ON THE APPLICATION OF MULTI-LAYERED PERCEPTRONS TO NONLINEAR EQUALIZATION FOR FREQUENCY SELECTIVE FADING CHANNELS AND NONLINEAR PREDICTION FOR TIME SELECTIVE RAYLEIGH FADING CHANNELS

By

W. R. KIRKLAND, M. Eng, B. Eng

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
DOCTOR OF PHILOSOPHY

McMaster University

APPLICATION OF MLPS TO FADING CHANNELS

DOCTOR OF PHILOSOPHY (1994) MCMASTER UNIVERSITY
(Electrical and Computer Engineering) Hamilton, Ontario

TITLE: On The Application of Multi-Layered Perceptrons to Nonlinear Equalization for Frequency Selective Fading Channels and Nonlinear Prediction for Time Selective Rayleigh Fading Channels

AUTHOR: W. R. Kirkland
M. Eng
B. Eng

SUPERVISOR: Dr. D. P. Taylor

NUMBER OF PAGES: xv, 158

Abstract

We show that in the additive white Gaussian noise channel that a single neuron using the standard sigmoidal nonlinearity function is effectively a Bayesian estimator for a binary (± 1) level signal and establish a link between the weight(s) of the neuron and the noise variance associated with the signal. This relationship is then extended to neurons where the sigmoidal nonlinearity function incorporates a gain term. By extending these results to multi-level signalling we develop a new nonlinearity function that incorporates a gain term α which simplifies the structure of the neural network. We show that this gain term is linked to the noise variance much like the gain term in a neuron using the sigmoidal nonlinearity for binary signalling.

In applying neural networks to the channel equalization problem for frequency selective fading we make use of complex neurons in the neural network. We look at both binary signalling in two dimensions, and higher level signalling as well. Our results show that while neural nets provide a significant performance increase in the case of binary signalling in two dimensions this performance is not reflected in the results for the higher level signalling schemes. In this case the neural net equalizer performance tends to parallel that of the linear transversal equalizer.

Under the Rayleigh fading channel, we demonstrate the feasibility of using a simple neural network as a phase predictor to significantly reduce the error rate for quadrature phase shift keying signalling in this channel. With this network it is possible to remove the error floor that is normally encountered in this channel for average signal to noise ratios of less than 60 dB.

Preface

The writing of this thesis is the cumulative effort of over 11 years of university. The completion of a Ph.D. is a remarkable accomplishment for anybody but it is especially so for me. For someone whom my parents thought would never make it to university and then once I did, thought I would never leave, I think I have come a long way. Yes, Mom/Dad this is it. I am done.

To have gotten this far with my work would not have been possible without the help of many people. I would like to thank them all for their help and kindness in seeing me along my goal. Special thanks go to Des Taylor, my supervisor, for without his help this work would not have been possible. I appreciate the help, funding, never ending support and the opportunities to not only attend conferences but also the chance to visit the University of Canterbury in Christchurch New Zealand (and points along the way). Thanks also go to TRIO, the Telecommunications Research Institute of Ontario, and NSERC, the National Sciences and Engineering Research Council of Canada, for their support of this work.

Thanks also go out to all of whom I have met at the CRL as they have all added something to my experiences there. Special thanks go to Anne, Sue, Helen, Lola and Doris and the other ladies there. Terry, thanks for all of the help with the computers! A load of thanks goes to one Andy Ukrainec for all of his help even when I was a pain in the butt. Milan, thanks also go to you, not only for your technical support but for the programs that you developed for the optimum solution of equalizer tap weights. I would also like to thank John Smerek for his friendship, never ending support and technical expertise.

To the one person who has shaped my life in so many immeasurable ways I owe a deep debt of gratitude. I have attended his classes for over 5 years and I still have not graduated, nor will I ever graduate as his classes are such that no one ever does. I can only hope to keep on studying and applying what I have learned to everything I do. To Philip McColl, more commonly known among his pupils as Sensei, thank you for everything you have taught me. I honestly believe that I would not have stayed on to finish this degree without your help.

To my parents, thank you ever so much for everything you have done for me for without your support it would not have been possible to finish this degree either. To Judy Bielefeld, that special person in my life, thank you for all of your support.

Bill Kirkland

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The purpose of this thesis is to study the application of multi-layered perceptrons to channel equalization/compensation schemes for two types of channels: the pure frequency selective fading channel and the correlated Rayleigh fading channel, a pure time selective fading channel. The goal of this research is not to propose neural networks as superior structures to present systems but merely to explore the possibilities of their use, to identify suitable network architectures, evaluate their performance and to lay down a solid ground work for their application to channel equalization/compensation schemes. For this we have chosen to look at multi-layered perceptron (MLP) structures using the standard back-propagation (BP) algorithm. While there are more efficient neural network structures than the standard back-propagation multi-layered perceptron it has been demonstrated that the MLP is capable of achieving arbitrary complex functional mappings[1]-[9]. This combined with the fact that the MLP allows for some interesting signal processing interpretations makes it an attractive structure to study.

1.1 History and Motivation

The present trend in the communication industry is towards digital transmission of both analog and digital information over analog channels. The continuous growth of the telecommunications market has demanded increased spectrum efficiency on band

limited channels. This has been met by the use of higher speed data transmissions which are more sensitive to channel disturbances. In turn, this has created the need for more effective equalization schemes to combat the effects of the channel. Channel disturbances may be of either an additive and/or multiplicative form [10] resulting from background thermal noise, impulse noise, co-channel and adjacent channel interference and fades which manifest themselves in the form of frequency translation and attenuation, nonlinear or harmonic distortion and time dispersion.

Our concern is mainly with the effects of time dispersion resulting from frequency selective fading and the effects of thermal noise, however the pure time selective fading channel (correlated Rayleigh fading) is also considered. Rayleigh fading is encountered in mobile radio communication channels. Its effect is to cause the transmitted baseband signal to be multiplied by a complex phaser of random amplitude and phase whereas time dispersion results when the frequency response of the channel deviates from the ideal response of constant amplitude and linear phase[10]. The result of time dispersion is that the effect of a transmitted symbol extends beyond the time interval used to represent that symbol. This is known as intersymbol interference, i.e. ISI. In the telephone channel time dispersion results from the presence of echoes on the telephone line[10]. In this case the channel is unknown but does not change with time. In wireless communication, in particular in line-of-sight digital microwave radio (DMR), [11], the channel disturbance results from the presence of multi-path propagation. Multi-path propagation may be viewed as transmission through a group of channels with differing relative amplitudes and delays [10]. In this case the equalizer must track time varying channel characteristics.

1.1.1 Frequency Selective Fading

Much of the work on adaptive equalization for time-dispersive channels has been well summarized by the literature [10]-[16]. Most certainly the invited paper by Quershi [10] is mandatory reading as it provides a summary of this work up until the early-mid 80's and an excellent list of references. For a more comprehensive survey of the literature (until the early 1970's) one should consult [12] and [13] and for an extensive bibliography of the current literature see Haykin[14].

Conventional equalization schemes have normally used an adaptive linear filter (LTE, a linear transversal equalizer) based upon the LMS (least mean squares) [17][18] or ZF (zero forcing) [19][20][21] adaptation algorithms. Most of the present work on adaptive equalization is based on the method of least squares. The founding work for this started in 1960 when Widrow and Hoff [17] presented the least mean-square (LMS) error adaptive filtering algorithm which has been the fundamental workhorse for adaptive filtering. In the early 60's work centered on the development and understanding of the zero-forcing algorithm developed by Lucky [19][20][21]. By the late 60's the LMS algorithm had been well described and understood. More recent work has been in the development of faster learning algorithms such as recursive least squares [22][23][24] and lattice structures[25]. While these structures are more computationally efficient than the LMS algorithm, their computational complexity has prevented their wide spread use[10].

Even though linear structures have demonstrated good performance, in many instances when the linear equalizer operates in a highly dispersive channel, its performance falls considerably short of the matched filter performance bound obtained by considering the reception of an isolated transmitted pulse [10][21]. Hence, this led to the development of nonlinear receiver structures. The optimum (minimum error probability) equalization scheme, subject to various constraints, has been shown to be a nonlinear structure either in the form of the maximum-likelihood sequence estimator [26], e.g. the Viterbi algorithm, or as a nonlinear tapped delay line structure [27][28]. An alternative though sub-optimum nonlinear receiver structures is the decision feedback equalizer (DFE) [39][16]. With the development of quadrature amplitude modulation (QAM) and the realization of its performance benefits all of these systems have been extended to complex-valued structures suitable for the joint equalization of the in-phase and quadrature signals for QAM[10].

Attention has also turned to the development of sub-optimum nonlinear receiver structures. In [16] Belfiore and Park point out that there is much interest in finding sub-optimum nonlinear receivers that will provide bit-by-bit detection with significant performance advantages over the linear equalizer without the discouraging complexity of the optimum solutions. In [28]-[38] Gibson and Cowan *et al* illustrate

through the use of a simple channel model that the optimal (minimum noise enhancement) combining network for a tapped delay line structure is nonlinear in nature. To achieve an adaptable nonlinear equalizer structure they have made use of feed-forward neural network type structures [28]-[38]. In [27] Ungerboeck proposed two sub-optimal nonlinear equalization strategies, both of which resemble a feed-forward neural network architecture. Ungerboeck's motivation for a nonlinear approach is "... that intersymbol interference between signals with quantized pulse amplitudes is of a discrete nature. Exploiting this discreteness, nonlinear methods can reduce intersymbol interference with less noise enhancement than linear methods." Both the work of Gibson, and Cowan *et al* and Ungerboeck suggest that there is merit in investigating the application of neural networks to adaptive nonlinear equalization. Neural networks with their ability to form arbitrary complex functional mappings appear to offer a flexible and adaptable equalization structure that is capable of achieving performance close to that of the optimum equalizer.

The application of nonlinear neural networks to adaptive channel equalization is relatively new. The most significant and prolific literature on the subject comes from Gibson and Cowan *et al* [28]-[38]. This work seems to have set the standard for comparison. Here, they have demonstrated that the optimum combining network for a tapped delay line structure is a nonlinear structure. Gibson and Cowan *et al* have demonstrated the superiority of neural network equalizers (both multi-layered perceptrons and radial basis function networks [28]-[38]) over conventional linear and decision feedback equalizers through the ability of the neural network equalizer to approach the performance of the optimum Bayesian signal detector. The problem with this work is that it has been restricted to PAM signalling through a 2-3 tap, T (symbol period) spaced channel filter and is based upon using a minimal decision delay and minimizing the number of taps in the equalizer. In particular they have not considered the effects of pulse shaping on neural network equalizer performance. What is needed is to extend this work to more accurately reflect an actual transmission system. Hence a more sophisticated transmission model is needed, one that allows for the use of pulse shaping, different signal constellations (i.e. extension to the case of two dimensional signal sets) and a more realistic channel model. Such work has

been carried out in this thesis, some of which has been reported in [40]. Here the multi-layered perceptron has been extended to the complex domain to allow for the equalization of a quadrature amplitude modulated signal (QAM) in a DMR type radio channel [11][41]. Similar work to that in [40] but using decision feedback equalization has been presented in [42]. Additional work has considered the use of polynomial perceptrons [31], PAO networks [43] and self-organizing networks [44].

With the exception of [32] the above work has focussed on linear channels. While it has been shown that the optimum equalization solution for a linear channel is nonlinear the real advantage of neural networks would seem to lie in their ability to equalize nonlinear channels. In [45] Falconer studied a nonlinear receiver (based upon a DFE) for a 9600-bps (bits per second) modem on a worse than average set of voiceband telephone channels. He found that by considering the nonlinear effects in the channels his nonlinear receiver yielded a lower error probability than a DFE or LTE for every channel considered. For 13 out of the 17 channels, the improvement in error rate was equal to or better than about an order of magnitude. The nonlinear receiver also increased the number of channels yielding a better-than-10^{-4} error rate from 8 to 15 out of 17 channels.

This is not to say that the application of neural networks to the problem of equalizing linear channels is not important. Indeed, while many channels are predominately linear they include some nonlinear effects as well, e.g. see [45][46][47][48]. (In [46] it is pointed out that the primary sources of nonlinearity for digital subscriber loops and voiceband data modems are the data converters, the transmitted pulse asymmetry and the saturation of transformers while in digital radio channels one must also consider the effects of nonlinearities in the high power amplifiers.) In improving performance in this channel a neural network equalizer must first demonstrate that it is at least capable of performing as well as linear techniques in combating the linear part of the channel disturbance before it will be of use in compensating for the less dominant nonlinear terms in the channel. The only work in this area with neural networks appears to be in [32][49] and [50]. In [32] and [49] they consider a simple polynomial channel nonlinearity while in [50] Benvenuto *et al* consider the problem of compensating for the nonlinear effects of a satellite high power amplifier.

With regards to this area, there has been a considerable amount of interest and attention paid to compensating for the effects of nonlinear transmitter amplifiers [51]-[59]. Other areas of importance are in optical fiber [60]-[63] and the magnetic recording media.

1.1.2 Mobile Radio - Rayleigh Fading

The work dealing with Rayleigh fading is not nearly as well summarized as that for frequency selective fading. Much of the work deals with a combination of Rayleigh and frequency selective fading. This is due to the wide range of communication systems operating in a mobile environment. Hence, in this section we present a general overview of mobile communication systems, applicable modulation techniques and channel compensation techniques.

Traditional users for mobile communication systems have been the military, airlines, fishing and shipping companies, the police, fire departments, ambulance services and taxi services. The resultant technology developed for these services has also led to the introduction of personal mobile communications (e.g. cellular radio). While these systems have relied on conventional analog communication systems the ever increasing demand for the radio spectrum has necessitated the need for more advanced and spectrally efficient communication technology, i.e. digital communication systems. This technology brings the ability to provide mobile personal communication services anywhere in the world (e.g. the Iridium project being developed by Motorola [64] and the MSAT, Mobile Satellite, program[65]) but it also brings with it a large number of technological hurdles that need to be overcome. One of the key components of these systems will be the ability to compensate for the fading experienced by a transmitted signal. The particular type of fading in the mobile radio channel we are interested in this thesis is known as Rayleigh fading. With Rayleigh fading the distribution of the amplitude samples of the fading process are Rayleigh distributed. In this channel there is no line-of-sight (LOS) signal present at the receiver, rather the received signal consists of scattered components of the transmitted signal. Such scattering is the result of transmitting around natural and man made objects, e.g. trees and buildings. For this type of fading to be prevalent the greatest difference in

delay between the arrivals of the various scattered signals at the receiver is negligible compared to that of a data symbol period but significant compared to the carrier wavelength. Such fading is restricted to cases where the channel bandwidth is small compared to the carrier frequency, e.g. 5 - 30 kHz bandwidths at 800 - 900 MHz as in the land mobile and satellite radio environments and some cellular radio systems. In the mobile satellite channel there is generally a line-of-sight component which experiences slow time varying fading as well which results in a log-normal fading process. The fading rates for the Rayleigh and log-normal processes however differ by at least one order of magnitude [66] and thus the channel appears to have a Rayleigh characteristic over short periods of time during the deeper fades of the LOS component. The fading rate is a measure of how fast or often the channel fades and is a function of the Doppler shift of the signal received at the mobile station. Doppler shift is in turn a function of the velocity of the mobile station and the carrier frequency. The product of the Doppler shift frequency and the symbol time is known as the BT product and is an indication of the fading rate of the channel, the higher the BT product the higher the fading rate. Typical rates for a 5 kHz channel at 900 MHz range from 0.01 to 0.1 depending upon the velocity of the vehicle. The net effect of the fading process is to cause the received signal to experience random amplitude and phase fluctuations. There is, however, no frequency selective fading.

Two important types of mobile radio systems are mobile satellite and land mobile radio systems. In the former case the bandwidths for these systems range from 5-7.5 kHz (MSAT) up to 20 kHz (Inmarsat-B1) for voice data rates from 4.2 kbits/s to 16 kbits/s[67]. The spectrum allocated to a satellite is divided up into narrow band channels which permits the satellite to be shared amongst a number of users. This is known as frequency division multiple access, FDMA. It is desirable to make these channels as narrow as possible to allow access to the maximum number of users. These low bandwidths channels are made possible with the development of special speech coders with data rates in the range of 8 kbits/s - 16 kbits/s[68][69] and lower. For the 5 kHz satellite channel there is a need for high quality speech encoders operating at 4.8 kbits/s[70].

These speech coders also find use in land mobile digital cellular systems. The

access scheme in digital cellular radio is predominantly time division multiple access, TDMA. Here, the low rate speech data is compressed in time and sent in a short bursts over a much higher rate channel. This time compression allows multiple users to share the same RF channel. Note that the bandwidth of the channel is by necessity much higher than for the narrow band satellite channels. The North American digital cellular standard, IS-54, uses a 30 kHz channel operating at 48.6 kbits/s. The actual voice channel is at 8 kbits/s but three voice channels channels are multiplexed onto a single RF channel. Extra information is also needed for error control, signalling for control channels, guard times and ramp times. The combined total brings the effective transmission rate to 48.6 kbits/s. The European digital cellular standard, GSM, uses an even higher bandwidth of 200 kHz. With these bandwidths, both Rayleigh and frequency selective fading are experienced. The difference between the Rayleigh fading seen by land mobile systems and satellite mobile systems is that the Rayleigh fading for the land mobile systems is much slower due to the larger bandwidths of these systems.

One of the main problems with the cellular radio standards is that the international standards are not fully defined and this has lead to the creation of a large number of different systems with differing data transmission requirements. Upward compatibility with the present North American cellular radio system demands the use of 30 kHz bandwidth digital radios, however, to increase the number of available radio channels a completely different system is likely needed, e.g. a complete restructuring of the system [69]. The one thing that can be said about standards for mobile radio is that they are continuously evolving and rapidly changing.

Considerable attention has focussed on determining suitable modulation techniques for mobile communications. A tutorial and introduction to modulation techniques for land mobile satellite and terrestrial systems can be found in [71]. The application of digital modems for land mobile radio is also discussed by Clark in [72]. Lodge[66] compares data modulation techniques for land mobile satellite channels. More up to date information for mobile communications is contained in[65][67] and [68]. These references provide an overview of mobile radio systems for both satellite and terrestrial systems and provide an idea of their data requirements. Summaries

of modulation techniques in general can be found in [73] and in [74]. Oetting, [73], provides a general summary of modulation techniques for digital radio from 1960 to 1979. Forney *et al* [74] summarize efficient modulation systems for band-limited channels up to about 1984.

Due to the rapid amplitude fluctuations in the mobile radio channel the main theme for modulation has been the use of constant amplitude signalling. The usual approach is through some form of phase shift keying (PSK) although there are proposals for quadrature amplitude modulation (QAM) such as 4-level $\pi/4$-phase shift keying ($\pi/4$-QPSK) with root raised cosine spectral filtering[75][76], 4 and 16 QAM ([77][78] and a variant of 16 QAM based upon a star configuration[79][80]. For $\pi/4$-QPSK the signal constellation consists of two 4-QAM constellations shifted by $\pi/4$ radians. The first data symbol is chosen from one of the symbols in the first constellation. The symbol for the next data symbol is chosen from the other constellation. Thus, the minimum phase shift between two transmitted signals is $\pi/4$ radians and the maximum is $3\pi/4$ radians rather than π. This helps to reduce spurious out of band signals. To further improve the bandwidth of the signal, spectral square root raised cosine pulse shaping is used. This modulation has been chosen as an interm standard (IS-54) for the North American digital cellular network. For 16-QAM, Webb *et al*[79][80] propose the use of a star constellation where one bit is differentially encoded onto the amplitude of the signal and the other three bits are differentially Gray coded onto the phase of constellation. Oversampling and interpolation are also used to compensate for the effects of Rayleigh fading. Another method proposed by Webb *et al* is to increase the symbol set size from 16 to 64, and to use the extra two bits so gained to add block coding in the form of RS or BCH codes to the data. These modulation schemes are driven by two needs: the need to be spectrally efficient and the need to combat the severe fading encountered in the mobile environment. The latter has created the intense interest in binary/quaternary constant envelope modulation.

The excessive errors that occur in mobile communication channel are typically due to the delay spread of the multi-path signals or the Doppler spread due to the motion of vehicles. One or both of these impairments can be significant depending on the data rate. For low bit rate systems (e.g. low kbits/s or 10's kbits/s) the

ratio of RMS delay to the symbol period is small causing very little degradation but the Doppler spread causes a significant error floor. However, the scenario is reversed in high bit rate systems [97]. Discussions of channel characteristics and modelling techniques can be found in [81] - [96]. Stein[81] provides a tutorial on fading channel issues in system engineering and reviews the character of multi-path-induced propagational fading along with the interpretations underlying the use of the Rayleigh fading model to describe the process statistics. Loo and Secord [83] provide a recent summary of computer models for Rayleigh, Rician, log-normal and land mobile satellite fading channels. The Rayleigh fading model used in this thesis is based on the model in [83] which uses a filtered complex Gaussian noise process to produce correlated Rayleigh fading. This model was developed in [90]-[92] and has been used in [82] and [89]. Other models are used in [93], which describes a model for GSM, and in [94], which provides a design and implementation of a channel simulator for wide band mobile radio. Studies on the effects of delay spread on portable radio communications channels with digital modulation have been done by Chuang [95][96].

Recently, there has been much work in the area of compensation techniques for the mobile radio channel. This is due to the increased demand for personal mobile communication services. Most of this work has been devoted to the wide band (30 kHz and greater) land mobile systems while work for satellite systems has mainly been in determining suitable modulation formats, e.g. [65][67][68][71] and [89]. There has also been work in studying and developing trellis coded modulation (TCM) techniques for the mobile channel. McLane *et al* [89][98] have looked at trellis coded modulation for mobile satellite communications. In addition, Divsalar and Simon[99][100], Schlegel and Costello [101] and Liu and Biglieri [102] have all looked at the applications of trellis coded modulation to Rayleigh fading. In [99] - [101], interleaving/deinterleaving is used to decorrelate the fading process. To decode the received signal estimates of the channel fading process are used, however, very little mention is given as to how these estimates are obtained. The construction of fully transparent phase/Doppler invariant trellis coded modulation is considered in [102] by Liu and Biglieri.

One way to obtain estimates of the channel fading process is through the use of pilot symbols or a pilot tone. These are known signals at the receiver and thus

the receiver can use them as a reference signal to estimate the channel. The problem with the use of a pilot tone is that additional energy must be sent with the data signal in the way of the tone signal and that it complicates the filtering that must be done at the receiver. The problem with the use of pilot symbols is that additional symbols must be sent which requires the use of a higher rate channel. An advantage of pilot symbols is that they can be incorporated into the framing format of the signal. Framing is required for all digital communications systems considered in this thesis. This is necessary in order to provide synchronization and clocking information. The use of pilot symbols and pilot tones applied to trellis coded MPSK has been studied by Chan [103]. Addition studies have been made in [66][77] and in [104] - [105]. With the use of pilot symbols, the receiver must accumulate enough pilot symbols to make a good estimate of the channel. Through the use of the Nyquist sampling theorem, Cavers[104] has determined that the pilot symbol spacing is dependent upon Doppler fading rate, e.g. for a BT of 0.05 a pilot symbol is needed every 8 symbols, and for a BT of 0.01 every 20 symbols. This has also been studied in [97]. For channels with frequency selective fading these pilot symbols allow periodic training of equalizers. Provided that the training patterns are close enough together it is possible for the equalizer to track the channel. Examples of such equalization techniques are [107]-[111].

