left(-\frac{z^2 + w^2}{2(1 - \rho^2)\sigma^2}\right) I_0 \left\{\frac{\rho zw}{(1 - \rho^2)\sigma^2}\right\}.$$ $$(C.3)$$

In $(C.3)$, ρ is the correlation coefficient between z and w, and $I_0(\cdot)$ is the modified Bessel function of the first kind. The Rayleigh bivariate cdf is obtained by a Simpson's integration of the bivariate pdf.

The lognormal marginal pdf is given by

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma x} \exp\left(-\frac{(\log x - \mu)^2}{2\sigma^2}\right). \qquad (C.4)$$

Here again, σ is the variance of the underlying Gaussian process and μ is the mean of underlying Gaussian process. By integration, the expression for the lognormal marginal cdf is obtained as

$$F(x) = \Phi\left(\frac{\log x - \mu}{\sigma}\right), \qquad (C.5)$$

where $\Phi(\cdot)$ is the normal distribution function defined as

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(\frac{-t^2}{2}\right) dt. \qquad (C.6)$$

The expression for the lognormal bivariate pdf is given as

$$f(w,z) = \frac{1}{2\pi\sqrt{1-\rho^2}\sigma^2 wz} \exp\left(-\left(\frac{(\log w - \mu)^2 + (\log z - \mu)^2}{2(1-\rho^2)\sigma^2}\right.\right.$$

$$\left.\left. -\frac{2\rho(\log w - \mu)(\log z - \mu)}{2(1-\rho^2)\sigma^2}\right)\right). \qquad (C.7)$$

Once again, ρ denotes the correlation coefficient between w and z. The lognormal bivariate cdfs are obtained via Simpson's integration of the bivariate pdfs.

References

[1] H. Vantrees, *Detection, Estimation and Modulation Theory, Part I*, John Wiley & Sons, New York, NY, 1968.

[2] H.V. Poor and J.B. Thomas, "Memoryless Discrete-time Detection of a Constant Signal in m-dependent Noise," *IEEE Trans. Information Theory*, vol. IT-25, pp. 54-61, Jan 1979.

[3] ———, "Memoryless Quantizer-Detectors for Constant Signals in m-dependent Noise," *IEEE Trans. Information Theory*, vol. IT-26, pp. 423-432, Jul 1980.

[4] S.A. Kassam, "Optimum Quantization for Signal Detection," *IEEE Trans. Communication Theory*, vol. COM-25, pp. 479-484, May 1977.

[5] D.W. Sauder and E. Geraniotis, "Optimal and Robust Memoryless Discrimination from Dependent Observations," *IEEE Trans. Information Theory*, 1989, to appear.

[6] E. Geraniotis, "Sequential Tests for Memoryless Discrimination from Dependent Observations - Part I: Optimal Tests," submitted to *IEEE Trans. Information Theory*, 1989, under review.

[7] A. Wald. *Sequential Analysis*. New York: John Wiley and Sons, 1947.

[8] B. L. S. Prakasa Rao, *Nonparametric Functional Estimation*, Academic Press, New York, NY, 1983.

[9] E. Masry, "Probability Density Estimation from Sampled Data," *IEEE Trans. Information Theory*, vol. IT-29, pp.696-709, Sept 1983.

[10] D.W. Sauder and E. Geraniotis, "Optimal One-Step Memory Nonlinearities for Signal Discrimination from Dependent Observations," appeared in *Proceedings of 1990 Conference on Information Sciences and Systems*, Princeton University, 1990.

[11] R. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, vol. 4, pp 4-22, April 1987.

[12] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representations by Error Propagation" in D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel Distributed*

Processing: Exploration in the Microstructure of Cognition. Vol. 1: Foundations., MIT Press, 1986.

[13] Osgood, *Advanced Calculus*, Macmillan, New York, NY, 1925.