Both the use of coded modulation and pilot tones/symbols requires that redundant information be sent along with the information carrying signal. Under certain circumstances it is possible to perform non-redundant error correction, e.g. PSK [112], MSK [113] and $\pi/4$ DQPSK [114]. These schemes essentially use a form of multiple symbol differential detection. Examples of other multiple symbol detection systems are [115]-[121]. Another form of multi-symbol detection is sequence estimation. In [122], Bune evaluates maximum likelihood sequence estimation (MLSE) equalization for frequency selective fading in a GSM (European digital cellular) system. A survey of results of detecting sequences in non-selective frequency fading is presented by Haeb and Meyr [123]. They also make the point that the estimation of the complex fading distortion is equivalent to carrier recovery. Clark and Jayasinghe [105] use a search technique that relies on four separate estimation and

prediction processes operating in parallel for symbol detection. One process is used for each symbol in a 4-QAM signal constellation. Estimates of the channel fading process are obtained with the use of a pilot tone and a one or two step predictor based upon a least squared fading memory polynomial filter. This approach is based on techniques developed in [124][125] and [126]. A Kalman filter approach is used by Haeb and Meyr [123] to estimate the channel. Their process works under the assumption that the channel is constant over 1 baud interval. This is a simplifying assumption. Rayleigh fading in the mobile radio channel is correlated and it does change over a single baud interval[83][82][89]. By using this information and by sub-sampling (i.e. over sampling) the received signal it is possible to improve upon the performance of systems that sample at a rate of one sample per baud. In [127] Makrakis developes a detection algorithm for MSK (minimum shift keying) signaling based on the maximum likelihood ratio test which uses sub-sampling (4 samples per baud). This results in a computationally complex evaluation process. Further, his results do not clearly indicate the channel parameters that he considered, i.e. no mention is made of the effect of the BT product on the performance of the receiver. Webb *et al*[80] also make use of oversampling and extrapolation methods to compensate for fading. The data rate used is 64 kbits/s with a 16-QAM star constellation. The fading experienced in this channel is much slower than that experience in much narrower band channels, e.g. a 5 kHz channel. Optimum quadratic receivers for rapid Rayleigh fading channels based on sub-sampling the received signal have been developed by Barrett [128] and Dam and Taylor [82].

In this thesis we wish to look at rapid Rayleigh fading such as that experienced in a 5 kHz mobile channel, e.g. an MSAT channel. Here, typical fading rates range from 0.01 to 0.1 (BT). We assume that the Rayleigh fading is correlated and model it as a low pass filtered complex Gaussian process[83][82][89]. By making use of the constant amplitude characteristic of PSK signalling and through oversampling the received signal it is possible to form estimates of the channel fading process. For the optimum quadratic receiver[128][82] the samples of the received signal are used to form the inverse correlation matrix (over two symbol intervals) of the channel fading process. This inverse correlation matrix is then used to detect the phase difference

between two consecutive data symbols. To maintain an accurate estimate of the fading process and to track the channel this inverse correlation must be adapted over time. The samples of the received signal could be used by a neural network to predict the phase of the next sample. More specifically the network could be used to predict the complex phaser of the fading process, $e^{j\theta}$. By training a small complex neural network over a group of channel characteristics ranging from slow to fast fading it could be possible for the network to act as a generalized phase predictor. Provided the predicted phase jumps are not significantly different from the actual phase jumps due to the channel correct data decisions can be made. This could result in a significant reduction in the error rate of the system.

1.1.3 Neural Network History

If one considers a linear combiner to be a single linear neuron then the first application of neural networks to adaptive equalization dates back to the 1960's with the development of zero-forcing and LMS equalization structures. Note that an LMS based linear equalizer is a special case of a multi-layered perceptron, as it may be viewed as a neural network consisting of a single linear neuron. Early work in the application of neural network structures to adaptive filtering can be credited to Widrow due to the development of the LMS algorithm [17][18] and work in nonlinear adaptive structures, the Adaline [17] (adaptive linear element, a linear combiner followed by a hard quantizer) and Madalines (multiple adaptive linear elements) [129] [130]. The most notable contribution of this work was the development of the LMS algorithm as his work with nonlinear neural networks (Adaline/Madaline) never met with much success in the area of adaptive equalization. In [129] Widrow credits the first major extension of the feed forward neural network beyond Madaline I to Werbos [131] in 1974 but he credits Rumelhart, Hinton and Williams [132][133] with making the technique of backpropagation widely known. A good survey and history of feed forward type neural networks (perceptons, madalines and backpropagation) can be found in [129]. For a quick introduction and summary of various types of neural networks the reader is referred to [134] and for more recent work [135].

1.2 Scope of Thesis

The second chapter of this thesis provides the necessary general back ground theory. Basic modulation theory is discussed and the three types of channels that are used in this thesis are described: the additive white Gaussian noise channel, the frequency selective fading channel and the Rayleigh fading channel. The last section of this chapter provides a general introduction to neural networks. In dealing with the networks, the general form of the sigmoidal nonlinearity which incorporates a gain term is used. It is shown how this gain term affects the learning behaviour of the neural network and how it is possible to have two equivalent networks, each with a different value for the gain in the sigmoidal nonlinearity. This section also extends the neural network into the complex domain.

The third chapter focuses on the AWGN channel. This chapter looks at the problem of Bayesian estimation of a digital signal in AWGN. While this is a simple problem it will be shown that it has some important implications in the area of neural networks. It is shown that there is a link between the weight of a single artificial neuron trained to estimate a 2-level signal in AWGN and the Bayesian estimator for the signal. By extending the results to multi-level signalling a new nonlinearity function is developed that incorporates a gain term α which simplifies the structure of the neural network. This chapter establishes the link between the variance of the noise process and the gain term for binary level and multi-level nonlinearities.

Chapter 4 deals with the application of neural networks to channel equalization for the frequency selective fading channel. This chapter discusses the problem of signal detection and estimation. The work in this chapter extends the work of Gibson and Cowan *et al* [28]-[38] from 2-PAM to 4 and 16-QAM (from two levels in one dimension to multiple levels in two dimensions) and includes the effects of pulse shaping. In particular a widely accepted (and perhaps more realistic) channel model is used - the Rummler multi-path fading channel model[41] for the DMR channel.

Chapter 5 addresses the problem of Rayleigh fading in the mobile radio channel. We assume that the Rayleigh fading is correlated. By making use of the constant

amplitude characteristic of PSK signalling and through oversampling the received signal it is possible to form estimates of the channel fading process. Using these estimates the feasibility of using a simple neural network as a phase predictor is demonstrated. With this network it is possible to significantly reduce the error rate for QPSK signalling in the Rayleigh fading channel.

Chapter 2

General Background Theory: Transmission, Equalization and Neural Networks

2.1 Introduction

This chapter serves as a general introduction and provides a background for the theory and systems discussed within this thesis. It is not meant to serve as a detailed discussion of this material but rather to provide introductory information and to establish the notation which is used throughout this thesis.

The first two sections provide an introduction for digital modulation techniques in one and two dimensions: pulse amplitude modulation (PAM), quadrature amplitude modulation (QAM) and phase shift keying (PSK) modulation. The following three sections discuss the additive white Gaussian noise channel, the frequency selective fading channel and the Rayleigh fading channel. These are followed by sections that serve as an introduction to linear equalization theory and neural networks.

2.2 Pulse Amplitude Modulation

In pulse amplitude modulation (PAM) the baseband transmitted signal $s(t)$ is given as

$$s(t) = s_k\, p(t - kT),\ k = 0,\ 1,\ 2,\ \ldots \tag{2.1}$$

where $p(t)$ is a rectangular pulse of duration T seconds (the symbol interval) and s_k is an M-ary (M even) symbol such that $s_k \in \{\pm 1, \pm 3, \ldots, \pm M - 1\} \equiv \underline{\xi}$. In order to restrict the maximum amplitude of the signal to ± 1 the values of s_k may be normalized by a factor of $\frac{1}{M-1}$ without any loss in generality. In doing so, the pulse amplitudes become

$$\frac{1}{M-1}[\pm 1, \pm 3, \ldots, \pm M - 1]$$

In an additive white Gaussian noise (AWGN) channel the sampled received signal, $r(kT)$, is

$$r(kT) = s_k + n(kT) \tag{2.2}$$

where $n(kT)$ is the additive noise with variance σ_n^2. For ease of notation we will denote symbols of the form $r(kT)$ as r_k where k denotes the k^{th} sampling instant[1] such that equation (2.2) becomes

$$r_k = s_k + n_k \tag{2.3}$$

The detection of the transmitted symbol is straight forward, one chooses the detected symbol $\hat{s}_k$ as the symbol in $\underline{\xi}$ that is closest to the received symbol r_k. A more difficult problem is determining the estimate of the transmitted symbol, $\hat{s}_k$, that minimizes the mean squared error between s_k and $\hat{s}_k$. This is the problem of Bayesian estimation which is discussed in chapter (3).

If the transmitted signal passes through a non-ideal channel, which does not have constant amplitude and linear phase[10], then the received signal may be described as

$$r(kT) = s_k h(t_0) + \sum_{j \neq k} s_j h(t_0 + kT - jT) + n(t_0 + kT). \tag{2.4}$$

[1]Each sampling instant is a multiple of T, i.e. we are not sampling the system faster than the baud rate.

where $h(t)$ represents the combined effects of the channel and of the transmit and receive filters [2]. Note that term t_0 is included to account for the channel delay and sampler phase[10]. The first term on the right hand side is the desired signal s_k weighted by $h(t_0)$. The last term is the additive noise with variance σ_n^2 while the middle sum is the intersymbol interference (ISI) from adjacent symbols. Generally, $h(t_0)$ will be handled by the receiver's automatic gain control system leaving the job of ISI cancellation to the adaptive equalizer.

To simplify and clarify the above express we may assume that t_0 has been compensated for and that it may be set to zero. The above equation then becomes

$$r_k = s_k h_0 + \sum_{j \neq k} s_j h_{k-j} + n_k \tag{2.5}$$

which more clearly shows how one might simulate a transmission system in discrete time as opposed to continuous time. To simulate such a system, the channel is represented as a finite impulse response (FIR) filter, a sequence of s_k's is convolved with the filter response and then noise is added to the resulting signal. This type of system is used to illustrate some of the basic ideas involved in equalization in chapter (4).

2.3 Quadrature Amplitude Modulation and Phase Shift Keying

The PAM signalling scheme described in the previous section is an example of a one dimensional signalling system. To make more efficient use of the available transmission bandwidth two dimensional signalling schemes are used. A general transmission scheme in quadrature form is shown in Figure (2.1) where the in-phase (I) channel uses a carrier [3], $cos\,(2\pi\, f_c t)$ and the quadrature (Q) channel uses a carrier, $sin\,(2\pi\, f_c t)$.

[2] In general, the transmitted signal is filtered at both the transmitter and receiver. This is in response to the need to meet bandwidth and pulse shaping requirements, and to minimize the effects of noise.

[3] For simplicity we normalize the magnitude of all the carriers to unity.

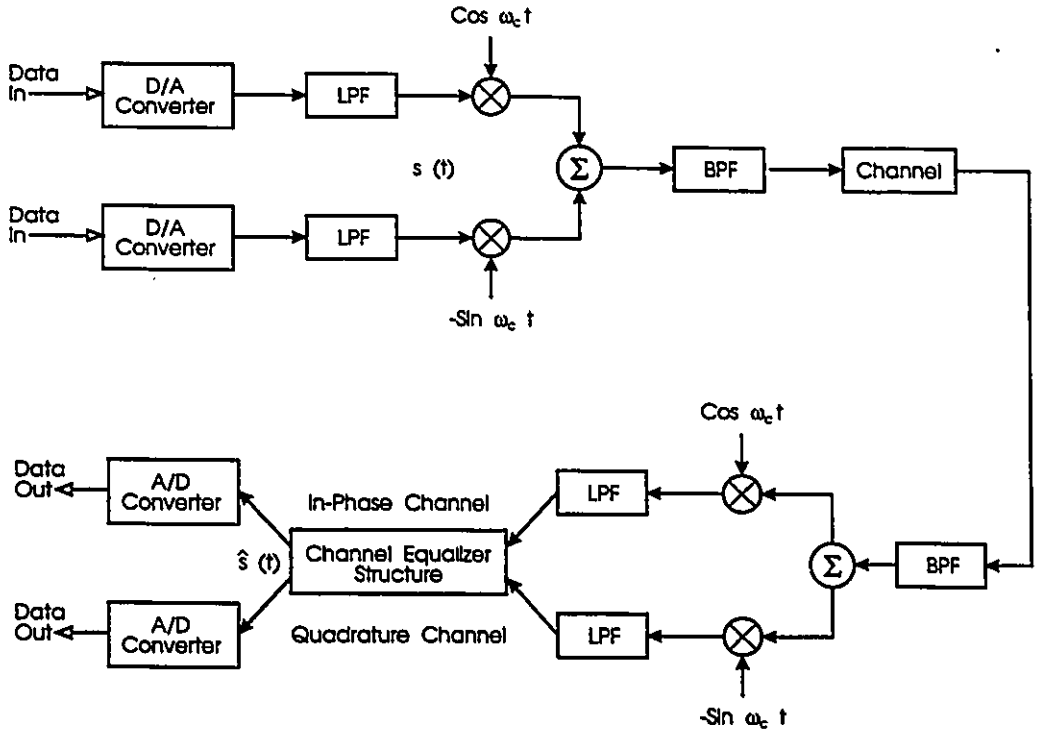

Figure 2.1: General transmission model.

In such a system the modulating signal $s(t)$ consists of two parts, the in-phase component $s_I(t)$ and the quadrature component $s_Q(t)$ where

$$s_I(t) = s_{I_k}\, p(t - kT)$$
$$s_Q(t) = s_{Q_k}\, p(t - kT)$$

and $p(t)$ is the pulse from the pulse shaping filter (PSF). The pulse shaping filter is shown as the low pass filter (LPF) in the in-phase and quadrature channels in Figure 2.1. The resulting transmitted signal $y(t)$ is

$$y(t) = s_I(t)\, cos(2\pi f_c t) - s_Q(t)\, sin(2\pi f_c t)$$

The band pass filter before the channel in Figure (2.1) is used to limit the spectrum of the transmitted signal while the band pass filter after the channel is used to provide receiver selectivity, to reduce or eliminate adjacent channel interference and to reduce the amount of received noise.

For simulation purposes, we represented the modulated waveforms in complex envelope form

$$\tilde{y}(t) = [s_I(t) + j\, s_Q(t)]$$

which can also be represent as

$$\tilde{y}(t) = r(t)\, e^{j\,\psi(t)}$$

where the real envelope $r(t)$ is

$$r(t) = [s_I^2(t) + s_Q^2(t)]^{1/2}$$

and the phase $\psi(t)$ is

$$\psi(t) = tan^{-1}\, \frac{s_Q(t)}{s_I(t)}$$

In simulating the transmission system one uses sample values of $\tilde{y}(t)$ rather than the bandpass modulated signal.

For M-QAM (quadrature amplitude modulation) transmission [136][137] s_{I_k} and s_{Q_k} are elements of $\{\pm 1, \pm 3, \ldots, \pm\sqrt{M} - 1\}$ and the pulse shaping filters are generally root raised cosine spectral filters defined as

$$p_{\sqrt{RC}}(t) = 8\beta \frac{\cos[(1/T + 2\beta)\pi t] + \sin[(1/T - 2\beta)\pi t](8\beta t)^{-1}}{(\pi T^{1/2})[(8\beta t)^2 - 1]} \tag{2.6}$$

where β is the "excess bandwidth" parameter. The combined response at the output of the receiver's pulse shaping filter is the raised cosine spectral pulse

$$p_{RC}(t) = \frac{\cos 2\pi\beta t}{1 - (4\beta t)^2} \left(\frac{\sin \pi t/T}{\pi t/T}\right) \tag{2.7}$$

The purpose of these filters is to limit the bandwidth of the signal and minimize or control intersymbol interference (ISI). Note that both $p_{\sqrt{RC}}(t)$ and $p_{RC}(t)$ are band-limited to $\beta + 1/2T$ and have an infinite time response but the raised cosine pulse has evenly spaced zero crossings every T seconds. The pulse shapes and spectrums of two raised cosine pulses, one with no roll off and the other with 100% roll off, are shown in Figure 2.2. Provided the receiver samples the signal every T seconds at the appropriate time instants there will be zero intersymbol interference. In practice the

receiver will not have perfect knowledge of the proper sampling instants and these must be recovered from the received signal. This is known as timing recovery. By sampling the output of the PSF at the proper time instants it is possible to detect the transmitted signal. If we assume perfect timing and carrier recovery then the output of the receiver's PSF will be (in complex baseband)

$$\hat{s}(kT) = s_{I_k} + j\, s_{Q_k}$$

What is not shown in Figure (2.1) is that the receiver must perform carrier and phase tracking as well as timing recovery. Unless this is done the carrier generated in the receiver will not be at the exact same frequency and phase as the carrier used to transmit the signal. This will cause a performance degradation. How much so will be dependent upon the magnitude of relative frequency and phase errors. To detect the transmitted signal the output of the receiver's PSF must be sampled at the correct instant every T seconds. If this is not done properly then ISI results, degrading system performance. In general the receiver has no knowledge of the proper sampling instants and this must be derived from the received signal through a timing recovery circuit. The discussion of carrier, phase and timing recovery systems is beyond the scope of this thesis. In order to reduce the complexity of the simulation systems considered it is assumed that there is perfect carrier, phase and timing recovery at the receiver.

For M'ary phase shift keying (PSK) the phase of the transmitted signal is modulated such that the resulting signal is

$$y(t) = cos(2\pi f_c t + \theta_k),\ kT < t \leq (k+1)T \text{ and } k = 0,\ 1,\ 2,\ \ldots$$

where f_c is the carrier frequency and θ_k is the modulating phase such that

$$\theta_k = 2\pi n/M,\ n = 0, 1, \ldots, M-1$$

This may be expressed in quadrature form as

$$y(t) = s_{I_k}\, cos(2\pi f_c t) - s_{Q_k}\, sin(2\pi f_c t)$$

in which case

$$\tilde{y}(t) = s_{I_k} + j\, s_{Q_k} = e^{j\theta_k}$$

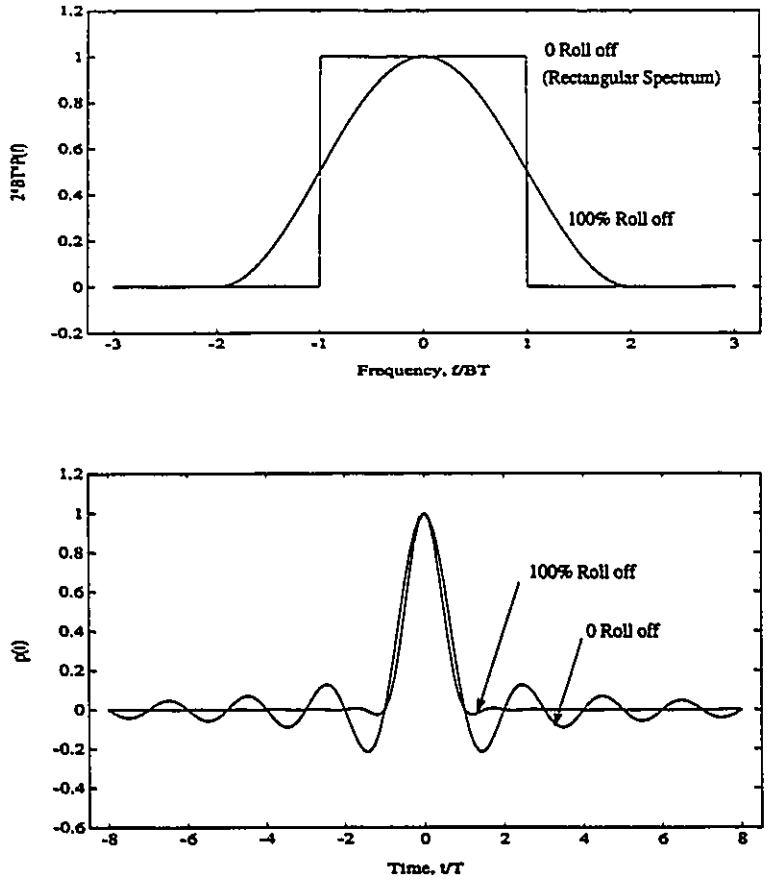

Figure 2.2: Example of a square root raised cosine pulse, a) the pulse spectrum and b) the time response

With the addition of noise, the received signal $r(t)$ is

$$r(t) = s_{I_k} + n_I(t) + j\,(\,s_{Q_k} + n_Q(t)\,)$$

One form of the optimum receiver for this signal is the integrate and dump circuit[138]. Assuming coherent detection (perfect carrier, phase and timing recovery) then the detected signal $\hat{s}_k$ is

$$\hat{s}_k = 1/T \int_{kT}^{(k+1)T} (s_{I_k} + n_I(t))dt + j\,1/T \int_{kT}^{(k+1)T} (s_{Q_k} + n_Q(t))dt$$

The effect of this operation is to average the effects of the noise over the duration of the symbol period which results in the optimum signal to noise ratio for this system.

For differential detection the integrate and dump circuit can still be used. The receiver must still perform timing recovery but no phase recovery is needed as the phase of the detected symbol is taken to be the phase difference between two adjacent received symbols. If the phase of the receiver's carrier is θ which is presumed to be relatively constant over two symbol intervals then the phase of s_k is $\theta_k + \theta$ and s_{k+1} is $\theta_{k+1} + \theta$. The phase of the detected signal is the phase of s_{k+1} less that of s_k

$$\begin{aligned} \theta_{detect} &= \theta_{k+1} + \theta - (\theta_k + \theta) \\ &= \theta_{k+1} - \theta_k \end{aligned} \tag{2.8}$$

2.4 Noise

One deterrent to the correct detection of transmitted signals is the thermal noise arising within the receiver. Individual circuits contain resistors, inductors, and capacitors as well as semiconductor devices[139]. The resistors and semiconductor elements contain charged particles subjected to random motion due to thermal agitation. The random motion of charged particles causes fluctuations in the current waveforms (information-bearing signals) that flow through these components. These fluctuations are called thermal noise and are often of sufficient strength to mask a weak signal and make the recognition of signals a difficult task, i.e. they can cause errors in the detection of the transmitted signal. This noise process is modeled as a zero-mean, stationary Gaussian random process that has a power spectral density that is independent of the operating frequency. Such a process is termed a white noise process. The two sided power spectral density (PSD) of the noise process is denoted as $N_0/2$.

$$S_{NN}(f) = N_0/2 \tag{2.9}$$

This spectral density is not physically realizable since it implies infinite average power, that is

$$\int_{-\infty}^{\infty} S_{NN}(f)df \rightarrow \infty$$

However, bandwidths of real systems are always finite, and since

$$\int_{-B}^{B} S_{NN}(f)df = N_0 B < \infty$$

for any finite bandwidth B, the spectral density given in (2.9) can be used over finite bandwidths.

If $n(t)$ has a power spectral density of the form

$$S_{NN}(f) = \begin{cases} N_0/2, & |f| < B \\ 0, & 0 \text{ elsewhere} \end{cases}$$

then $n(t)$ is called band-limited white noise (with bandwidth B). For an arbitrary low-pass filter of transfer function $H(f)$ we may define the noise equivalent bandwidth[138], B, as

$$B = \frac{\int_0^\infty |H(f)|^2 \, df}{H^2(0)} \tag{2.10}$$

When white noise is passed through an arbitrary low-pass filter of transfer function $H(f)$ where $|H(f)| = 0, f > B$ the power spectral density of the noise becomes

$$S_{NN}(f) = |H(f)|^2 N_0/2$$

This has the effect of colouring the noise, i.e. the noise at the output of the filter is now correlated. This has important implications in the simulation of transmission systems as it says that after the noise passes through the filtering process of the receiver it can no longer be represented as white Gaussian noise. Thus, this must be addressed in the simulation of transmission systems.

In QAM receivers the typical receiver pulse shape is a root raised cosine spectral shape as discussed in section 2.3. It is assumed that the bandwidth of the noise process is greater than the bandwidth of the pulse shaping filters (which is generally true) and hence the power spectral density of $n(t)$ is

$$\begin{aligned} S_{NN}(f) &= |H_{\sqrt{RC}}(f)|^2 N_0/2 \\ &= |H_{RC}(f)| N_0/2 \end{aligned} \tag{2.11}$$

In order to detect a received signal the output of the pulse shaping filter is sampled every T (symbol duration) seconds. Note that the bandwidth of the PSF is greater than the Nyquist frequency of the sampling process, $1/T$. The effect of this is to cause aliasing, the frequency components of the signal outside of the Nyquist bandwidth

are folded back into the Nyquist bandwidth. Because of the raised cosine spectral shape of the power spectral density of the noise process this causes the spectral shape of the sampled noise process to be flat across the Nyquist bandwidth of the signal. Hence, the sampled noise process is white Gaussian noise. Note that this is only true if the samples are taken every T seconds. For PSK we make the assumption that the receivers bandpass filter is ideal in that it has a rectangular frequency response, hence the noise within the bandwidth of the signal is white, i.e. it is white bandlimited Gaussian noise.

2.5 The AWGN Channel

In the AWGN channel we are only concerned with the disturbance caused by the presence of additive white Gaussian noise (AWGN) on the received signal. The assumption is that the channel is ideal and the filtering of the signal at the transmitter and receiver is perfect, i.e. no ISI is generated. This results in a simple transmission model, the received signal, r_k, is equal to the transmitted signal, s_k, with the addition of some noise component (white Gaussian noise), n_k,

$$r_k = s_k + n_k$$

In the case of M-QAM and M-PSK r_k (complex baseband) is given as

$$r_k = s_{I_k} + n_{I_k} + j(s_{Q_k} + n_{Q_k})$$

As the I and Q channels are independent this model may be broken down into two PAM systems corresponding to the real (r_{R_k}) and imaginary (r_{I_k}) parts of r_k,

$$\begin{aligned} r_{R_k} &= s_{R_k} + n_{R_k} \\ r_{I_k} &= s_{I_k} + n_{I_k} \end{aligned}$$

Thus the study of the detection or estimation of QAM and PSK signals in AWGN may be reduced to the study of PAM systems[4]. The estimation of PAM signals is considered in chapter (3).

[4]Note that in the case of PSK modulation the values that s_{I_k} and s_{Q_k} may take on are not evenly spaced as is the case for QAM modulation

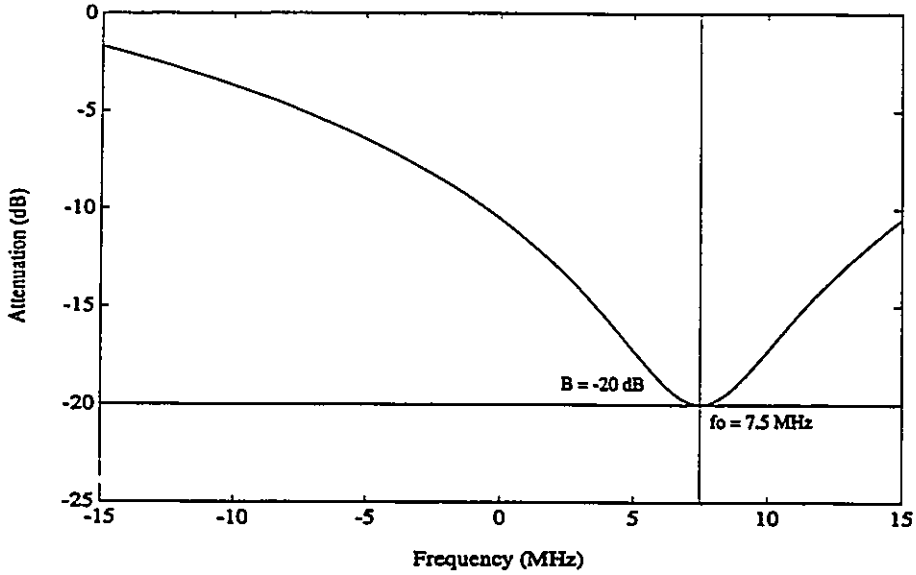

Figure 2.3: Frequency response of the Rummler channel model for a 20 dB notch at 7.5 MHz

2.6 Frequency Selective Fading

When the frequency response of the channel deviates from the ideal response of constant amplitude and linear phase[10] (i.e. frequency selective fading) time dispersion results. The result of time dispersion is that the effect of a transmitted symbol extends beyond the time interval used to represent that symbol. This is known as intersymbol interference, i.e. ISI. In the telephone channel time dispersion results from the presence of echoes on the telephone line[10]. The channel is unknown but does not change with time. In wireless communication, in particular in line-of-sight digital microwave radio (DMR), [11], the channel disturbance results from the presence of multi-path propagation. Multi-path propagation may be viewed as transmission through a group of channels with differing relative amplitudes and delays [10]. In this case the channel is unknown and time varying.

In providing a meaningful study of the effects of ISI we are interested a realistic communications channel. We have chosen to study the line-of-sight DMR channel through the use of Rummler's multi-path fading channel model[41]. This permits the use of existing software to determine the performance of optimum linear transversal

equalizers (LTE) based upon the LMS algorithm in order for comparisons to be drawn between LTE's and neural networks based on the transversal structure (tapped delay line), i.e. neural network transversal equalizers (NNTE). This follows from the work of Amitay and Greenstein [140] in their study on the optimum performance of the LMS linear equalizer in the DMR channel based upon this channel model. While this model is for a wide band (30 MHz) channel, the overall transmission scheme may be used to model a lower bandwidth channel by a suitable normalization of transmission parameters. The channel model has the form

$$H_c(f) = a\{1 - be^{-j2\pi(f-f_0)\tau}) \tag{2.12}$$

$A =$	$20\log(a)$	flat fade component (dB)
$B =$	$20\log(1-b)$	notch depth (dB)
f_0		notch location relative to the center of the channel (MHz)
$\tau =$	$6.3nS$	a constant, approx. 5 times the inverse of the channel bandwidth

which displays both a flat fading component (constant level of frequency attenuation over the whole channel) of depth A (dB) and a frequency selective component in the form of a notch located at f_0 MHz from the center of the channel with a depth of B (dB). The constant τ is chosen as approximately 5 times the inverse of the channel bandwidth, e.g. Rummler [41] uses $6.3\,nS$ for a 30 MHz channel. In practice the flat fade or median fade component of the channel is compensated for by the automatic gain control (AGC) of the receiver and only serves to change the signal to noise ratio (SNR). For this reason a may be set to unity and one may concentrate on the notch characteristics of the channel, namely, the parameters B and f_0 at various signal to noise ratios.

This is a simplified three path fading model as the delay τ is fixed. The model[41] is assumed to consist of a direct path which is unfaded; a second path similar in strength and close enough in delay to the first path that their composite response over the channel width is constant (the flat fading parameter a); and a third path at

relative delay τ which provides the frequency shaping of $H_c(j\omega)$. For a true three path model the delay would be allowed to vary. However, unless the channel response can be determined to an accuracy on the order of 0.001 dB, a unique set of parameters, a, b, τ and f_0 can not be determined for more than half of the faded channel conditions encountered in Rummler's propagation experiment[141]. To avoid this problem, it is necessary to suppress or fix one of the model parameters. Rummler[141] shows that the delay, τ is the only parameter which, when fixed, produces a reasonable model.

For a transmission model, 4 and 16-QAM where considered. A baud rate of 22.5 MBaud was used with a channel bandwidth of 30 MHz at a signal to noise ratio (SNR) of 63 dB, a typical value for such a channel. The pulse shaping was achieved using square root raised cosine spectral pulses with a roll off factor of 0.33.

This may seem like an excessively high speed implementation for neural networks but it should be remembered that even though the neural networks were used in this explicit channel it is only the ratios (baud rate, bandwidth and SNR) that matter, e.g. this could just as easily represent 22.5 KBaud at 30 KHz. The time/frequency parameters merely serve to specify the links between various components of the transmission model.

2.7 Rayleigh Fading

A special case of pure-time selective fading is the Rayleigh fading channel (the distribution of samples of the fading process are Rayleigh distributed). In this channel there is no line-of-sight (LOS) signal present at the receiver, rather the received signal consists of scattered components of the transmitted signal. Such scattering is the result of transmitting around natural and man made objects, e.g. trees and buildings. For this type of fading to be prevalent the greatest difference in delay between the arrivals of the various scattered signals at the receiver is negligible compared to that of a data symbol period but significant compared to the carrier wavelength. The result is that although there is no frequency selective fading the amplitude of the received signal randomly varies in time. The effect of the fading process is to cause the transmitted baseband signal to be multiplied by a complex phasor of random

amplitude and phase, $a(t)$. If $s(t)$ is the transmitted signal then the received signal is

$$r(t) = a(t)\, s(t) + n(t)$$

where $a(t)$ is the complex channel gain. That is,

$$a(t) = r_c(t) e^{j\theta_c(t)}$$

where $r_c(t)$ is the time varying amplitude of the channel fading process and $\theta_c(t)$ is the time varying phase. Such fading is restricted to cases where the channel bandwidth is small compared to the carrier frequency, e.g. 5 - 30 KHz at 800 - 900 MHz as in the cellular radio and mobile satellite radio environments.

In the mobile satellite channel there is generally a line-of-sight component which experiences time varying fading which results in a log-normal fading process. The fading rates for Rayleigh and log-normal processes however differ by at least one order of magnitude at least [66] and thus the channel appears to have a Rayleigh characteristic over short periods of time during the deeper fades of the LOS component. The fading rate is a measure of how fast or often the channel fades and is a function of the Doppler shift of the transmitted signal at the mobile station. The Doppler shift is a function of the velocity of the mobile station and the carrier frequency. The greatest Doppler shift occurs when the vehicle is moving directly towards or away from an oncoming radio wave. The Doppler shift will be determined by the vehicle's velocity and the frequency of transmission. For a vehicle velocity, v, and carrier wavelength, λ,

$$f_d = \frac{v}{\lambda}$$

The fading rate is characterized by its time-bandwidth product, BT, where BT is the product of f_d and T (symbol duration). For a 2400 baud mobile channel at 900 MHz typical BT products range from 0.01 to 0.1. This type of fading makes it very difficult to perform coherent detection as the fading process results in rapid phase and amplitude changes in the received signal as can be seen in Figures (2.4 and 2.5) which shows a typical fading channel sequence for a BT product of 0.05. For this type of channel it is highly desirable to transmit a signal that has a constant envelope, e.g.

PSK, and to perform differential detection at the receiver. For PSK signalling the received signal $r(t)$ is

$$\begin{aligned} r(t) &= r_c(t)e^{j\theta_c(t)}e^{j\theta_k} + n(t) \\ &= r_c(t)e^{j(\theta_c(t)+\theta_k)} + n(t) \end{aligned} \tag{2.13}$$

For simulation purposes the complex channel gain, $a(t)$, is generated by filtering two separate white Gaussian noise processes and using the output of one filter as the real part of $a(t)$ and the other as the imaginary part of $a(t)$. The type of filter used is a third order Butterworth filter with a bandwidth determined by the desired BT product for the channel[89].

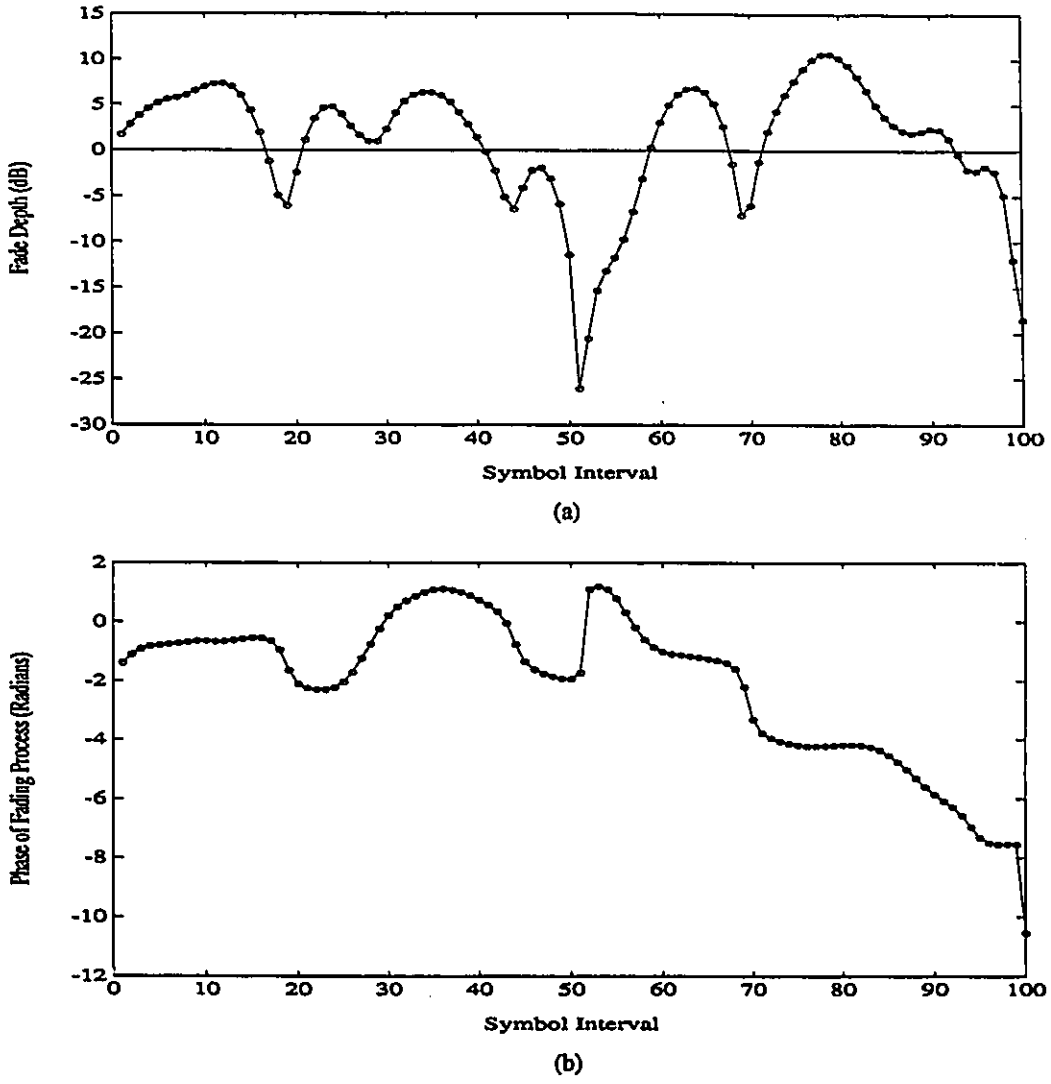

Figure 2.4: An example of the Rayleigh channel fading process, (a) Magnitude of the fading process, and (b) the phase of the fading process.

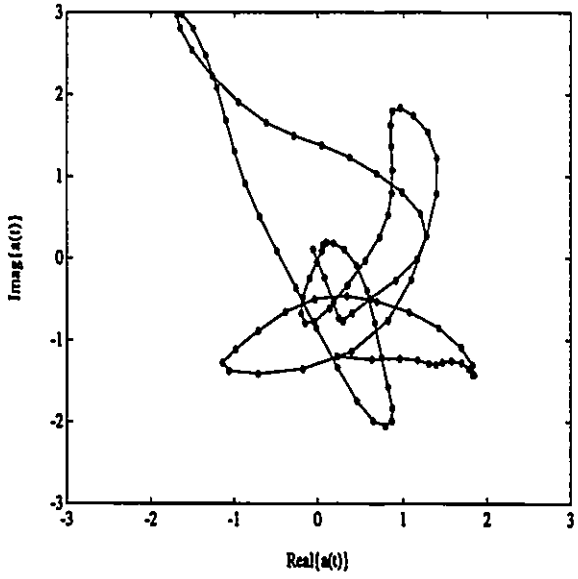

Figure 2.5: A look at the same fading process projected down the time axis

2.8 Neural Networks

For our purposes, the goal of using neural networks for channel compensation is to use their nonlinear capabilities to approach (or achieve) the performance of optimum channel compensation schemes without the associated complexity. While neural networks are more complex and exhibit slower learning times than linear techniques they offer the advantage of being able to form nonlinear mappings and hence should be able to achieve improved performance over linear techniques. The networks that seem particularly suitable to this problem are based upon the feedforward architecture such as multi-layered perceptron (MLP) and radial basis function (RBF) networks. Indeed, most of the literature on neural network equalization has been dedicated to these two structures with some other work being done on the use of Petri Nets[43], Kohonen networks[44] and Madaline structures[130][129].

The advantage of the RBF network is that it has only a single hidden layer of basis functions which are linearly combined to form the output of the network. As such, RBF networks are generally less complex and exhibit faster learning times than MLP networks that achieve the same functional mapping. In addition, MLP networks are more sensitive to the choice of learning parameters and network topology than RBF networks. Though, both networks are capable of generating arbitrary complex nonlinear decision regions. The problem that both networks share is in determining the size/configuration of the network. Chen *et al*[37] have used an orthogonal least-squares (OLS) algorithm to solve this problem for the RBF network and have extended it to multi-output RBF networks. For MLP networks there have been numerous algorithms proposed that determine the allocation of neurons and the pruning of network weights though none of these schemes have been applied to adaptive equalization. The advantage of using MLP's is that they are easy to use and that they are easily extended to the complex domain which makes them suitable for use with QAM and PSK signals[5]. MLP networks allow us to achieve some insight into the behaviour of neural networks. The same behaviour can also be achieved with the use of RBF networks. Because the inputs to a MLP are connected directly to the weights of the

[5]For simulations QAM and PSK signals are represented in complex baseband

first hidden layer of the network it is possible to apply linear filter theory to these weights. This can not be done with RBF networks as the inputs to the network are connected directly to the network's nonlinear basis functions. It also turns out that the nonlinearity that is commonly used in MLP's has an interesting signal processing interpretation which is discussed in Chapter (3.2). It is for these reasons and time restrictions that only MLP's are considered in this thesis. The use of only the MLP is not unreasonable as both RBF and MLP networks are capable of achieving the same functional mappings.

2.9 Basic Terminology for the MLP

The multi-layer perceptron consists of interconnected layers of neurons with the outputs of neurons from one layer only connected to the neurons in the next layer. There are no interconnections between neurons within a given layer of the network. This is seen in Figure (2.6). A single neuron is shown in Figure (2.7). The output of an individual neuron is a function of the sum of its weighted inputs. This function is termed the activation of the neuron and is usually a nonlinear function. The most common activation function is the sigmoidal or logistic function which is depicted in Figure (2.8). For the j^{th} neuron of the network this function is given as[134, 133]

$$\begin{aligned} f_j(net_j) &= logistic(net_j) \\ &= \frac{1}{(1+e^{-net_j})} \\ &= 1/2(1+tanh(net_j)) \end{aligned} \tag{2.14}$$

where

$$\begin{aligned} net_j &= \sum_i x_i w_{ij} - \theta_j \\ x_i &= f_i(net_i) \end{aligned}$$

and where x_i is the output of the i^{th} neuron, w_{ij} is the weight linking the i^{th} neuron to the j^{th} neuron and θ_j is a bias term which is similar to a threshold level in a binary valued neuron. The adaptation of the i^{th} weight of the j^{th} nonlinear neuron, w_{ij}, at

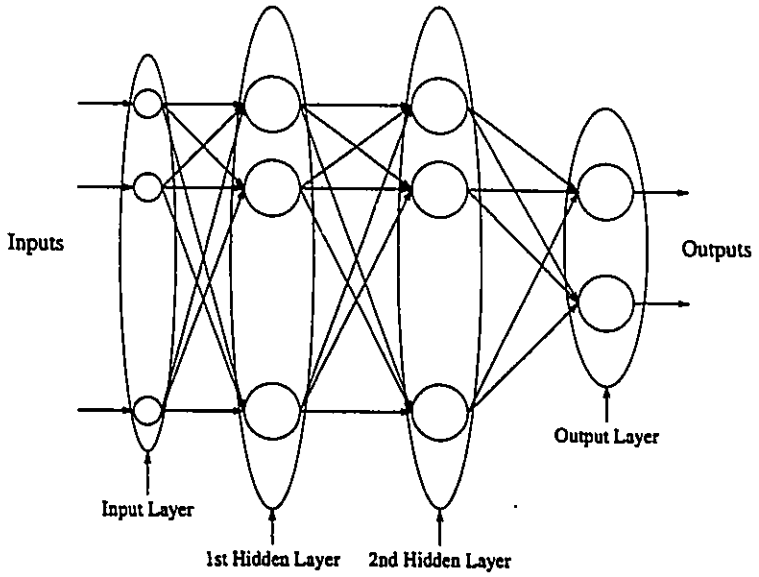

Figure 2.6: Multi-Layered Perceptron.

time $n+1$ is governed by[134, 133]

$$w_{ij}(n+1) = w_{ij}(n) + \eta\, \delta_j(n)\, x_i(n)$$

where η is the learning constant and $\delta_j(n)$ is the error term for the j^{th} neuron at time n. If the j^{th} neuron is an output node then $\delta_j(n)$ is defined as

$$\begin{aligned} \delta_j &= f_j'(net_j)(d_j(n) - x_j(n)) \\ &= x_j(n)\,(1 - x_j(n))(d_j(n) - x_j(n)) \end{aligned} \tag{2.15}$$

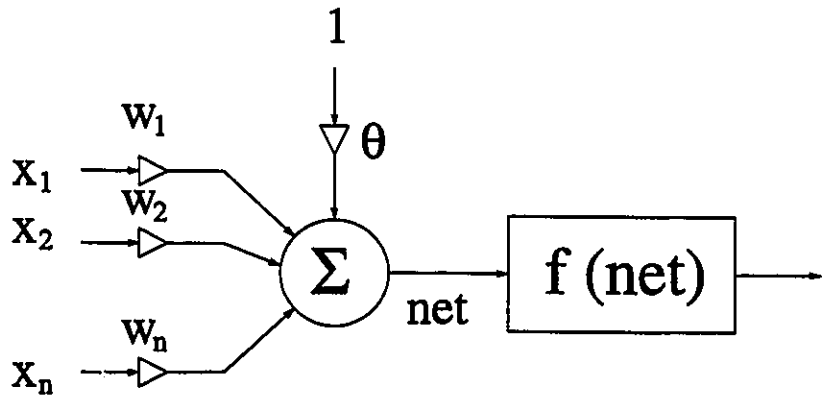

Figure 2.7: A single neuron.

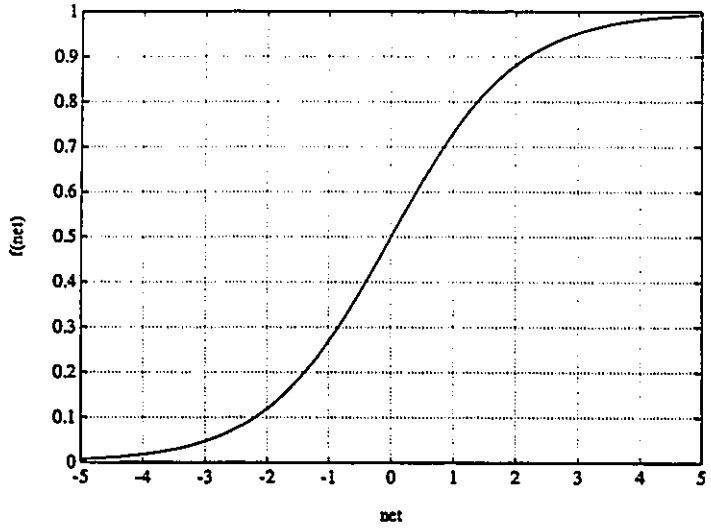

Figure 2.8: The sigmoidal activation function.

where $d_j(n)$ is the desired output of node j and $x_j(n)$ is the actual output at time n. Note that the derivative of the sigmoidal function (2.15) is given as:

$$\begin{aligned} f'(net_j(n)) &= f(net_j(n))\,(1 - f(net_j(n))) \\ &= x_j(n)\,(1 - x_j(n)) \end{aligned} \tag{2.16}$$

If the j^{th} neuron is an internal, hidden node, then

$$\begin{aligned} \delta_j(n) &= f'_j(net_j(n)) \sum_k \delta_k(n)\, w_{jk}(n) \\ &= x_j(n)\,(1 - x_j(n)) \sum_k \delta_k(n)\, w_{jk}(n) \end{aligned} \tag{2.17}$$

where k ranges over all nodes that connect to the output of node j. The thresholds, θ_j, are adapted in a similar manner by assuming them to be connection weights on links from auxiliary constant-valued inputs. A momentum term is sometimes added to assist learning. In our experiments we found that provided η was chosen appropriately the momentum term made little difference, hence we set it to zero.

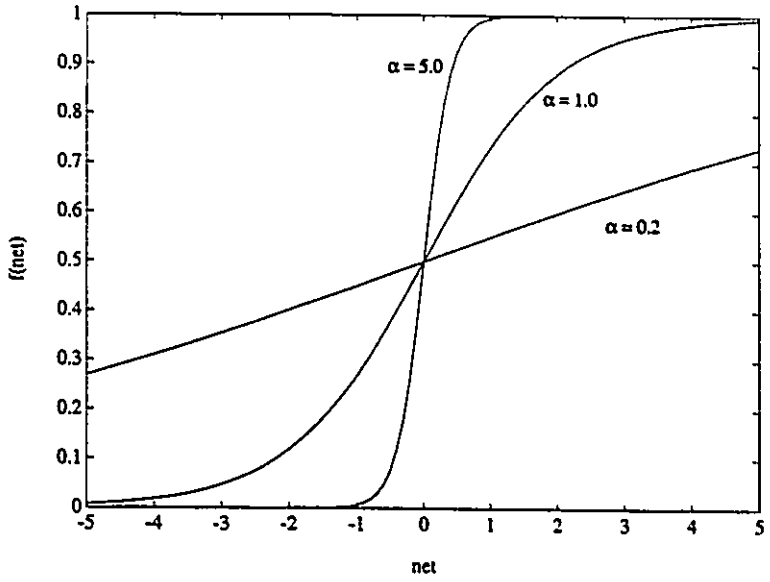

Figure 2.9: The effect of the gain term on the sigmoidal nonlinearity.

2.10 The Gain Term

A more general definition of the nonlinearity function allows for the addition of a gain term, α, such that

$$f(net) = logistic(\alpha\, net) \tag{2.18}$$

This is shown in Figure (2.9) for three different values of α - 5, 1, 0.2.

For a neural network, if α is made a constant then nothing is gained by introducing this gain term. The reason is that two networks, one with a gain term of α_1 and the other with a gain of α_2, can be made equivalent by appropriately scaling the weights and bias terms of the neurons such that

$$w_{2ij} = \frac{\alpha_1}{\alpha_2} w_{1ij} \tag{2.19}$$

$$\theta_{2j} = \frac{\alpha_1}{\alpha_2} \theta_{1j} \tag{2.20}$$

The subscripts 1 and 2 refer to the terms from the first and second networks respectively. Once training is initiated the networks will not remain equivalent unless the following constraints are placed upon the adaptation parameter η and the momentum

term, μ, (Appendix A)

$$\eta_1 \alpha_1^2 = \eta_2 \alpha_2^2 \tag{2.21}$$

$$\mu_1 = \mu_2 \tag{2.22}$$

According to these equations there is nothing that can be achieved with the addition of the gain term that could not be achieved by appropriately scaling the parameters of the neural network. If, however, the gain term is allowed to adapt[6] then what has been achieved is a means of effectively adapting the adaptation parameter η of the back propagation algorithm for each neuron in the network. For the j^{th} neuron, if α_j is increased by a factor of γ then from equation (2.21) the effective change in η for the j^{th} neuron at time $n+1$ is

$$\eta_{n+1} = \gamma^2 \eta_n$$

The back propagation of the gain term α is easily achieved with very little increase in computational complexity as most of the terms associated with back propagating α are available from the equations associated with the standard back propagation algorithm[142]. The results in [142] show that adapting the gain term effectively increases the learning rate of the network. What has been shown here is that adapting the gain term amounts to adapting the adaptation parameter η of the standard back propagation algorithm for each neuron in the neural network.

2.11 Complex Neural Networks

In order to use a MLP network with QAM or PSK it is necessary to consider complex valued MLP's. The complex neuron is shown in Figure (2.10). The appropriate equations for a complex MLP are as follows:

$$\begin{aligned} x_i &= x_{R_i} + j\, x_{I_i} \\ &= o_i \\ w_{ij} &= w_{R_{ij}} + j\, w_{I_{ij}} \\ net_j &= net_{R_j} + j\, net_{I_j} \end{aligned}$$

[6]The derivation of the adaptation equations for α is shown in Appendix B

$$
\begin{aligned}
&= \mathbf{X}^{\mathrm{T}}\mathbf{W} = \mathbf{W}^{\mathrm{T}}\mathbf{X} \\
f_j(net_j) &= f_{R_j}(net_{R_j}) + j\, f_{I_j}(net_{I_j}) \\
&\quad f_{R_j}(x) = f_{I_j}(x) \\
o_j &= o_{R_j} + j\, o_{I_j} \\
&= f_j(net_j) \\
e_j(n) &= d_j(n) - o_j(n) \\
\delta_j(n) &= e_{R_j}(n) f'_{R_j}(net_{R_j}(n)) + j\, e_{I_j}(n) f'_{I_j}(net_{I_j}(n)) \\
\delta_i(n) &= f'_{R_j}(net_{R_j}) Real[\sum_j \delta_j(n) w^*_{ij}(n)] + \\
&\quad f'_{I_j}(net_{I_j}) Imag[\sum_j \delta_j(n) w^*_{ij}(n)] \\
w_{ij}(n+1) &= w_{ij}(n) + \eta\, \delta_j o^*_i(n)
\end{aligned}
$$

where

x_i	is the complex input to the j^{th} neuron from neuron i i.e. it is the output of the i^{th} neuron, o_i
w_{ij}	is the complex weight linking neuron i to neuron j
net_j	is the complex activation of neuron j
$f_j()$	is the complex activation function for neuron j
$o_j(n)$	is the complex output of the j^{th} neuron at time n
$e_j(n)$	is the complex error between the desired response, $d_j(n)$, and the output of the j^{th} neuron, $o_j(n)$ at time n
*	denotes complex conjugation

In forming the complex neuron the variables used for the real valued neuron simply take on their complex forms and the rules for complex arithmetic are followed. The only problem is what form should the activation function take. For our work we define the complex activation function $f(net)$ as follows[7]:

$$f(net) = f_R(net_R) + j\, f_I(net_I) \tag{2.23}$$

[7]This activation function follows heuristically by applying an Adaline structure to the complex lms algorithm[18]. This function satisfies the criterion for a complex activation function as given in [143] which was published after this work was originally completed

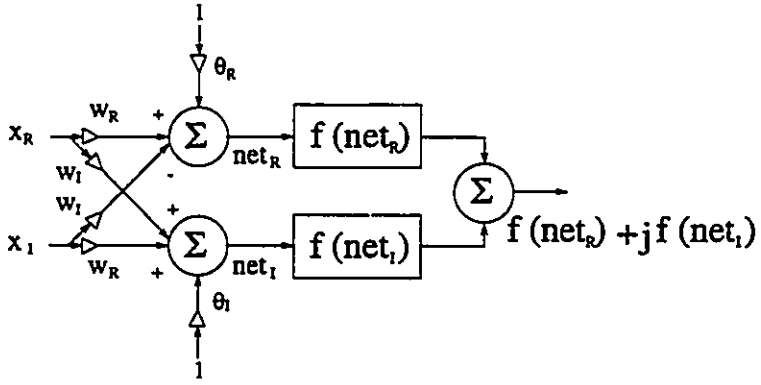

Figure 2.10: Diagram of a complex neuron.

Following the work of Widrow *et al*[18] we wish to minimize the average total error power, $E[ee^*]$. Using this criterion we can show that the following adaptation equations hold:

$$\begin{aligned}
\delta_j(n) &= e_{R_j}(n) f'_{R_j}(net_{R_j}(n)) + j e_{I_j}(n) f'_{I_j}(net_{I_j}(n)) \ \textit{output layer} \\
\delta_i(n) &= f'_{R_j}(net_{R_j}) Real[\sum_j \delta_j(n) w^*_{ij}(n)] + \\
&\quad f'_{I_j}(net_{I_j}) Imag[\sum_j \delta_j(n) w^*_{ij}(n)] \ \textit{hidden layers} \\
w_{ij}(n+1) &= w_{ij}(n) + \eta\, \delta_j o^*_i(n)
\end{aligned}$$

Similar equations can be derived when the momentum factor is considered. For our networks we found that the use of the momentum factor made little difference in the learning ability of the networks so it was set to zero.

In theory this complex neuron could be represented by two real valued neurons with the network trying to learn to form a complex neuron. This would, however, not allow for a frequency analysis of the weights connecting the tapped delay line to the first hidden layer of a neural network in an equalization structure[8]

[8]Simulations have been run using real valued neurons in a MLP equalizer. The results obtained from these networks agreed with those obtained from comparable complex valued neural networks.

Chapter 3

The AWGN Channel - Bayesian Estimation

3.1 Introduction

This chapter looks at the additive white Gaussian noise (AWGN) noise channel for PAM signalling and the problem of estimating the transmitted signal from the received signal. The goal of the estimator is to minimize the expected value of the square of the difference between the transmitted symbol and the estimated symbol. This is known as Bayesian estimation. In the AWGN channel this is a simple problem but it has some important implications in the area of neural networks. We show that a single neuron using the standard sigmoidal nonlinearity function is effectively a Bayesian estimator for binary (± 1) signalling in AWGN. This establishes the link between the weight(s) of the neuron and the noise variance associated with the binary valued signal. This relationship is then extended to neurons where the sigmoidal nonlinearity function incorporates a gain term [144, 145, 142].

In extending these results to multi-level signalling a new nonlinearity function is developed that incorporates a gain term α which simplifies the structure of the neural network. It is shown that this gain term is linked to the noise variance much like the gain term in a neuron using the sigmoidal nonlinearity for binary signalling. When α is small this function behaves as a soft limiter with a graded response bounded

by $(-1, +1)$, much like $tanh\,(x)$. However, when α is allowed to increase it develops into a stair case function similar to that of a hard quantizer. This behaviour is characteristic of a N-level (N > 2) Bayesian estimator when the noise variance is small. We show that this function can be used to reduce the complexity of the neural network and that provided that the gain term is back propagated it overcomes some learning/training problems that occur when the networks use linear output neurons.

This form of nonlinearity was also developed independently in [146]. In addition, Amit [147] discusses a ternary valued nonlinearity. Si and Michel [148] conduct an analysis and synthesis of discrete-time neural networks with multi-level threshold-functions and Banzhaf [149] considers a network of multistate units capable of associative memory and pattern classification.

Studying the relationship between these simple neural networks and Bayesian estimation provides for a more fundamental understanding of the behaviour and performance of neural networks and provides a basis for the work in the next chapter on neural network channel equalization. In the next section, the Bayesian estimation problem is outlined and the equivalence between a single neuron and the Bayesian estimator for binary signalling is established. This relationship provides a basis for modifying the activation function to include a gain term that is inversely related to the noise variance. These results are then extended to quaternary signalling and in general M-ary signalling. Section 3.3 describes the results of computer experiments to train neural networks as Bayesian estimators and section 3.4 provides a summary of the work presented in this chapter.

3.2 Bayesian Estimation

Let us consider the reception of a PAM signal, $s(t)$ that is corrupted by additive white Gaussian noise, (AWGN). The received signal, $x(t)$, is given as

$$x_k = s_k + n_k,\ s_k \in \{\pm 1, \pm 3, \ldots, \pm M - 1\} \tag{3.1}$$

For x_k we wish to find the optimum (Baye's minimum mean square error) [150] estimator $\hat{s}_k$ of s_k, $\hat{s}_k = B(x_k)$, that is we seek to minimize $E\,(s(t) - \hat{s}(t))^2$. Let us

define the following:

1: the noise is AWGN with zero mean and variance σ^2 such that

$$p_n(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}} \tag{3.2}$$

2: s_i, s_j, and s_k are iid random variables where

$$s_i,\ s_j,\ \text{and}\ s_k \in \frac{1}{N-1} \begin{cases} \pm 1 & \text{for 2-PAM} \\ \pm 1, \pm 3 & \text{for 4-PAM} \\ \pm 1, \pm 3, \pm 5, \pm 7 & \text{for 8-PAM} \\ \pm l_1, \pm l_2, \pm l_3, ..., \pm l_{N/2} & \text{for } N\text{-PAM},\ N > 8 \end{cases} \tag{3.3}$$

$$p_s(z) = \begin{cases} \frac{1}{N} & \text{for } z \in \pm l_1,\ \pm l_2,\ \pm l_3,\ \ldots\ ,\pm l_{N/2} \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

Because of the independence of the x_k's

$$\begin{aligned} \hat{s}_k &= E\{s_k | x_k\} \\ \hat{s}_k &= \int_{\underline{\xi}} s_k p_s(s_k|x_k) ds_k \end{aligned} \tag{3.5}$$

where the range of integration is the set $\underline{\xi}$ of all possible values of s_k.
Using Baye's theorem

$$\hat{s}_k = \int_{\underline{\xi}} \frac{s_k\, p_x(x_k|s_k)\, p_s(s_k)}{p_x(x_k)} ds_k \quad (-\infty < s_k < \infty) \tag{3.6}$$

Because the s_k are discrete the above integral becomes

$$\begin{aligned} \hat{s}_k &= \sum_{\underline{\xi}} \frac{s_k\, p_x(x_k|s_k)\, p_s(s_k)}{p_x(x_k)} \\ &= \frac{1}{N} \sum_{i=1}^{N} \frac{s_i\, p_x(x_k|s_i) \sum_{j=1}^{N} \delta(s_i - s_j)}{p_x(x_k)} \end{aligned} \tag{3.7}$$

where $s_j \in \underline{\xi}$

We now need expressions for $p_x(x_k|s_k)$ and $p_x(x_k)$. Since the received signal consists of the desired signal s_k with the addition of AWGN we obtain:

$$p_x(x_k|s_k) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(\frac{-(x_k - s_k)^2}{2\sigma^2}\right)} \tag{3.8}$$

Note that $p_x(x_k|s_k)$ is Gaussian with mean s_k and variance σ^2. For $p(x_k,\, s_k)$:

$$\begin{aligned} p(x_k,\, s_k) &= p_{x,s}(x_k, s_k) \\ &= p_x(x_k|s_k)\, p_s(s_k) \\ &= \frac{1}{\sqrt{2\pi}\sigma} e^{\left(\frac{-(x_k-s_k)^2}{2\sigma^2}\right)} \frac{1}{N} \sum_{j=1}^{N} \delta(s_k - s_j) \\ p_x(x_k) &= \int_{\underline{\xi}} p_{x,s}(x_k, s_k)\, ds_k \\ &= \sum_{\underline{\xi}} p_{x,s}(x_k, s_k) \\ &= \frac{1}{N\sqrt{2\pi}\sigma} \sum_{i=1}^{N} e^{\left(\frac{-(x_k-s_i)^2}{2\sigma^2}\right)} \end{aligned} \tag{3.9}$$

Substituting equations (3.8) and (3.9) back into equation (3.7) we obtain:

$$\begin{aligned} \hat{s}_k &= \frac{1}{N} \sum_{i=1}^{N} \frac{\frac{s_i}{\sqrt{2\pi}\sigma} e^{\left(\frac{-(x_k-s_i)^2}{2\sigma^2}\right)} \sum_{j=1}^{N} \delta(s_i - s_j)}{\frac{1}{N\sqrt{2\pi}\sigma} \sum_{l=1}^{N} e^{\left(\frac{-(x_k-s_l)^2}{2\sigma^2}\right)}} \\ &= \frac{\sum_{i=1}^{N} s_i\, e^{-\frac{(x_k-s_i)^2}{2\sigma^2}}}{\sum_{l=1}^{N} e^{-\frac{(x_k-s_l)^2}{2\sigma^2}}} \end{aligned} \tag{3.10}$$

If we consider 2-PAM, s_k can take on the values of ± 1 and this expression simplifies to

$$\begin{aligned} \hat{s}_k &= B(x_k) \\ &= \frac{e^{-\frac{(x_k-1)^2}{2\sigma^2}} - e^{-\frac{(x_k+1)^2}{2\sigma^2}}}{e^{-\frac{(x_k-1)^2}{2\sigma^2}} + e^{-\frac{(x_k+1)^2}{2\sigma^2}}} \\ &= tanh\left(\frac{x_k}{\sigma^2}\right) \end{aligned} \tag{3.11}$$

which is shown in Figure 3.1. As can be seen, this is a limiting type of function with σ^2 controlling the "hardness" or steepness of the function. When the noise level is sufficiently small the estimator function effectively becomes a hard quantizer, $+/-1$. Note that the *tanh* function is a scaled and shifted version of the sigmoidal function,

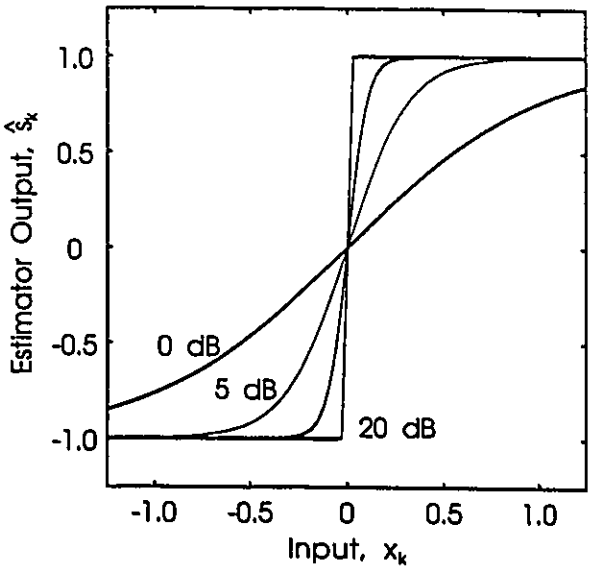

Figure 3.1: The Bayesian estimator function for 2-PAM with SNR's of 0, 5, 10 and 20 dB.

that is

$$\begin{aligned} tanh(\frac{x}{2}) &= \frac{e^{\frac{x_k}{2}} - e^{-\frac{x_k}{2}}}{e^{\frac{x_k}{2}} + e^{-\frac{x_k}{2}}} \\ &= \frac{2}{(1+e^{-x})} - 1 \\ &= 2\,logistic(x) - 1 \end{aligned} \tag{3.12}$$

Thus the Bayesian estimator function for 2-PAM appears to be a single MLP with a *tanh* nonlinearity and a single input weight, w, of

$$w = \frac{1}{\sigma^2} \tag{3.13}$$

such that

$$\begin{aligned} \hat{s}_k &= tanh(w\,x_k) \\ &= tanh(\frac{x_k}{\sigma^2}) \end{aligned} \tag{3.14}$$

It is interesting to note that the requirement for training a MLP to estimate s_k from x_k is that the network minimizes $E(s(t) - \hat{s}(t))^2$ which means that the goal of the

network is to form the Bayesian estimator function for s_k. We have just shown a single neuron is capable of doing this by using a single weight as given by equation (3.13) when its threshold value θ is set to zero. This establishes a link between a single neuron and the Bayesian estimator for 2-PAM. What happens if we wish to use more than two signal levels? The Bayesian estimator for 4-level PAM is shown in Figure 3.2 and is given as

$$\hat{s}_k = B(x_k) = \frac{\sum_{i=1}^{4} s_i \, e^{-\frac{(x_k - s_i)^2}{2\sigma^2}}}{\sum_{j=1}^{4} e^{-\frac{(x_k - s_j)^2}{2\sigma^2}}} \tag{3.15}$$

where

$$s_i, \, s_j \in \{-1, \, -1/3, \, 1/3, \, 1\}$$

Notice that when the noise variance, σ^2, is large (SNR of 0 dB) this function behaves as a soft limiter, much like the *tanh* function, yet when the noise variance is small (SNR of 20 dB) this function acts as a hard quantizer, i.e. it is a stair case function as shown in Figure 3.2. In forming an estimate of s_k for 4-level PAM we are interested

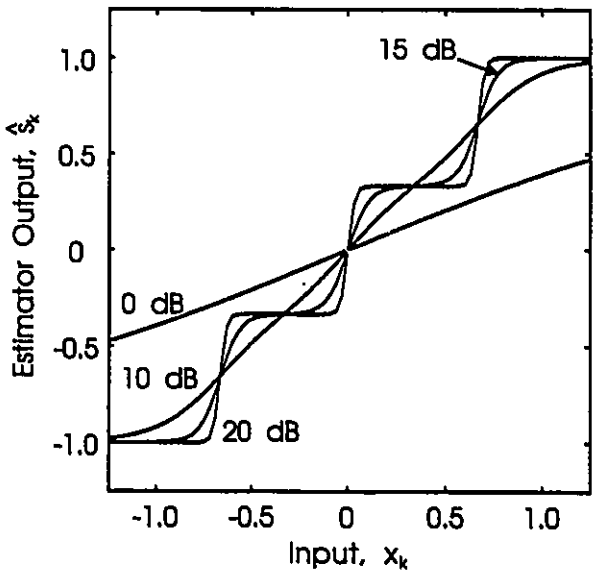

Figure 3.2: The Bayesian estimator function for 4-PAM with SNR's of 0, 10, 15 and 20 dB.

in knowing the minimum network size. This may be done intuitively by considering

the case where the noise variance is small which results in the Bayesian estimator function acting as a hard quantizer. For 2-level PAM there is only a single step, while for 4-level PAM there are 3 steps. Each of these steps may be approximated with a single $tanh(x)$ function that is characteristic of the output nonlinearity function of a single neuron. Thus 2-PAM would require a single nonlinear neuron and 4-PAM would require 3 nonlinear neurons. In general N-PAM would have N-1 steps, requiring N-1 neurons.

All that is left to do is to chose appropriate weights and thresholds for the neurons and to linearly combine their outputs. The weights and thresholds may be obtained through network training and a neuron with a linear activation function is chosen for the linear combiner. Thus the minimum network size for 4-level PAM is a network of three nonlinear hidden neurons with a single linear output neuron. This is denoted as a $3\,0\,1\,l$ network. A network is labelled as $n_1 n_2 n_{out}\alpha$ where: n_1 is the number of neurons in the 1^{st} hidden layer, n_2 is the number of neurons in the 2^{nd} hidden layer, n_{out} is the number of output neurons and $\alpha \equiv nl$ corresponds to a network with nonlinear output neurons while $\alpha \equiv l$ denotes a network with linear output neurons. Note that in the case of 2-PAM there is no need to have a network with a linear output neuron, i.e. a $1\,0\,0\,nl$ network is sufficient. These networks are shown in Figure 3.3a-b.

An alternative to this type of network would be to consider a single neuron with a multi-level nonlinearity function, as in Figure 3.3c. A suitable first choice for such a function might be the Bayesian estimator function itself as given in equation 3.15 for 4-level PAM where a gain term α would be used instead of $1/\sigma^2$. Unfortunately this function has several problems associated with it. The first is that α would have to be fixed in order to insure the necessary degree of hardness for the activation function. The second is that this function is expressed in the form of a quotient. Thus its derivative, which is needed for network training, will be quite complicated. For practical purposes it is desirable to have a function in which both it and its derivative are easily evaluated, i.e. they are not computationally complex. This function might

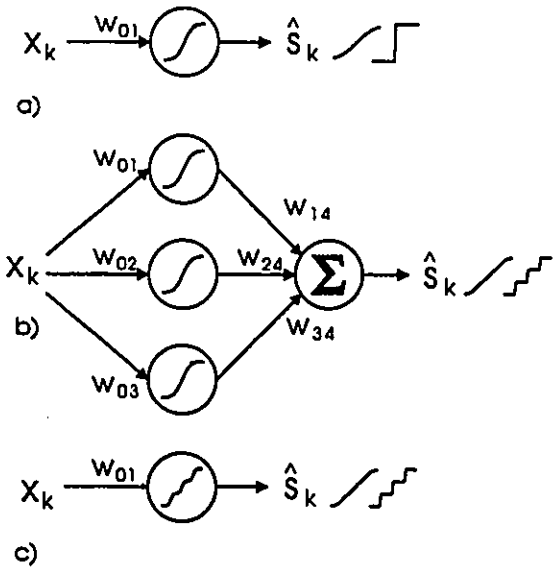

Figure 3.3: These are the three neural network architectures used in estimating the Bayesian estimator function, a) 100nl network with a *tanh* nonlinearity, b) 301l network and c) 100nl network with a multi-level nonlinearity.

still be used if its derivative [129] is approximated by

$$f'(x) = \frac{f(x) - f(x - \Delta x)}{\Delta x} \tag{3.16}$$

as $f(x)$ is easily evaluated.

A better alternative is to chose a function of the following form which has the desired multi-level limiting characteristics:

$$f(\alpha\, x) = \frac{1}{N-1} \sum_{i=1}^{N} tanh\left(\alpha\,(x - \theta_i)\right) \tag{3.17}$$

where $N+1$ is the desired number of signal levels and the thresholds or biases, θ_i's, are the mid-points between adjacent signal points in the signal set, e.g. $\{\pm 2/3,\, 0\}$ for 4-PAM. This function is shown in Figure 3.4 and may be compared to the Bayesian estimator function shown in Figure 3.2 for 4-PAM. In essence we have taken $N-1$ neurons, fixed the bias terms to a desired level and linearly combined their outputs to form a single neuron. The derivative with respect to the activation of the j^{th} neuron,

net_j is easily calculated as

$$\frac{\partial f_j(\alpha net_j)}{\partial net_j} = \frac{1}{N-1}\sum_{i=1}^{N}\frac{\partial tanh\,(\alpha net_{ji})}{\partial net_{ji}} \tag{3.18}$$

where

$$net_{ji} = net_j - \theta_i$$

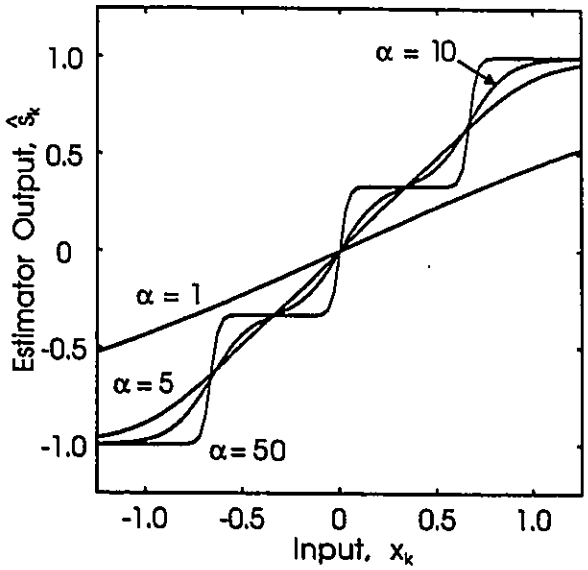

Figure 3.4: The multi-level nonlinearity function for α's of 1, 5, 10, and 50.

There is an inherent problem with this type of function, namely, that if the gain term α is chosen high enough the function will exhibit multiple plateaus. At these points the derivative of $f(\alpha net_j)$, $\partial f_j(\alpha net_j)/\partial net_j$, approaches zero. This poses a learning problem as the change in weights of a neuron is directly proportional to $\partial f_j(\alpha net_j)/\partial net_j$. In order for neurons not to get stuck on or in between plateaus the network should be started off with a small value of α and trained for a long enough period of time to allow the neurons to properly distribute their outputs. Once this has been achieved the network may then be hardened. This suggests that it might be advisable to back propagate the α parameter in order to "harden" the nonlinearity function in the neurons. The change in α can be calculated as (Appendix B)

$$\Delta\,\alpha_j \propto \left(\sum_k \delta_k w_{kj}\right)\frac{\partial f_j(\alpha_j net_j)}{\partial \alpha_j} \tag{3.19}$$

If α is kept small, then when the weights grow large the resulting nonlinearity function will be a binary hard quantizer much like that of the $tanh(\alpha x)$ function. This may seem to be a problem yet it works to our advantage. It implies that if the multi-level nonlinearity is placed in an existing neural network the performance of the neural network will be almost identical to when the network is employing the $tanh(x)$ function provided that α is kept small (i.e. 1). Only if α is allowed to increase will the network take on a new form.

An interesting problem is to consider what will happen as the number of signal levels is increased. In the limit a continuous uniform distribution of signals in the interval of $[-1, 1]$ is obtained. The functional form of this estimator is extremely complex and will not be detailed here but its form is shown in Figure 3.5. Note that at low signal to noise ratios this function looks sigmoidal in nature yet at high SNR the function is linear in the region within $[-1, 1]$. At high SNR this is to be expected, as the noise level is small and the signal is uniformly distributed in this interval.

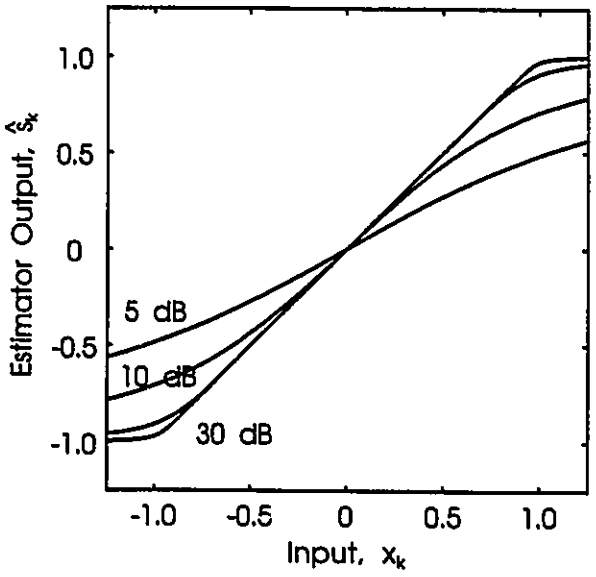

Figure 3.5: The Bayesian estimator function for a uniformly distributed signal with SNR's of 5, 10, 20 and 30 dB.

As an aside, there are several other interesting interpretations of the $tanh(\alpha x)$ function. In particular, Amit[147] shows that this type of activation function can be

related to: the mean firing rates of biological neurons, the firing probability of a two-state neuron operating in a noisy environment, and the probability of a magnetic moment changing its spin in the Ising spin glass model. For a more detailed treatment and interpretation of the functions the reader is referred to Amit[147].

3.3 Computer Simulations

3.3.1 PAM Simulation

In this experiment it was desired to see if a neural network could learn to become a Bayesian estimator for the PAM signal, s_k. The training pattern consisted of a data signal, s_k, perturbed by AWGN, n_k, such that the input to the network at time k, x_k, is given as $x_k = s_k + n_k$. The training sequence consisted of 1000 samples of x_k where s_k was chosen randomly from $\{\pm 1\}$ for 2-PAM and $\{\pm 1, \pm 1/3\}$ for 4-PAM. These levels were chosen in order avoid problems associated with input scaling of the networks.

The networks considered for evaluation are based upon the minimum number of neurons required to perform the appropriate Bayesian estimation function as was discussed in the previous section. For 2-PAM, a $1\,0\,0\,nl$ network was used with a fixed gain term of $\alpha = 1$. For 4-PAM, two networks were used: a $3\,0\,1\,l$ network with a fixed gain term and a $1\,0\,0\,nl$ network that utilized the multi-level nonlinearity function of equation 3.17. The multi-level nonlinearity was started with a gain of $\alpha = 1$ which was allowed to adapt along with the rest of the network parameters. In training the networks, the learning constant,η, was set to 0.001 for the weight and bias terms and 0.1 for the gain terms (when α was allowed to adapt). The network was allowed to adapt after the presentation of each input/output pair.

Both of the $1\,0\,0\,nl$ networks started off with "soft" nonlinearity functions and were allowed to adapt until there was very little change in the mean squared error (MSE) performance of the network. In the case of the $3\,0\,1\,l$ network several values of α (1, 10, 50 and 100) were tried and the network was allowed to adapt until there was little change in the MSE performance. That is, the network had effectively

minimized the MSE between s_k and $\hat{s}_k$. Unfortunately this does not imply that the networks became Bayesian estimators for s_k. Rather, the networks only adjusted themselves to fit the training patterns. In the case where the signal to noise ratio was low (SNR = 0 dB) the training patterns are well distributed over the interval from $[-1, 1]$, however, when the SNR is high the noise level is low and the training patterns are concentrated around $\{\pm 1\}$ for 2-PAM and $\{\pm\frac{1}{3}, \pm 1\}$ for 4-PAM. Hence there is no information contained in the training patterns about the other regions of the Bayesian estimator function. Thus at high signal to noise ratios the networks failed to "harden" up as they only adjusted themselves enough to pass through the data points contained within the training set. This is shown in Figures 3.6 and 3.7 for both of the $100nl$ networks (2-PAM and 4-PAM). By hardening we refer to the network's ability to form the hard quantization levels that are characteristic of the Bayesian estimator function when the noise variance is small. By softening we refer to the network's ability to display a graded response like that of a soft limiter that is bounded by $(-1, 1)$.

It was found that the only way to obtain a Bayesian like response from any of these networks was to choose a gain term of α close to $\frac{1}{\sigma^2}$. In general, σ^2 is seldom known apriori, making it difficult to choose a suitable value for α. One way to over come this would be to choose an α much greater than necessary and allow the network to adapt α. While this works for the $100nl$ networks considered it does not work well for the $301l$ network due to the linear output neuron. Allowing the weights of the output neuron to adapt creates too many degrees of freedom for the network. Effectively, changes in the weights of the linear output neuron tend to counteract the changes in the hidden nonlinear neurons. The problem is that the weights of the nonlinear neurons directly determine the Bayesian behaviour of the network yet they are hidden from the output of the network by the linear output neuron. Because of the back propagation algorithm the weights of the linear output neuron adapt faster than the weights of the nonlinear hidden neurons. Hence they tend to counter-act the development of the Bayesian behaviour of the network.

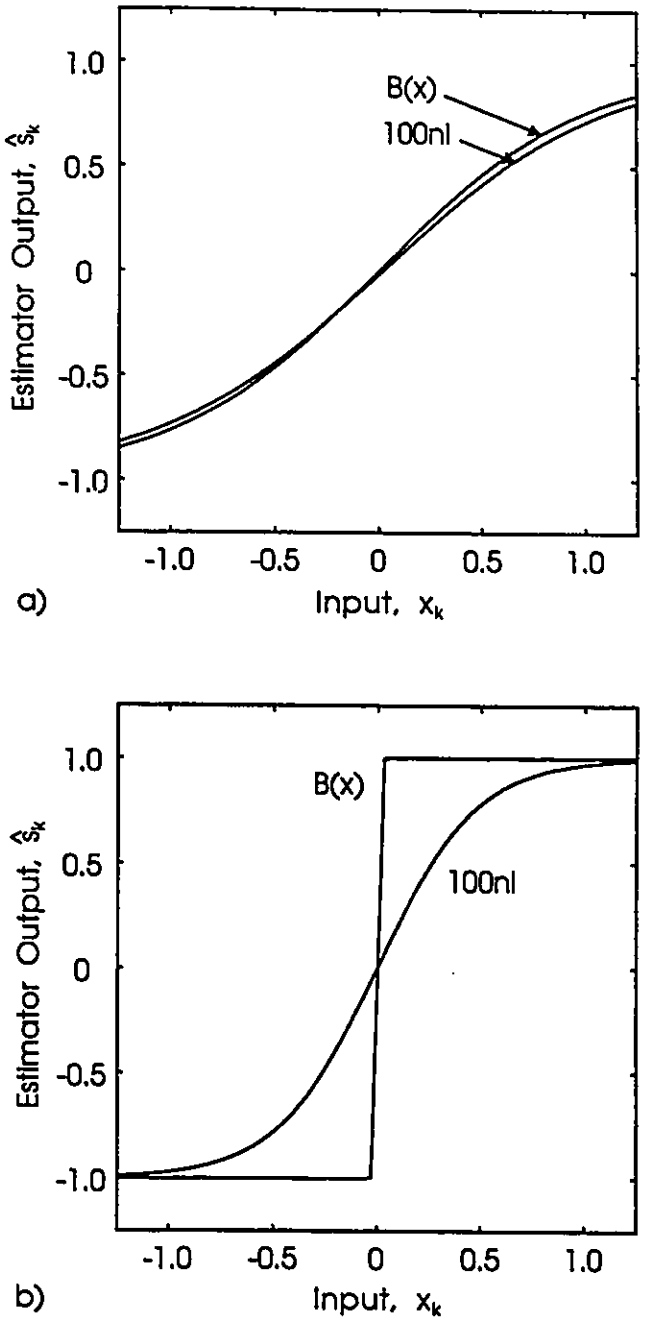

Figure 3.6: These two graphs show the input/output relationship of the 100nl network after training with a 2-PAM signal. The SNR ratios used are: a) 0 dB and b) 20 dB.

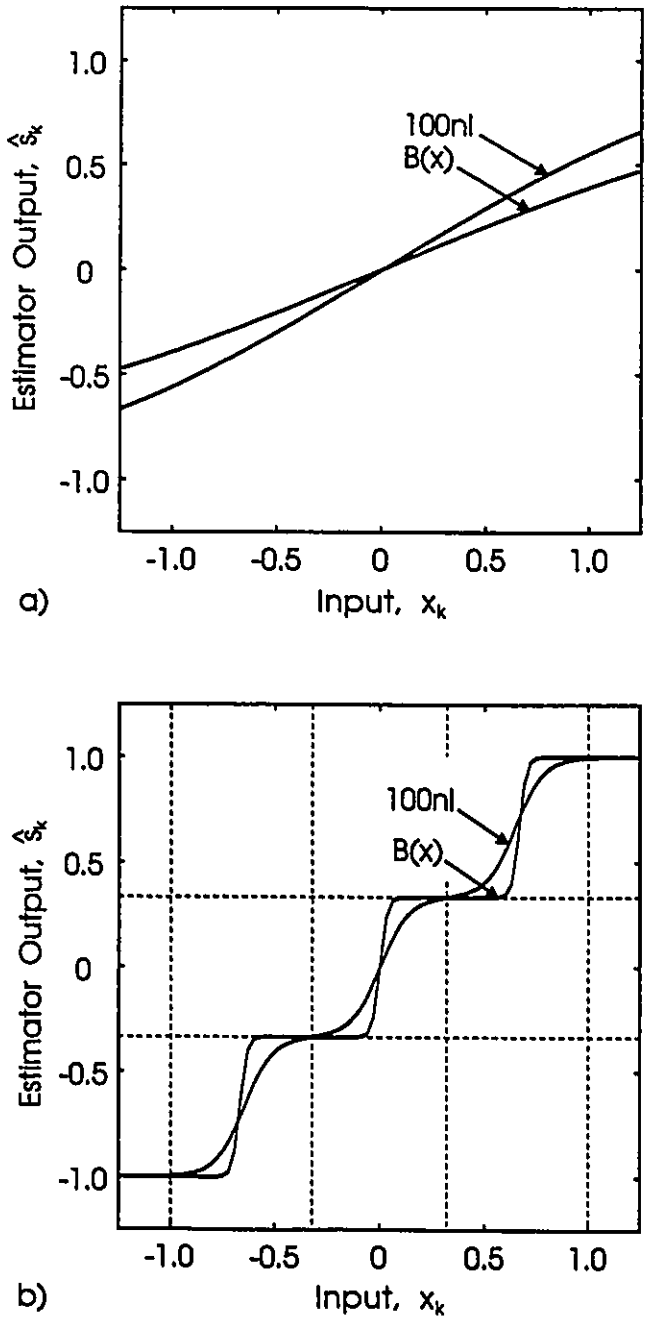

Figure 3.7: These two graphs show the input/output relationship of the 100nl network after training with a 4-PAM signal. The SNR ratios used are: a) 0 dB and b) 20 dB.

3.3.2 Uniformly Sampled Bayesian Estimator Function

In this simulation it was desired to see how well the network could approximate the Bayesian estimator function, $B(x)$. The networks were trained on 300 uniformly spaced samples of $B(x)$ where x ranged over the interval $[-1.5, 1.5]$. These functions are given by equations 3.12 and 3.15 for 2 and 4-PAM respectively and were evaluated for signal to noise ratios of 0, 10 and 20 dB. In addition, a SNR of 30 dB was used for 2-PAM. This could not be considered for 4-PAM due to numerical precision problems. The networks that were evaluated were the same as those used in the previous section, namely the $100nl$ network for 2-PAM and the $301l$ and $100nl$ networks for 4-PAM.

From the simulations, both of the $100nl$ networks (for 2-PAM and for 4-PAM) converged to functions that were near optimum. The functions for 4-PAM are shown in Figure 3.8 for signal to noise ratios of 0 and 20 dB. Obviously the output function of the $100nl$ network is very close to the desired function. In the case of the $301l$ network, the same problems that were detailed in the previous section plagued the performance of this network. Namely, that in order for the $301l$ network to approximate the desired Bayesian estimator function it was necessary to choose the gain term, α, sufficiently close to $\frac{1}{\sigma^2}$. This meant that it was necessary to experiment with the adaptation constant η in order to obtain satisfactory performance. When the gain term was allowed to adapt for the nonlinear hidden neurons, improved performance was obtained although nowhere near the performance obtained from the $100nl$ multi-level nonlinearity network. As was the case in the previous section, this was due to the adaptation of the weights of the linear output neuron.

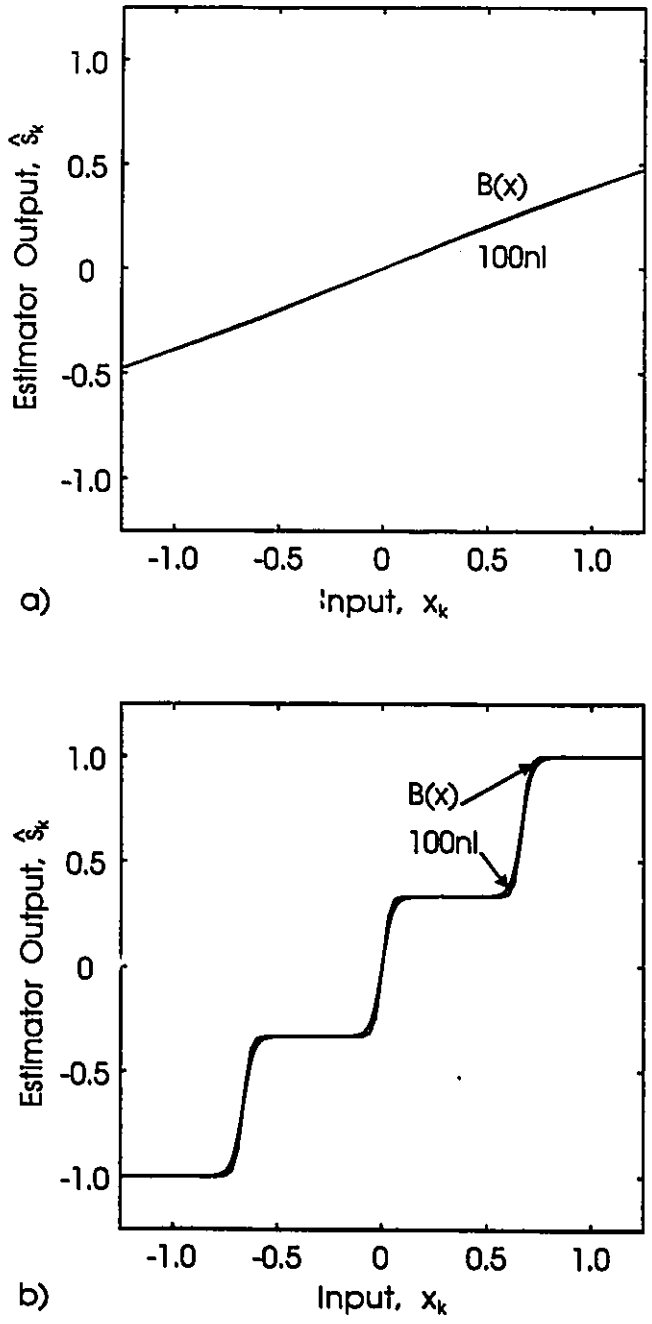

Figure 3.8: These two graphs show the input/output relationship of the 100nl network for 4-PAM after training with uniformly spaced samples of the Bayesian estimator function for SNR ratios of: a) 0 dB and b) 20 dB.

3.4 Summary

In this chapter we have established a link between a single neuron and the Bayesian estimator for 2-PAM in AWGN. Stemming from the Bayesian estimator function for N-PAM a new activation function was presented that exhibits multiple (> 2) quantization levels when its gain parameter, α, is chosen sufficiently large yet when α is small it behaves as a soft limiter much like the $tanh(x)$ function. In order for a network to learn when using this function the network should be initiated with a small value of α so that the neural outputs display a soft limiting function. As the neurons begin to distribute their outputs over the activation function it is then possible to increase α, i.e. to harden the activation function. This leads to the need to back propagate the α parameter. With regards to back propagating α, there has been some work done for binary neurons, $tanh(\alpha x)$, that indicates this leads to faster learning times[142]. What we have shown is that in this case, adapting the gain term amounts to adapting the adaptation parameter η for each neuron in the neural network. In the case of a multi-level nonlinearity, back propagating the gain term leads to an effective change in the adaptation parameter as well as changing the characteristics of the nonlinearity function.

Chapter 4

Neural Network Channel Equalization

4.1 Introduction

The goal of using neural networks is to achieve the performance of the optimal signal detector or estimator without the complexity associated with these structures. While neural networks are more complex and exhibit slower learning times than linear equalizers they offer the advantage of being able to form nonlinear mappings and hence should be able to achieve improved performance. With this in mind, the work in this chapter extends the work of Gibson and Cowan *et al* [28]-[38] from 2-PAM to 4 and 16-QAM (from two levels in one dimension to multiple levels in two dimensions) and includes the effects of pulse shaping. In particular a widely accepted (and perhaps more realistic) channel model is used - the Rummler multi-path fading channel model[41] for the DMR channel.

The next section discusses signal detection and estimation and includes work by Gibson and Cowan *et al* [28]-[38] to show why the optimum equalizer is a nonlinear one. This is followed by a section on linear equalization theory which helps to put the work of Gibson and Cowan in the proper context. The following section details the neural network equalizer structure. This is followed by a section on simulation results and a summary of the work presented in this chapter.

4.2 Signal Detection and Estimation

For the received signal given by equation (4.1), (section (2.2), equation (2.4)) the optimum signal detector is a maximum likelihood sequence estimator [26] which at time k detects the entire data sequence $\{s_k, \ldots, s_0\}$ based upon the entire received sequence $\{r_k, \ldots, r_0\}$ where

$$r(kT) = s_k h(t_0) + \sum_{j \neq k} s_j h(t_0 + kT - jT) + n(t_0 + kT) \tag{4.1}$$

This receiver is quite complex as it must search through all the possible combinations of transmitted signals to determine the most likely combination that resulted in the received sequence. Such a receiver will not be detailed here, rather what is desired is a sub-optimum structure whose performance may be considered close to optimum. Note that it is possible to take multiple samples per symbol of the received signal, however, we shall restrict ourselves to one sample per symbol. The format that we consider is symbol by symbol detection (or estimation) based upon a small subset of the entire received sequence. We wish to detect or estimate s_{k-d} given the m-element received signal vector $\mathbf{x}_k^T = \{r_k, \ldots, r_{k-d}, \ldots, r_{k-m+1}\}$ where d is a suitably chosen decision delay, i.e. at time k, s_{k-d} is detected, not s_k. The receiver structure is shown in Figure 4.1. Aside from the processor function the parameters that we are free to choose are the number of received signal samples m, the spacing between the received samples and the decision delay. All of these affect the performance of the system. Once these factors have been set it is up to the processor unit to make the most of the situation. If the channel response is known it is possible to determine the optimum processor function. Generally the channel response is not known so it is desirable to have a network (e.g. neural network) that can be trained to learn the desired processing function. Hence, an error signal e_k has been included in the system so that it is possible to train the processor and to allow the processor to adapt to time varying changes in the channel response.

If the output of the processor is continuously valued or soft quantized then the equalizer structure is based upon an estimation type of structure as it attempts to produce an estimate of the transmitted symbol. If the output is hard quantized

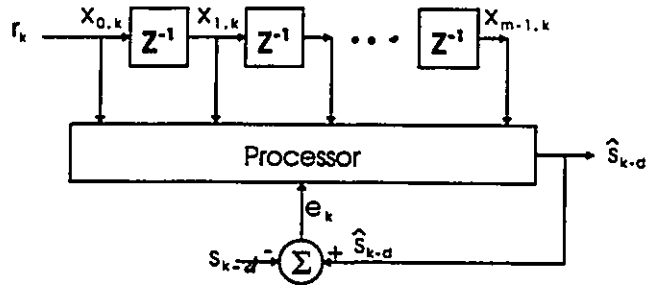

Figure 4.1: General Equalization Structure

then the equalizer is based on a detection type of structure, that is, its goal is to detect which one of the m signals out of the m-ary alphabet was transmitted. By soft quantization we mean that if the output symbol is represented by binary bits then more than $\log_2(m)$ bits are used, whereas in the case of hard quantization only $\log_2(m)$ bits or m decision lines are used.

The rationale for choosing an estimation or a detection type of structure is arbitrary but is influenced by the architecture following the equalizer. In the case of convolutional coding and trellis coded modulation (TCM) [151] the decoding structure following the equalizer is based upon soft decisions at the output of the equalizer. Hard decisions could be used but this results in a stiff penalty in SNR (signal to noise ratio) performance of about 2-2.5 dB between hard quantization and soft decisions (based upon three bits of quantization per symbol interval). This means that we are interested in equalization structures which make soft decisions on the output symbol as well as structures which make hard decisions.

4.2.1 Bayesian Detection

To determine the optimum signal detector for a finite impulse response channel filter, $H(z) = \sum_{i=0}^{n} h_i z^{-i}$, we consider the optimum signal detector for the m-element tap input vector $\mathbf{x}_k$ where $\mathbf{x}_k^T = [r_k, \ldots, r_{k-m+1}]$ and

$$r_k = \sum_{j=0}^{n} s_{k-j} h_j + n_k \tag{4.2}$$

This detector is based upon Bayesian detection theory [152][36][37]. To determine the Bayesian detector all of the possible states of the tap input vector $\mathbf{x}_k$ must be considered. Let $\mathbf{S}_\theta$ denote the set of all possible combinations of the channel input sequence $[s_k, s_{k-1}, \ldots, s_{k-d}, \ldots, s_{k-m+1-n}]$ where the desired signal $\hat{s}_{k-d} = \theta$. Denote the resulting set of states for the received signal vector $\mathbf{x}_k$ for the case of noiseless transmission as $\mathbf{X}_\theta$ and let $\mathbf{X}_{j,\theta}$ represent the j^{th} element of this set. The M Bayesian decision variables corresponding to the M possible values for $\hat{s}_{k-d}$ are computed as

$$d_{\theta,k-d} = \sum_{j=1}^{M^{n+m-1}} \frac{p(\mathbf{X}_{j,\theta})}{\sqrt{2\pi\sigma_n^2}} \exp\left[\frac{-\|\mathbf{x}_k - \mathbf{X}_{j,\theta}\|^2}{2\sigma_n^2}\right] \tag{4.3}$$

The detected symbol is chosen such that $\hat{s}_{k-d} = \hat{\theta}$ where $d_{\hat{\theta},k-d} = \max(d_{\theta,k-d})$. The resulting decision boundaries for $m = 2$, $d = 0$ and $s_k \in \{\pm 1\}$ are shown in Figures 4.2a, b for the two channels, $H_1(z) = 1.0z^0 + 0.5z^{-1}$ and $H_2(z) = 0.5z^0 + 1.0z^{-1}$. These two channels have been used extensively by Gibson and Cowan *et al* [28]-[38] to illustrate the nonlinear behaviour of the optimum equalizer. In both cases the decision boundary, the line separating the regions where $s_{k-d} = 1$ and $s_{k-d} = -1$, is nonlinear but only in the first case, $H_1(z)$, are the decision regions linearly separable as shown in Figure 4.2.

$H_1(z)$ and $H_2(z)$ are known as minimum and maximum phase channels respectively. A minimum phase filter in the z-domain has all of its zeros located within a unit circle centered at the origin in the z-domain whereas a maximum phase filter has all of its zeros outside of this unit circle. In the multi-path channel, minimum phase indicates that the signal from the principle path arrives first as opposed to non-minimum phase where the signal from one or more of the secondary paths arrives first at the receiver. The principle signal path or ray is usually taken to be that path with the greatest signal strength. The significance in the phase of the channel is in how the decision delay should be chosen. This will be described further in section 4.3.

The main problems with this structure are that it requires knowledge of the channel impulse response to calculate the channel states and its complexity. While the complexity grows linearly with the number of states of $\mathbf{x}_k$ the number of states grows exponentially with the number of non-zero ISI terms, n, of equation (2.4) and

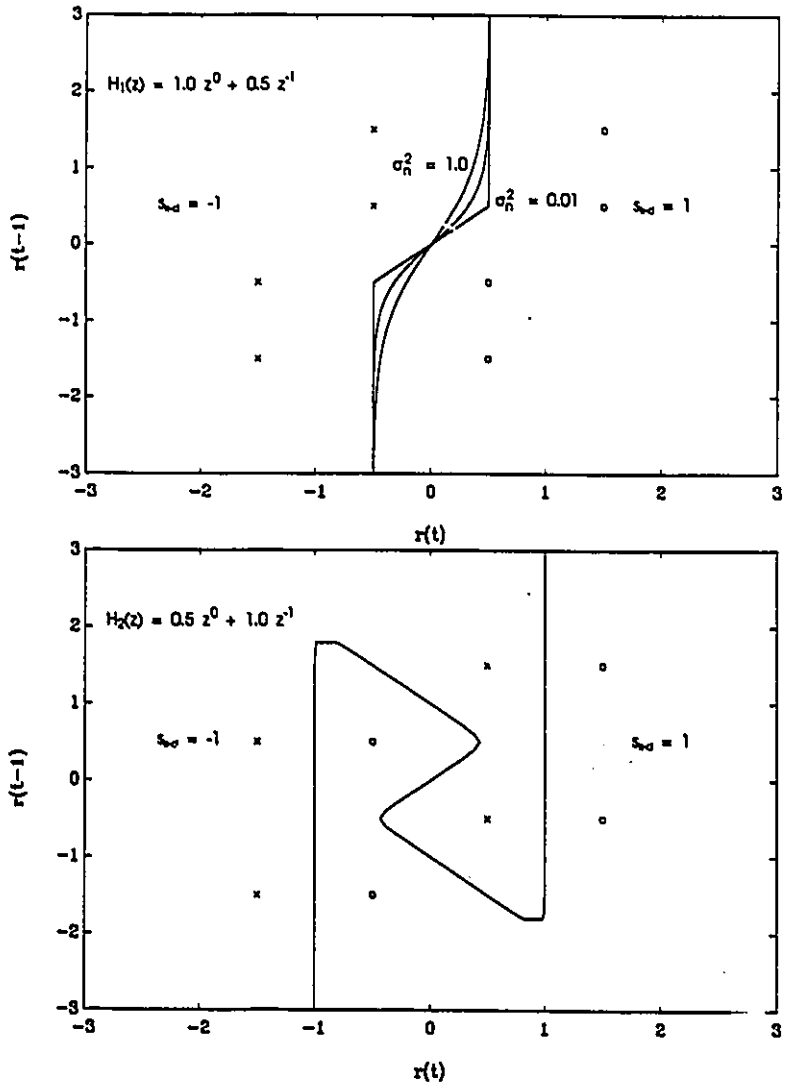

Figure 4.2: The decision boundaries for $H_1(z)$ and $H_2(z)$ for $m = 2$ and $d = 0$. Note that σ_n^2 is 0.01 for $H_2(z)$.

the length of the tap input vector $\mathbf{x}_k$, m. The total number of possible states for $\mathbf{x}_k$ is M^{m+n}. In the case of pulse shaping, e.g. a raised cosine pulse, the number of states can be quite large. Rather than evaluate equation (4.3) directly it is desired to train a neural network to approximate the action of the Bayes signal detector. This has been the basis for much of the work of Gibson and Cowan *et al* [28]-[38].

In mapping the M possible values for s_{k-d} to the output of a neural network there are two possible choices. The first choice is to use a network with M output nodes, each node representing one of the decision variables. The detected symbol is then chosen as in the case of the Bayesian detector. The network is trained by indicating that the desired node should be at a high level, e.g. $+1$, and that the other nodes should be at a low level, e.g. -1. There has been some indication [153]-[156] that for this type of network the activation levels of each of the M output neurons correspond to the probabilities that the symbol represented by the neuron was transmitted

$$d_{\theta,\,k-d} \sim P\left(s_{k-d} = \theta\right)$$

If this true then in the case of coded transmission it may be advantageous to feed the outputs of the neural network to a decoder rather than make hard decisions on the signal. This would correspond to soft decision decoding and may allow the system to achieve some of the coding gain that may otherwise be lost by the use of hard decisions on the signal. At present, no one has considered the effect of neural network equalizers on code performance.

An alternative mapping is to label each of the M values for s_{k-d} with a binary number with $\log_2(m)$ bits of ± 1's. The advantage of this is it reduces the number of output nodes for the network, however, the activation levels will no longer correspond to probabilities. For binary level signalling $\{\pm 1\}$ this produces a network with a single output where the desired output is ± 1 which is equal to s_{k-d}. The network thus acts as an estimator for the transmitted signal.

4.2.2 Effects of Pulse Shaping

As has already been alluded to in the previous section 4.2.1 and in section 2.3, the pulse shaping filters for the data pulse have a significant effect on the performance of the system. Figure 4.3 depicts the impulse response, $p(t)$, and the normalized spectrum, $2\,BT\,P(f)$, for four different pulse shapes: the rectangular (RECT) pulse, the raised cosine (RC) pulse and the spectrally raised cosine (SRC) pulse with no roll off and with a roll off of 100 percent. These pulses all meet the Nyquist requirement for zero inter-symbol interference (ISI), i.e. p(t) is zero for all nT where n is not equal to zero. For the rectangular pulse, $p(t)$ is zero for $|t| > T/2$. For the raised cosine pulse $p(t)$ is zero for $|t| \geq T$ and for the spectrally raised cosine pulse $p(t)$ is zero for $t = nT$, $n \neq 0$. Provided that the receiver samples at the appropriate sampling instant, $t = nT$, then, for all of these pulse shapes, there will be zero ISI at the sampling instant. However, if the sampling time is off or if there channel is non-ideal then ISI may occur. The effects of pulse shaping can best be seen with the aid of an eye diagram. The eye diagrams resulting from the use of these pulse shapes with binary level (± 1) signalling are shown in Figure 4.4. Note that for clarity, the appropriate sampling instants have been circled.

An eye diagram is a method of displaying the received signal, $r(t)$ by displaying samples of the received signal $r(t)$ in an overlapping manner, e.g. by overlapping samples of $r(t - kT)$ taken over two symbol intervals where

$$(k - 1/2)T \leq t \leq (k + 2 - 1/2)T,\ k = 0,\ 1,\ \ldots$$

and plotting the result the eye diagrams of Figure 4.4 are obtained. For these diagrams the required signal for $r(t)$ was generated by digital simulation using 8 samples per baud. Hence, two symbol intervals corresponds to 16 samples of the received signal.

One way to interpret the eye diagram is to view it as a state diagram. At any sampling time instant the possible states that the received signal may be in can be seen from the eye diagram. This may be seen by comparing the eye diagrams of Figure 4.4 which depict the proper sampling instants with those of Figure 4.5 in which the sampling time has been offset by $T/8$. At the proper sampling instant the number of states is equal to the number of elements in the signal set, i.e. two for

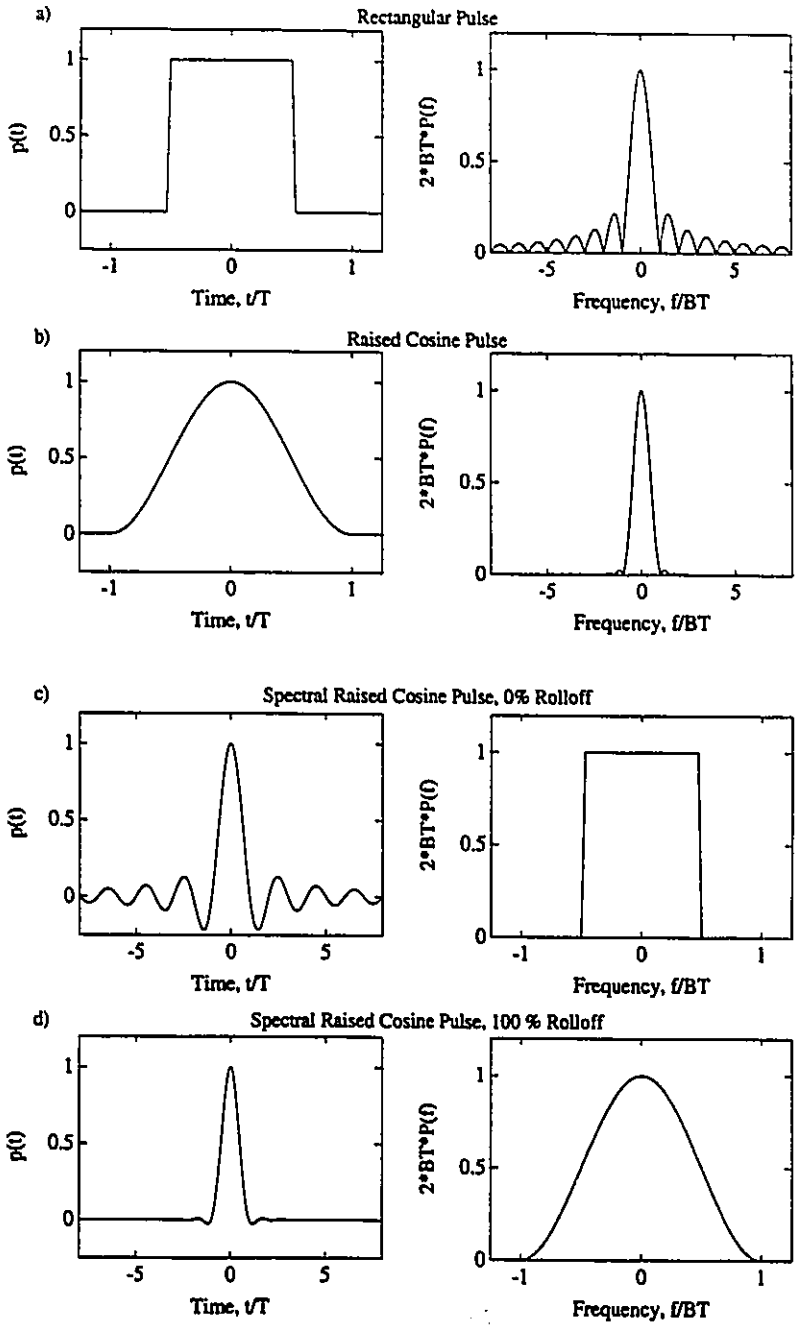

Figure 4.3: Pulse shaping filters, a) rectangular pulse, b) raised cosine pulse, c) spectral raised cosine pulse with no roll off, d) spectral raised cosine pulse with 100 % roll off.

binary signalling, and there is a one to one correspondence between a state and an element of the signal set. In Figure 4.5 with the sampling time offset by $T/8$ it can be seen that the location and the number of states is quite different from the previous figure. This is the result of intersymbol interference (ISI) and can be seen to be a function of the pulse shape. The amount of ISI is determined by the length of the data pulse shape and the magnitude of the tails of the pulse. The length of the pulse shape refers to the time it takes the tails of the pulse to become small enough that they can no longer be considered significant.

It can be seen from Figure 4.3 that the SRC pulse with no roll off has the largest and longest tails. This results in it having the most ISI when the sampling time is incorrect as can be seen in Figure 4.5. By increasing the roll off factor for the SRC filter the tails of the pulse decay much more quickly and this can be seen in Figure 4.5 to greatly reduce the amount of ISI. The amount of ISI can be further reduced by using the pulse shaping with an RC or RECT pulse shape.

To simulate the effects of time dispersive fading the following channel was used

$$h_c(t) = \delta(t) + 0.5\delta(t - \tau)$$

such that the received signal $r(t)$ is given as

$$r(t) = s(t) + 0.5\, s(t - \tau)$$

where τ was chosen to correspond to a delay of $T/2$ or T and

$$s(t) = s_k p(t - kT)$$

For the case of τ equal to T this channel corresponds to the channel $H_1(z)$ used in section 4.2.1. The effect of this channel and the effect of the choice of τ can be seen in Figure 4.6, $\tau = T/2$, and Figure 4.7, $\tau = T$. When τ is equal to T the effect of the channel is to double the number of distinct states at the desired sampling instant. Note that each state corresponds to only one element of the signal set and hence the transmitted symbol can be uniquely determined. The effect of the pulse shape is as discussed in the case where there was no fading. That is, the pulse shape determines the amount of ISI that occurs when the sampling instant is incorrect. If

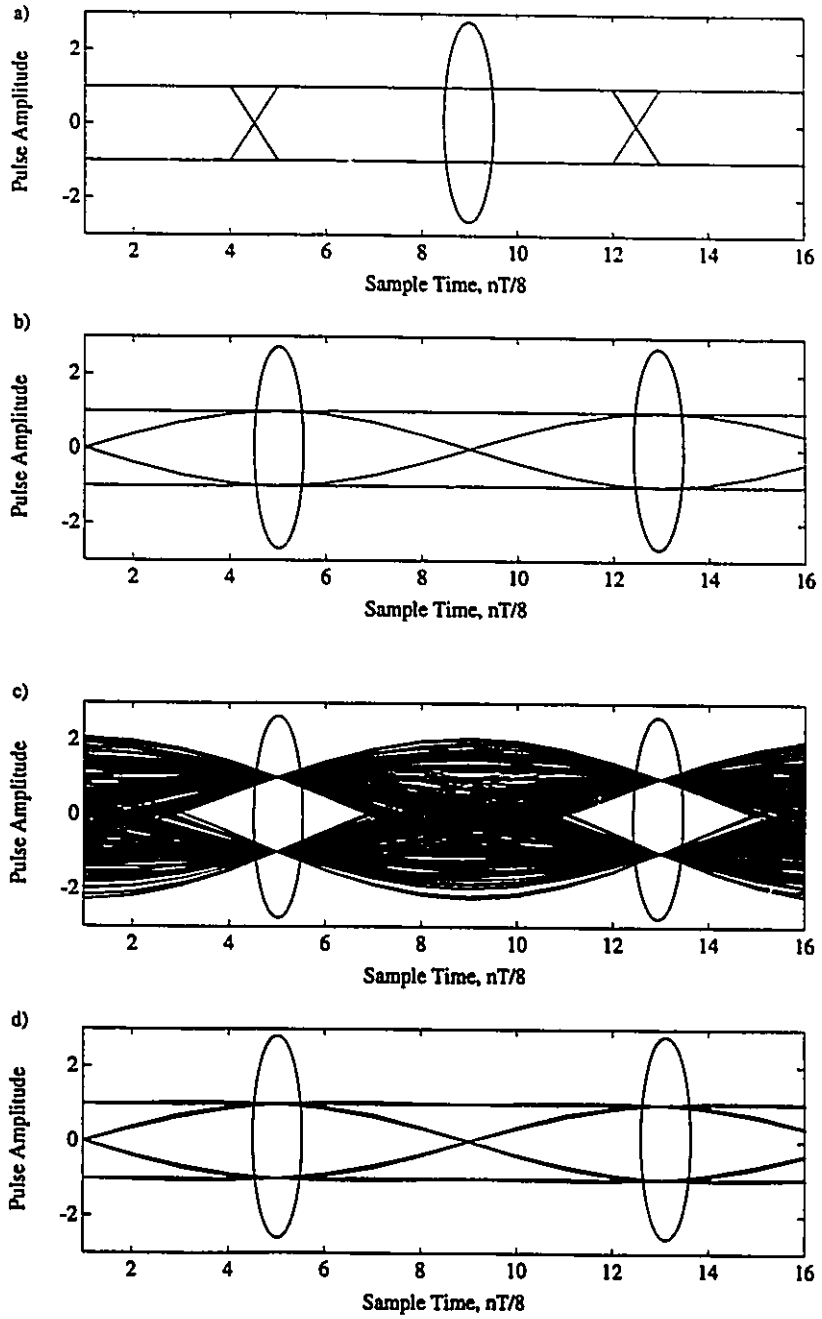

Figure 4.4: Eye diagrams for the fours pulses of the previous figure, a) rectangular pulse, b) raised cosine pulse, c) spectral raised cosine pulse with no roll off, d) spectral raised cosine pulse with 100 % roll off, depicting the correct sampling instants.

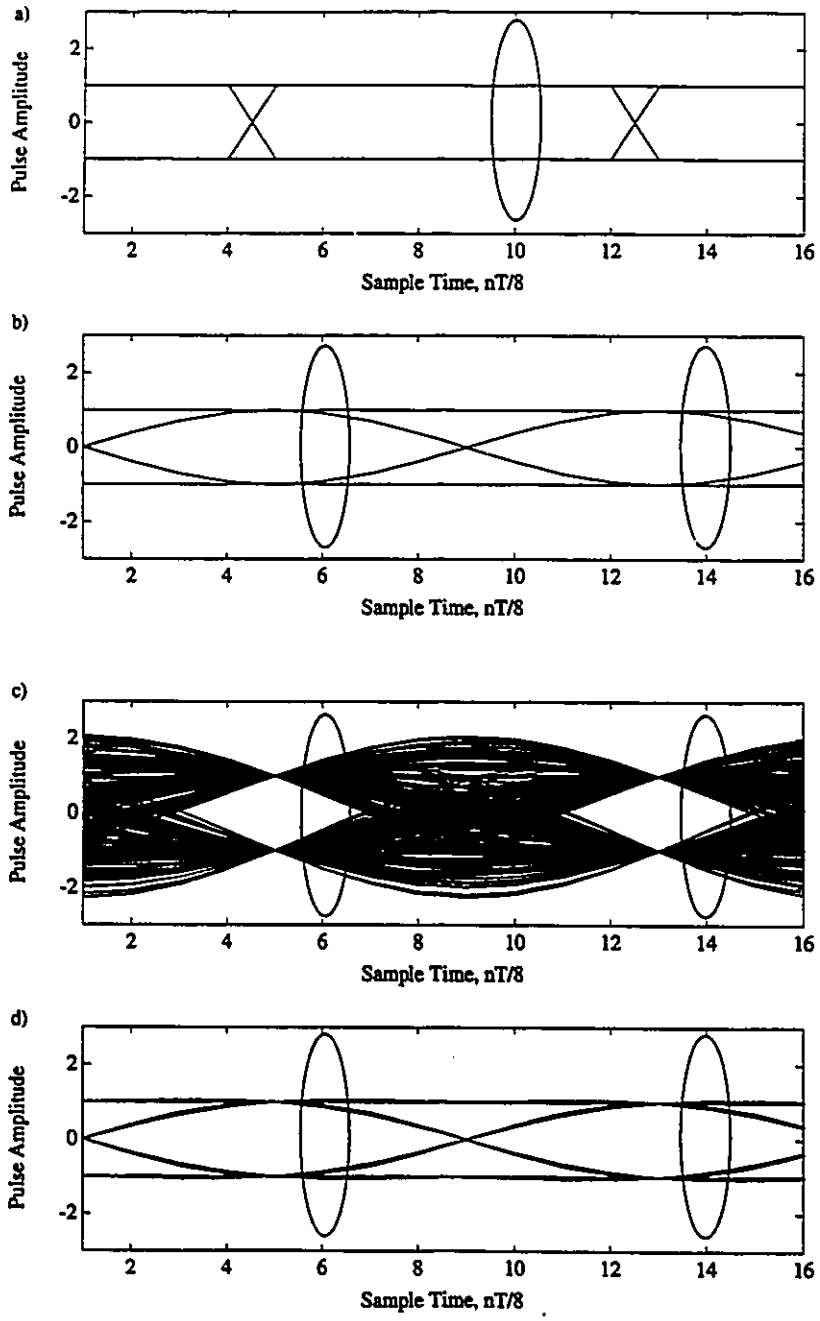

Figure 4.5: Eye diagrams for the fours pulses a) rectangular pulse, b) raised cosine pulse, c) spectral raised cosine pulse with no roll off, d) spectral raised cosine pulse with 100 % roll off with the sampling time offset by $T/8$.

one assumes perfect timing recovering then in this case ($\tau = T$) the pulse shape does not affect the signal at the sampling instants. However, for τ equal to $T/2$ it can be seen in Figure 4.6 that pulse shape has considerable effect on the signal even at the appropriate sampling instants, e.g. for an SRC pulse with no roll off a considerable amount of ISI results but with an RC or RECT pulse there is only a minimum amount of ISI.

In general, the signal paths resulting from a time dispersive channel are not delayed at integer multiples of the symbol time T. Hence the data pulse shape plays a key roll in determining the amount of ISI that results under these channel conditions. Clearly, a channel model based on delay paths who's relative delays are integer multiples of the symbol time is insufficient to accurately model an actual transmission system. Hence, a more realistic channel model is desired. As most of the work on neural network equalizers is based on such a model one must question the validity of the work as this work relies on a simplifying assumption, i.e. the relative path delays are integer multiples of the symbol time. What this work [28]-[38] has been useful in doing is illustrating that the optimum equalizer is indeed nonlinear.

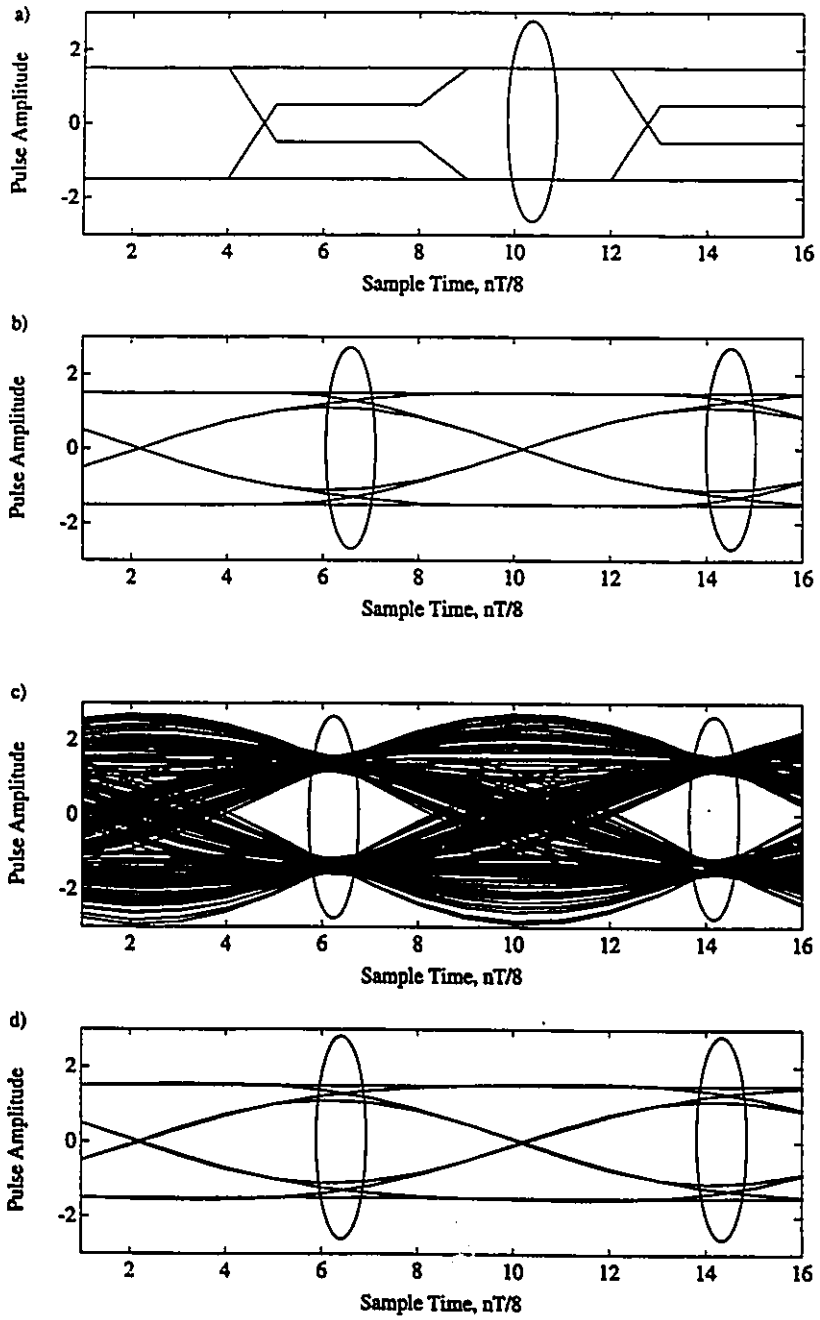

Figure 4.6: Eye diagrams for the fours pulses after being filtered by $h(t) = \delta(t) + 0.5\delta(t - T/2)$, a) rectangular pulse, b) raised cosine pulse, c) spectral raised cosine pulse with no roll off, d) spectral raised cosine pulse with 100 % roll off.

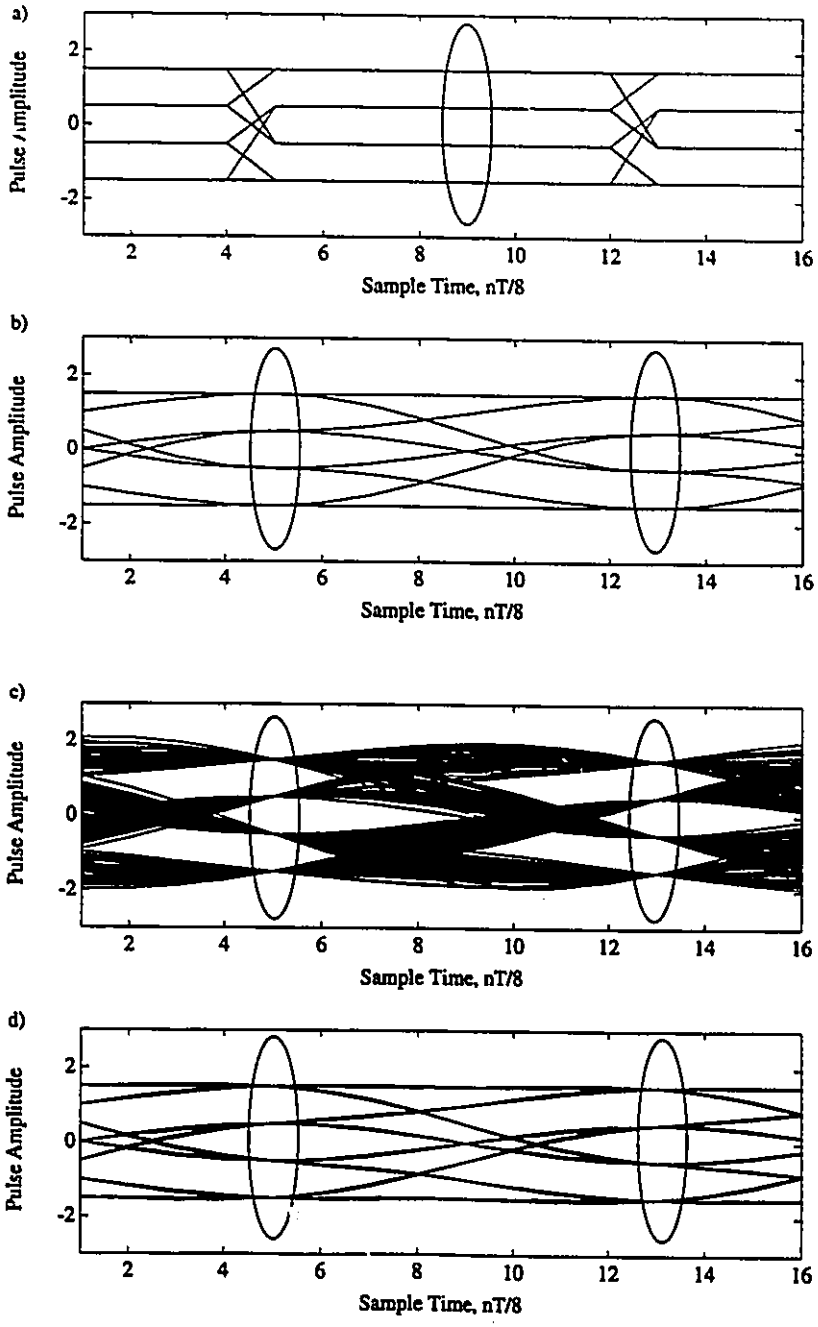

Figure 4.7: Eye diagrams for the fours pulses after being filtered by $h(t) = \delta(t) + 0.5\delta(t-T)$, a) rectangular pulse, b) raised cosine pulse, c) spectral raised cosine pulse with no roll off, d) spectral raised cosine pulse with 100 % roll off.

4.2.3 Bayesian Estimation

An alternative to detecting the signal s_{k-d} is to estimate it and then to make a decision on the symbol based upon the estimate. The estimator that minimizes the mean squared error between the estimate $\hat{s}_{k-d}$ and s_{k-d} is the Bayes estimator [152] given by

$$\hat{s}_{k-d} = \int_{\underline{\xi}} s\, p(s|\mathbf{x}_k)\, ds \tag{4.4}$$

where the range of integration $\underline{\xi}$ is over all possible values of the transmitted signal s_{k-d}. As s_{k-d} is an M'ary signal this reduces to the summation

$$\hat{s}_{k-d} = \sum_{j=1}^{M} s_j\, p(s_j|\mathbf{x}_k) \tag{4.5}$$

where $s_j \in \underline{\xi}$. For this function a neural network with a single output is used and the network is trained to minimize the expected value of the error power of e_k where $e_k = \hat{s}_{k-d} - s_{k-d}$.

This estimator is subject to the same problems as the Bayesian detector; the fact that we need to know how to calculate $p(s_j|\mathbf{x}_k)$ and that its computational complexity increases exponentially with the length of the channel's impulse response[1] and the length of the received signal vector $\mathbf{x}_k$. However, the Bayes estimator for the special case where the signal s_k is only corrupted by white Gaussian noise, $x_k = s_k + n_k$ yields an interesting result. As shown in the previous chapter the estimator function can be calculated as

$$\hat{s}_k = \begin{cases} tanh\left(x_k/\sigma_n^2\right) & s_k \in \pm 1 \\ \dfrac{\sum_{i=1}^{M} s_i\, e^{-\frac{(x(t)-s_i)^2}{2\sigma^2}}}{\sum_{j=1}^{M} e^{-\frac{(x(t)-s_j)^2}{2\sigma^2}}} & s_i, s_j \in \underline{\xi} \end{cases} \tag{4.6}$$

The *tanh* function may be recognized as a scaled and shifted version of the sigmoidal nonlinearity commonly used in the back propagation [134][133] algorithm.

[1]The length of the channel's impulse response refers to the time it takes the tails of the impulse response to become small enough that they can no longer be considered significant.

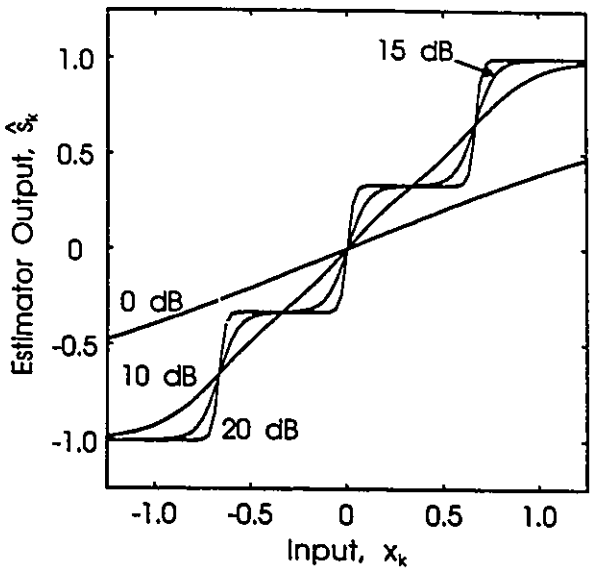

Figure 4.8: The Bayesian estimator function for 4-PAM with SNR's of 0, 10, 15 and 20 dB.

The Bayes estimator for $s_k \in \{\pm 1/3, \pm 1\}$ is shown in Figure 4.8. While the estimator for an M'ary signal is quite complex it may be closely approximated with

$$f(\alpha x) = \frac{1}{M-1} \sum_{i=1}^{M-1} tanh\left(\alpha (x - \theta_i)\right) \tag{4.7}$$

where α is a gain term chosen to reflect the noise level present in the system and the θ_i's are thresholds or biases chosen as the mid-points between adjacent signal points in the signal set $\underline{\xi}$, e.g. $\{\pm 2/3, 0\}$ for 4-PAM. From section 3.2 this function may be seen as the sum of $M-1$ neurons each with a *tanh* activation function with a gain of α, weights of unity and appropriately chosen bias levels. Evidently the back propagation neural network seems well suited to signal estimation. For low noise levels a minimum network would consist of the linear combination of the outputs of $M-1$ neurons in a single hidden layer. An alternative would be a single neuron using the multi-level nonlinearity given in 4.7. This would seem to make a case for neurons with multi-level activation functions.

4.3 Linear Equalization Theory

We have already seen how the optimal decision boundary can be nonlinear particularly when the decision delay d is minimal but what has not been established is the effect of the number of taps m and the length of the decision delay on equalizer performance. For this we look at linear filter theory. To illustrate the principles involved let us consider the the minimum and non-minimum phase channels considered previously. These two channels, $H_1(z)$ and $H_2(z)$ are given as:

$$H_1(z) = 1.0z^0 + 0.5z^{-1} \tag{4.8}$$

$$H_2(z) = 0.5z^0 + 1.0z^{-1} \tag{4.9}$$

If one ignores the constraints of causality and stability and is only concerned with eliminating ISI then the optimum ISI cancelling filters for $H_1(z)$ and $H_2(z)$ would be:

$$H_1^{-1}(z) = \frac{1}{1.0z^0 + 0.5z^{-1}} \tag{4.10}$$

$$H_2^{-1}(z) = \frac{1}{0.5z^0 + 1.0z^{-1}} \tag{4.11}$$

Note that these filters are infinite-impulse response (IIR) filters and that $H_2^{-1}(z)$ is unstable as its pole lies outside of the unit circle.

While the optimum ISI cancellation filter is generally an infinite-impulse response (IIR) filter, adaptive IIR filters are generally not used in practice due to a lack of guaranteed stability, lack of a quadratic performance surface and a minor performance gain over transversal equalizers[10]. Generally, adaptive transversal equalizer structures are used. The output of this filter is the weighted sum of the tap outputs of a tapped delay line, where the output may be written as

$$\hat{s}_{k-d} = \mathbf{w}_k^T \mathbf{x}_k \tag{4.12}$$

and $\mathbf{w}_k$ is the tap weight vector at time k defined as $\mathbf{w}_k^T = \{w_{0,k}, \ldots, w_{m-1,k}\}$. If the criterion for adjusting the tap weights is to eliminate ISI then this is known as a zero-forcing (zf) equalizer[21]. Essentially the transversal zero-forcing equalizer tries to approximate the inverse channel response using a finite number of taps. The problem

with a zero-forcing equalizer is that when there are deep notches in the frequency response of the channel, the zero-forcing equalizer can lead to excessive enhancement of the noise present on the received signals. The least mean-square (LMS) equalizer is more robust as it maximizes the signal to noise plus distortion ratio. That is, it minimizes the mean-squared value of the error signal, e_k, determined from the equalizer's output, i.e. $E\left[(s_{k-d} - \hat{s}_{k-d})^2\right]$ is minimized through a linear combination of the components of $\mathbf{x}_k$. This is simply a linearization of the Bayesian estimator.

For real valued data, the adaptation of the i^{th} tap weight at time k, $w_{i,k}$, is governed by the following equations:

$$\begin{aligned} w_{i,k+1} &= w_{i,k} - \eta\, e_k\, x_k \\ e_k &= \hat{s}_k - s_k \end{aligned}$$

where η is an adaptation parameter controlling the step size of the change in weights, e_k is the error signal at time k.

If the values of the channel impulse response at the sampling instants are known then the optimum m element tap weight vector $\mathbf{w_o}$ can be obtained by solving a set of m linear simultaneous equations [14] given by

$$\mathbf{R}\,\mathbf{w_o} = \mathbf{p} \tag{4.13}$$

where $\mathbf{R}$ is the auto-correlation matrix of the tap input vector $\mathbf{x}$, $E[\mathbf{x}\mathbf{x}^T]$, and $\mathbf{p}$ is the cross-correlation vector formed from the desired equalizer response d and the tap input vector $\mathbf{x}$, $E[d\mathbf{x}]$. Equation 4.13 is known as the *normal equation* [14]. The minimum obtainable mean-squared error is given as

$$J_{min} = \sigma_d^2 - \mathbf{p}^T\mathbf{w_o}. \tag{4.14}$$

where σ_d^2 is the variance of the desired response.

The performance of this filter is limited by three factors[15]: the presence of noise on the received signal, the necessity for the inclusion of a delay in the equalizer response, and the finite number of taps used to approximate the ideal response. The reason for the inclusion of a delay in the equalizer response is that the transmitter,

receiver and channel filters are causal systems. This means that the transmitted signal will be delayed in time as it passes through these filters. The use of a delay of d samples permits much lower values of minimum mean-squared error and causes the converged adaptive impulse response, when convolved with that of the overall channel response, to approximate an impulse with a delay of d [15]. A delayed inverse is also advantageous whenever the channel is non-minimum phase[15]. A reciprocal transfer function would have poles outside of the unit circle. In order for such an inverse to be stable, the impulse response would need to be left-handed in time (i.e. non-causal). A delayed non-causal response can be approximated, however, by a causal impulse response truncated in time. Widrow[15] indicates that when the inclusion of a delay is not critical, a good rule of thumb is to choose a delay equal to half the time length of the filter. Generally, the choice of the delay time is not critical as long as it corresponds to roughly the middle area of the filter. With the use of equations (4.13) and (4.14) we can evaluate the effect of the delay d and the number of taps on equalizer performance for $H_1(z)$ and $H_2(z)$. Figure 4.3a shows the effect of decision delay on equalizer mean-squared error (MSE) performance for a 16 tap delay line. For the minimum phase channel, the optimum delay corresponds to the beginning of the equalizer tapped delay line. For the maximum phase channel, the optimum delay corresponds to the end of the equalizer tapped delay line. Clearly, the optimum decision delay is a function of the phase of the channel, yet, from Figure 4.3a it can be seen that a delay corresponding to the middle of the tapped delay line results in good MSE performance. Figure 4.3b shows the effect of increasing the number of taps on equalizer MSE performance where the optimum decision delay was used for each equalizer. Here, equalizer performance levels off with the use of about 9 taps.

The work of Gibson and Cowan *et al* [28, 33, 34] is useful in showing why the optimum transversal structure is nonlinear. For the two channels $H_1(z)$ and $H_2(z)$ the plots of x_{k-1} versus x_k from section 4.2.1 (Figures 4.2a and 4.2b) show that the decision boundary is nonlinear and that the signal points are only linearly separable by a single line for the case of the minimum phase channel $H_1(z)$. However, it can be seen from the reduction of the mean-squared error in Figure 4.3a that the signals become linearly separable as the decision delay is allowed to increase for $H_2(z)$. This

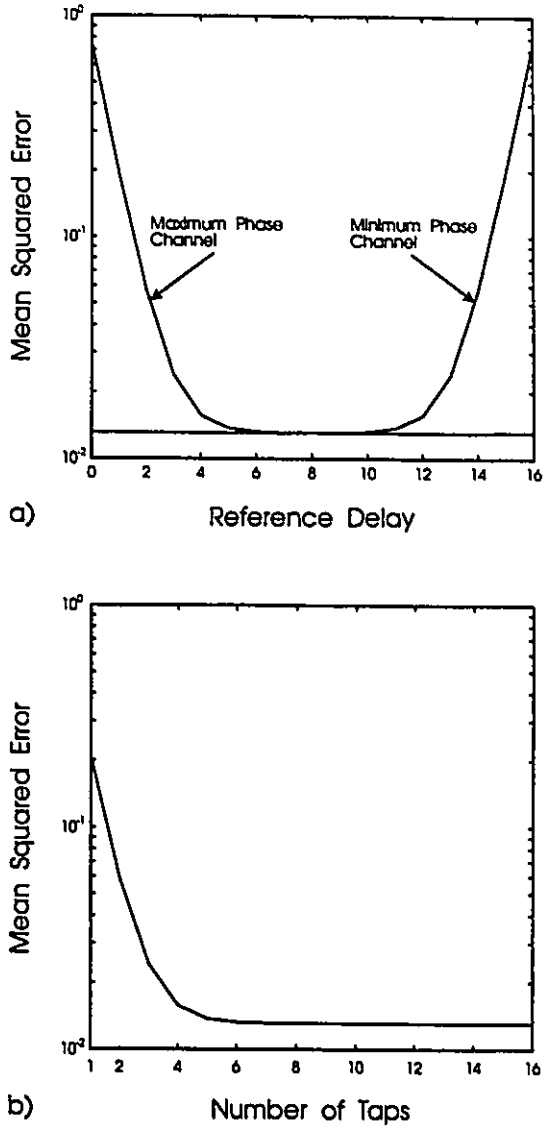

Figure 4.9: Part a: shows the MSE performance of a 16 tap linear equalizer as a function of the decision delay for the minimum and maximum phase channels $H_1(z)$ and $H_2(z)$. Part b shows the effect of increasing the number of taps of the equalizer on MSE performance. Note that the optimum decision delay was used for each equalizer.

is not to say that the optimal decision boundary is a linear one when the optimum decision delay is chosen rather that even though the signals may be linearly separable a nonlinear equalizer may still offer improved ISI and noise performance. What must be noted is that the optimum decision delay is a function of the phase of the channel.

The use of the Bayes signal detector forms the optimum signal detection scheme and allows for a minimum number of taps and minimum decision delay but at the cost of complexity. With the addition of a few more taps and a slightly increased decision delay the complexity of the system can be greatly decreased with only a slight degradation in system performance provided that the decision delay is appropriately chosen.

The performance of the LTE may be further improved through the use of decision feedback equalization (DFE) [16]. In this case the signals preceding r_{k-d}, $[r_{k-d-1}, \ldots, r_{m-1}]$, are replaced with the detected symbols of $[\hat{s}_{k-d-1}, \ldots, \hat{s}_{m-1}]$. Provided the detected values of $[\hat{s}_{k-d-1}, \ldots, \hat{s}_{m-1}]$ are correct the output noise is only a function of $[r_0, \ldots, r_{k-d}]$ rather than of $[r_0, \ldots, r_{k-m+1}]$. The effect of this on Bayesian detection/estimation is to reduce the number of states in $\mathbf{X}$ thereby reducing the complexity of the detector/estimator. A DFE generally performs better than a LTE, however, the DFE suffers from the problem of error propagation. If the output symbol of the DFE is in error then feeding it back into the equalizer will likely produce another symbol error. Hence the output of a DFE is subject to bursts of errors. The theoretical aspects of the DFE are beyond the scope of this chapter. The interested reader is referred to [16].

4.4 Neural Network Equalizer Structure

The neural network equalizer structure is identical to that shown in Figure (4.1) where the processor function represents the neural network. The outputs of the tapped delay line are the inputs to the neural network and the output of the network is a representation of the transmitted symbol $\hat{s}_k$. The objective of the network is to minimize the average mean squared error between the output of the equalizer, $\hat{s}_k$ and the desired signal, s_k. We are not interested in sequence estimation nor the application

of decision feedback equalization as in[33]. The type of setup where the tap outputs from the delay line are the inputs to the neural network and the output of the neural network is a representation of the signal s_k seems ideally suited to the application of feed-forward neural networks. This application is particularly appealing when one realizes that the structure of the LMS transversal equalizer is that of a neural network transversal equalizer consisting of a single neuron with a linear activation function.

A similar approach has been used by Gibson and Cowan *et al* [28]-[38] for binary signalling (2-PAM) however their channel model is based upon a multi-path fading model where the relative delays between paths are integer multiples of T, the symbol period. In effect their model does not take into account the effects of pulse shaping. A spectral raised cosine pulse has zeros every T seconds, hence in their model only adjacent pulses will cause ISI whereas in general if the path delays are non-integer multiples of T numerous pulses will interfere with each other due to the rather long (theoretically infinite but practically finite) tails of the raised cosine pulse. Furthermore, their approach is to utilize the neural network structure to minimize the decision delay discussed in section 4.3. In particular they fix the reference tap as the second tap of the delay line, a delay of one symbol. In the case of [31] they compare their nonlinear equalizer structure to various orders (number of taps) of linear transversal equalizers with a fixed symbol delay of 2 regardless of the equalizer order. It has already been shown in section 4.3 that the optimum delay is a function of the channel phase. Hence it is undesirable to choose a minimal delay and simply increase the order of the linear equalizer to improve performance unless the channel has a minimum phase response. If one has a priori knowledge of the channel then it is a simple task to determine a suitable symbol delay, however, in general no knowledge of the channel is known and in fact it may be randomly varying as in the case of the DMR channel. Here a more suitable strategy is to choose the center tap as the reference tap, representing a symbol delay of $\frac{N-1}{2}$ where N is odd and to choose a sufficient number of taps to ensure satisfactory performance. For our channel model (the Rummler channel model[41]), Amitay and Greenstein [140] indicate that with this delay strategy the performance of a 9 or 11 tap equalizer is quite close to that of an optimum infinite tap linear equalizer.

This neural network structure is strikingly similar to a sub-optimum nonlinear equalizer structure proposed by Ungerboeck[27]. Ungerboeck derives the optimum nonlinear equalizer based upon a maximum aposteriori probability approach. The optimum equalizer is shown to consist of a matched filter followed by a very complex nonlinear filter. Ungerboeck's type I sub-optimum structure[27] is shown in Figure 4.10. It consists of a matched filter followed by a tapped delay line. Each of the off center tap outputs is fed to a weighted squashing function much like a sigmoidal function. The output of the equalizer consists of the sum of the outputs of the squashing functions and the weighted value of the center tap. This structure is similar to a feed forward neural network with a linear output node which suggests the need to consider neural networks with linear output neurons. The difference between Ungerboeck's type-I structure and our approach with a neural network is that the tap outputs are fed to all of the input neurons in the neural network and we let the neural network sort out the appropriate weighting of the tap inputs through network training. The following work in this chapter extends the work of Gibson and

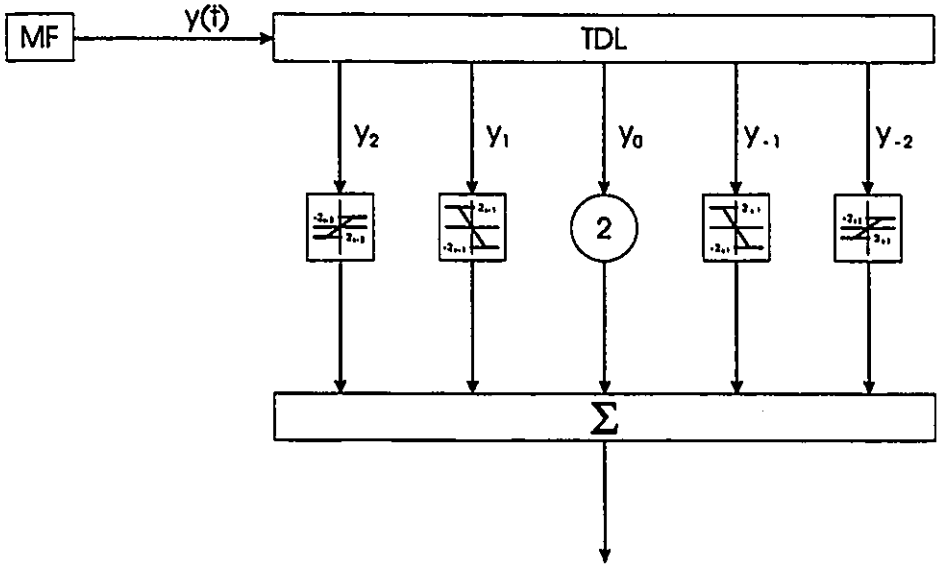

Figure 4.10: Ungerboeck's Type I nonlinear equalizer structure

Cowan *et al* [28]-[38] from 2-PAM to 4 and 16-QAM (from two levels in one dimension to multiple levels in two dimensions) and includes the effects of pulse shaping.

In particular a widely accepted (and perhaps more realistic) channel model is used i.e. the Rummler model[41] for the DMR channel. Further, we remove the restriction of trying to minimize the decision delay of the equalizer.

4.4.1 Neural Network Definition

Unless otherwise noted all networks considered here are based upon a 5-tap tapped delay line with the reference tap taken as the center tap. A neural network is labelled as $n_1 n_2 n_{out}[nl, l]$ where n_1 number of neurons in the 1 st hidden, n_2 number of neurons in the 2 nd hidden, n_{out} is the number of output neurons, and where the suffix nl denotes nonlinear output neurons while a suffix of l denotes linear output neurons.

The basic nonlinear activation function used in this work is a scaled and shifted form of the logistic or sigmoidal activation function with the inclusion of a gain term α

$$\begin{aligned} f(net) &= logistic(\alpha\, net) \\ &= \frac{1}{1+e^{-\alpha\, net}} \end{aligned}$$

where net is the activation of the neuron. If this is suitably scaled and shifted the activation function becomes

$$\begin{aligned} f(net) &= 2\, logistic(\alpha\, net) - 1 \\ &= \frac{2}{1+e^{-\alpha\, net}} - 1 \\ &= \tanh(\alpha\, net/2) \end{aligned}$$

This allows the use of outputs of ± 1 which are more appropriate for use with our transmission schemes.

An alternative to the above binary level function is the multi-level function considered in section 4.2.3 which is shown below.

$$f(\alpha\, net) = \frac{1}{N-1} \sum_{i=1}^{N} tanh\left(\alpha\left(net - \theta_i\right)\right) \tag{4.15}$$

Here, α is used to control the 'hardness' of the function. When α is small the function looks much like the $tanh$ function but as α increases the function becomes more like

a hard quantizer, i.e. a staircase like function. For both of these functions it may be desired to vary α to change the characteristics of the network, in which case, it may be desirable to backpropagate α. This is derived in Appendix B.

For use with two dimensional signalling systems the complex neurons discussed in section 2.11 are used, effectively the variables used for the real valued neuron take on their complex forms with the rules for complex arithmetic followed. The complex activation function is given as:

$$f(net) = f_R(net_R) + j\, f_I(net_I) \tag{4.16}$$

Here, the in-phase (I) channel corresponds to the real value of the complex output neuron and the quadrature (Q) channel corresponds to the imaginary values of the complex output neuron.

4.4.2 Input - Output Mappings

There are essentially two types of output mappings for the neural network equalizers; those based on estimating the transmitted signal, e.g. making a soft decision, and those based on detecting the transmitted signal, e.g. making a hard decision on the data symbol. For our work the estimation mapping uses a continuously but sometimes limited range variable output. This is achieved through the use of either a single linear or nonlinear output neuron. An alternative is to represent the output symbol by soft quantizing it and using the quantization bits to form the output mapping for the network. This is not considered in this work as one gets into the problem of just how many bits should one use. For networks that are based upon signal detection, the output neurons use the nonlinear functions describe in the previous section. These mappings are discussed further below.

4.4.3 Estimator Networks

Two types of estimation networks are considered here. In one, the output neuron uses a linear output function instead of the nonlinear sigmoidal or *tanh* functions. This is labelled as a $n_1 n_2 1 l$ network and is denoted as a linear neural network transversal

equalizer, LNNTE as opposed to a $n_1 n_2 n_{out} nl$ network which is denoted as a neural network transversal equalizer, NNTE. Note that a neural network equalizer with a single neuron with a linear activation function degenerates to a LMS equalizer structure. The other network uses a multi-level activation function (equation (4.7)) and is labelled as a $n_1 n_2 1 nl$ network. When this function is broken down into I and Q channels it is given as (section 4.2.3):

$$f(\alpha\, net) = tanh(\alpha\, net/2) \tag{4.17}$$

for 4-QAM and

$$f(\alpha\, net) = \frac{1}{4-1}\sum_{i=1}^{4} tanh\left(\alpha\left(net - \theta_i\right)\right) \tag{4.18}$$

for 16-QAM. In both of these functions α is used to control the 'hardness' of the function. From the Bayesian estimator theory discussed previously, α can be related to the amount of noise present with the received signal.

These types of networks may be desired if the equalizer is followed by a soft decision decoder, e.g. the use of the Viterbi algorithm for the decoding of convolutional codes or in some cases soft decision block code decoder.

A network with the *tanh* nonlinearity might be considered for multi-level signalling (e.g. 16-QAM) but this is not done here as it may be viewed as a combination of a linear and nonlinear activation function. The extremes of ± 1 are based upon the nonlinearity, the limiting part of the *tanh* activation function, while the values in between ± 1 fall along the linear like section of the *tanh* function.

4.4.4 Detector Networks

Detector networks make hard decisions on the transmitted signal and make use of nonlinear output neurons. For QAM the I channel corresponds to the real part of the output neurons and the Q channel corresponds to the imaginary part of the output neurons. In this way a QAM signal is broken down into two channels with each channel using signalling levels of $\{\pm 1\}$ for 4-QAM and $\{\pm 1, \pm 1/3\}$ for 16-QAM. A symbol is detected by using the detected symbols from the I and Q channels to determine the corresponding QAM symbol, e.g. if a 1 is detected in the I channel and

a $-1/3$ is detected in the Q channel then the corresponding detected QAM symbol is $\{1, -1/3\}$ or in complex notation $1 - j\,1/3$.

The first hard quantization functional mapping maps each of the signal levels of $\{\pm 1\}$ (4-QAM) or $\{\pm 1, \pm 1/3\}$ (16-QAM) on to a single complex output neuron, i.e. for a signal with m signal levels in the I and Q channels m complex output neurons are used, with each neuron assigned one of the m symbols. The detected symbol for the I channel is chosen as the symbol associated with the neuron that has the highest (or lowest) real valued output. The symbol for the Q channel is determined in an identical manner except that the imaginary outputs of the neurons are used.

For the case of binary level signalling in the I and Q channels this is similar to the work of Gibson *et al* [28, 31, 34] in that their network is based upon taking the sign of a cost function that is derived from Bayesian detection theory. That is they chose

$$\begin{aligned} m_1 \quad & if\, C_1 < C_2 \\ m_2 \quad & if\, C_2 < C_1 \end{aligned} \tag{4.19}$$

where C_1 is the cost associated with making an error by choosing symbol m_1 if symbol m_2 was transmitted and C_2 is the cost associated with choosing m_2 if m_1 was transmitted, e.g. a $n_1\, n_2\, 2nl$ network. For the I and Q channels for 16-QAM this becomes

$$\begin{aligned} m_1 \quad & if\, C_1 < C_2, C_1 < C_3, C_1 < C_4 \\ m_2 \quad & if\, C_2 < C_1, C_2 < C_3, C_2 < C_4 \\ m_3 \quad & if\, C_3 < C_1, C_3 < C_2, C_3 < C_4 \\ m_4 \quad & if\, C_4 < C_1, C_4 < C_2, C_4 < C_3 \end{aligned} \tag{4.20}$$

for each channel, e.g. a $n_1\, n_2\, 4nl$ network. These mappings are shown in Table 4.1

As mentioned in section 4.2.1 there has been some indication [153]-[156] that for this type of network the activation levels of each of the m output neurons correspond to the probabilities that the symbol represented by the neuron was transmitted which in the case of coded transmission it may be advantageous to feed the outputs of the neural network to a decoder rather than to make hard decisions on the signal. This would correspond to soft decision decoding and may allow the system to achieve some of the coding gain that may otherwise be lost by the use of hard decisions on the signal. This has not been investigated in this thesis.

Output symbol	Condition for Output Neurons (n_i)
-1	$n_1 > n_2$
$+1$	$n_2 > n_1$

Output symbol	Condition for Output Neurons (n_i)
-1	$n_1 > n_2$, $n_1 > n_3$, $n_1 > n_4$
$-1/3$	$n_2 > n_1$, $n_2 > n_3$, $n_2 > n_4$
$+1/3$	$n_4 > n_1$, $n_3 > n_2$, $n_3 > n_4$
$+1$	$n_4 > n_1$, $n_4 > n_2$, $n_4 > n_3$

Table 4.1: The mapping of m data symbols (from either the I or Q channels) onto the outputs of m neurons

The conditions for the binary case can be re-arranged so that we chose

$$\begin{aligned} m_1\,(-1) \quad & if\ sgn\,(C_1 - C_2) < 0, \quad i.e. -1 \\ m_2\,(+1) \quad & if\ sgn\,(C_1 - C_2) > 0, \quad i.e. +1 \end{aligned} \tag{4.21}$$

Thus, the output function of the network is equivalent to $sgn(C_1 - C_2)$, e.g. a $n_1\, n_2\, 1nl$ network, which for binary signalling (± 1) amounts to having the NNTE make hard decisions on the received data. It is not readily apparent as to how to apply this to more than two signal levels as it is impossible to combine the above conditions into one analytic expression $f(C_1, C_2, C_3, C_4)$ to choose the appropriate message. In training a network to produce $sgn\ (C_1\ -\ C_2)$ it should be realized that the network is actually being trained to minimize the mean squared error between the output of the network, the estimate $\hat{s}_{k-d}$, and s_{k-d}. This effectively means that the network is being trained as a Bayesian estimator rather than as a Bayesian detector. With this in mind it is possible to have a neural network rearrange equation (4.20) into a single analytical expression by using a network with a multi-level output neuron. In which case the network is acting as a Bayesian estimator.

An alternative mapping is to label each of the m signal values for the I and Q channels with a binary number consisting of $\log_2(m)$ bits where a bit has a value of ± 1. The advantage of this is it reduces the number of output nodes for the network, however, the activation levels will no longer correspond to probabilities. For binary level signalling $\{\pm 1\}$ this produces a network with a single output where the desired output is ± 1. Hence, for binary level signalling this network is equivalent to the

Bayesian estimator network, a $n_1\, n_2\, 1nl$ network.

Another approach is to use $m-1$ neurons and order their outputs. Such a mapping is shown in Table 4.2 using networks of the form $n_1\, n_2\, 1nl$ and $n_1\, n_2\, 3nl$ for 4 and 16 QAM. Note that if we consider the function $1/3\Sigma_{i=1}^{3} N_{i\,out}$ we obtain the mapping $\{\pm 1, \pm 1/3\} \Rightarrow \{\pm 1, \pm 1/3\}$. Thus a $n_1\, n_2\, 3nl$ network is very similar to a $n_1\, 3\, 1l$ or a $3\, 0\, 1l$ network with the output neuron acting as a linear combiner.

Transmitted Symbol	Output neuron n_1
-1	-1
$+1$	$+1$

Transmitted Symbol	Output neuron		
	n_1	n_2	n_3
-1	-1	-1	-1
$-1/3$	$+1$	-1	-1
$+1/3$	$+1$	$+1$	-1
$+1$	$+1$	$+1$	$+1$

Table 4.2: The mapping of m data symbols (from either the I or Q channels) onto the outputs of $m-1$ neurons

4.4.5 Minimum Number of Neurons

One of the 'arts' in neural networks is determining an appropriate network size. While a small network will be less complex and demonstrate faster learning times than a larger network, the larger network may offer improved performance through its ability to generalize better than a small network. The question is how large or small should a network be? The answer is generally determined by experimentation. This combined with the numerous possibilities for input/output mappings and trying to determine suitable adaptation parameters for each network can turn one's search for the ideal network size/configuration into a horrendous problem. There are algorithms that can assist in doing this but the approach taken here is based on intuition. While this approach does not have a strong theoretical background it is intuitively pleasing. The approach is based on considering the Bayesian estimator for an M-ary signal (in one dimension). This was done in in sections 3.2 and 4.2.3 where it was demonstrated that the minimum MLP network configuration for the estimation of an M-ary signal in white Gaussian noise consists of the linear combination of $M-1$ neurons arranged in a single hidden layer or alternatively, a single neuron with the multi-level activation function of (3.17). For M-QAM this corresponds to $\sqrt{M}-1$ complex neurons or a single complex neuron with a multi-level nonlinearity function. Hence, the minimum network size for 4-QAM is a single complex neuron, e.g. $001nl$ or $001l$, and for 16-QAM, three nonlinear complex neurons in one hidden layer or one complex neuron with a multi-level nonlinearity function, e.g. $301l$, $302nl$, $303nl$, $304nl$ or $301nl_{multi-level}$.

4.5 Simulation Results

4.5.1 SNR

This section investigates neural network equalizer performance as a function of SNR for 4 and 16-QAM. The equalizer consists of a 5 tap delay line with the reference tap corresponding to the center tap. Networks of the form $001nl$ and $951nl$ were

considered for 4-QAM and $303nl$, $304nl$, $953nl$ and $954nl$ were considered for 16-QAM. In addition, a $001nl$ with a multi-level nonlinearity was also used for 16-QAM. The smaller networks represent the minimum size networks for 4 and 16-QAM while the more complex $95n_{out}nl$ networks were used to determine if increased performance resulted from the use of larger networks.

The channel is a 20 dB notch placed either in the center of the channel, $f_0 = 0$ MHz, or half way to the band edge, $f_0 = 7.5$ MHz. The purpose of these two choices is to provide two channel cases, one where the channel response is symmetric about the center frequency, hence the I and Q channels are mutually independent, $f_0 = 0$ MHz, and the other where the channel response is asymmetric, hence the I and Q channels experience cross talk (they interfere with each other). The notch level and locations were chosen to so that equalization is required but that once equalized performance is dominated by the signal to noise ratio at low SNR's, e.g. 5 to 25 dB. This allows one to achieve error rates of around 10^{-1} - 10^{-3} where most of the errors are due to noise and not ISI which allows one to see if neural networks provide increased signal to noise performance in the presence of fading. Simulation of lower error rates is not desirable as this would greatly increase the required simulation time. To determine the error rate, the channel was simulated until 100 symbol errors were collected. Note that when discussing error rates, we are referring to the symbol error rate not the bit error rate. This makes the error rate independent of the mapping of bits onto the signal set. It is also easier to count symbol errors than bit errors.

The simulation involved two phases - one, the training of the equalizer network, the other, determining its performance. In training a network for a particular channel 4096 symbols were generated and passed through the channel simulator. This resulted in a training pattern of 4092 symbols for the equalizer. This is less than the original 4096 symbols that were generated due to the buffering effect of the tap delay line and the associated symbol delay of the equalizer. Each equalizer structure was allowed to adapt over the entire training sequence. Once the training sequence had been passed through, training was repeated using the same sequence until the equalizer converged. Convergence was said to have occurred when the MSE between successive training sequences was less than 10^{-4}. To evaluate the performance of the

equalizer, blocks of 4096 symbols were generated (each block different) until at least 100 symbol errors had been accumulated in the symbol error counter. At this point the simulation was stopped and the symbol error rate was calculated.

The results are presented in Tables 4.3-4.5. The overall result is that although numerous network configurations were tried none of them offered significantly improved performance over the LMS linear equalizer. While in some cases it appears that some of the networks converged faster than the LMS equalizer it should be remembered that there other adaptation algorithms for linear equalizers such as the recursive least squares (RLS) algorithm that exhibit significantly faster convergence rates, e.g. an order of magnitude or more, albeit with an increase in complexity. The advantage of the LMS algorithm over these other algorithms is its simplicity. The advantage of the use of neural networks over these linear techniques is its non-linear behaviour.

Statistically speaking, the results might be considered insignificant in that only one run was done for each trial but (with the exception of the backpropagation of α) the networks exhibit stable behaviour and all of them converged. From the tables one can see that there is a consistent behaviour and a pattern in the performance of the networks. That is the MSE decreases as the SNR is increased and the performance of the neural networks closely parallels that of the LMS equalizer. If anything, these results suggest that there may be a slight improvement in error performance with the use of neural networks, i.e. a small percentage increase. This seems to contradict the results of Gibson and Cowan *et al* [28]-[38]. However, their work is based on far simpler channel models with their equalizer structures based upon minimizing the decision delay of the equalizer. In contrast, we have chosen a more general approach in making the reference tap the center tap of the equalizer which corresponds to a decision delay of $\frac{N-1}{2}$ symbols (N odd) for an N tap equalizer. This likely allows the decision regions to be well separated by hyper-planes thereby limiting any improvement that might be possible with a nonlinear equalizer structure. Furthermore, they do not consider the effects of pulse shaping.

Symbol Error Rate							
f_0	SNR	lms $\mu = 0.025$	lms $\mu = 0.0025$	001nl $\eta = 0.05$	001nl $\eta = 0.005$	951nl $\eta = 0.05$	951nl $\eta = 0.005$
0	5	5.7e-1	5.4e-1	5.4e-1	5.4e-1	5.6e-1	5.4e-1
	10.	4.2e-1	4.1e-1	4.0e-1	4.0e-1	4.2e-1	4.2e-1
	15	2.3e-1	2.0e-1	2.1e-1	2.1e-1	2.2e-1	2.1e-1
	20	5.5e-2	5.0e-2	5.1e-2	4.9e-2	6.2e-2	5.1e-2
	25	1.8e-3	1.6e-3	2.0e-3	2.0e-3	3.2e-3	1.7e-3
7.5	5	5.0e-1	4.8e-1	5.0e-1	4.9e-1	4.9e-1	4.8e-1
	10	3.3e-1	3.1e-1	3.2e-1	3.2e-1	3.5e-1	3.1e-1
	15	1.5e-1	1.4e-1	1.3e-1	1.4e-1	1.5e-1	1.4e-1
	20	2.6e-2	2.3e-2	2.7e-2	2.7e-2	3.3e-2	2.6e-2
	25	9.5e-4	9.1e-4	8.1e-4	9.4e-4	1.2e-3	9.6e-4
Convergence (epochs)							
f_0	SNR	lms $\mu = 0.025$	lms $\mu = 0.0025$	001nl $\eta = 0.05$	001nl $\eta = 0.005$	951nl $\eta = 0.05$	951nl $\eta = 0.005$
0	5	3	9	3	4	14	7
	10	3	34	3	12	20	17
	15	3	25	12	55	18	12
	20	3	63	54	162	200	22
	25	3	89	59	153	13	37
7.5	5	3	19	3	5	81	7
	10	3	32	4	15	293	24
	15	3	35	15	65	203	22
	20	3	70	55	143	87	29
	25	3	65	48	125	12	31

Table 4.3: This table shows the symbol error rate and the number of epochs required for convergence as a function of SNR and notch location for 4-QAM for various equalization structures. The channel is a 20 dB notch located at either of 0.0 MHz or 7.5 MHz.

Symbol Error Rate							
f_0	SNR	lms $\mu = 0.025$	lms $\mu = 0.0025$	303nl $\eta = 0.005$	303nl $\eta = 0.05$	304nl $\eta = 0.005$	304nl $\eta = 0.05$
0	5	8.2e-1	8.3e-1	8.3e-1	8.2e-1	7.4e-1	7.5e-1
	10	6.7e-1	6.7e-1	6.5e-1	6.5e-1	6.0e-1	6.2e-1
	15	4.0e-1	4.0e-1	3.5e-1	3.6e-1	3.8e-1	3.8e-1
	20	8.1e-2	7.8e-2	7.4e-2	8.5e-2	1.8e-1	1.9e-1
	25	9.1e-4	9.6e-4	2.0e-3	2.8e-3	1.0e-1	1.2e-1
7.5	5	7.9e-1	7.8e-1	7.7e-1	7.6e-1	7.0e-1	7.1e-1
	10	6.1e-1	6.2e-1	5.6e-1	5.7e-1	5.4e-1	5.8e-1
	20	5.4e-2	5.8e-2	5.1e-2	5.1e-2	1.8e-1	1.6e-1
	25	1.3e-3	1.2e-3	1.7e-3	1.8e-3	1.5e-1	1.5e-1
Convergence (epochs)							
f_0	SNR	lms $\mu = 0.025$	lms $\mu = 0.0025$	303nl $\eta = 0.005$	303nl $\eta = 0.05$	304nl $\eta = 0.005$	304nl $\eta = 0.05$
0	5	3	4	7	8	89	25
	10	3	7	9	7	65	25
	15	3	9	18	7	145	94
	20	3	10	37	12	308	57
	25	3	10	29	11	135	84
7.5	5	3	4	7	8	58	19
	10	3	6	12	8	67	35
	15	3	8	20	9	153	78
	20	3	9	36	14	152	71
	25	3	9	29	11	151	39

Table 4.4: This table shows the symbol error rate and the number of epochs required for convergence as a function of SNR and notch location for 16-QAM for the LMS equalizer and the $303nl$ and $304nl$ networks. The channel is a 20 dB notch located at either of 0.0 MHz or 7.5 MHz.

Symbol Error Rate							
f_0	SNR	lms $\mu = 0.025$	lms $\mu = 0.0025$	953nl $\eta = 0.005$	953nl $\eta = 0.05$	954nl $\eta = 0.005$	954nl $\eta = 0.05$
0	5	8.2e-1	8.3e-1	8.3e-1	8.1e-1	7.4e-1	7.5e-1
	10	6.7e-1	6.7e-1	6.4e-1	6.4e-1	6.1e-1	6.3e-1
	15	4.0e-1	4.0e-1	3.6e-1	3.7e-1	3.7e-1	4.0e-1
	20	8.1e-2	7.8e-2	8.0e-2	1.2e-1	1.2e-1	1.4e-1
	25	9.1e-4	9.6e-4	9.4e-1[1]	1.1e-3	4.2e-2	4.9e-2
7.5	5	7.9e-1	7.8e-1	7.6e-1	7.6e-1	7.0e-1	7.2e-1
	10	6.1e-1	6.2e-1	5.6e-1	5.6e-1	5.4e-1	5.7e-1
	15	3.0e-1	3.1e-1	2.8e-1	2.9e-1	2.9e-1	3.2e-1
	20	5.4e-2	5.8e-2	9.4e-1[1]	6.9e-2	1.2e-1	9.2e-2
	25	1.3e-3	1.2e-3	1.7e-3	1.7e-3	1.1e-1	1.1e-1

Convergence (epochs)							
f_0	SNR	lms $\mu = 0.025$	lms $\mu = 0.0025$	953nl $\eta = 0.005$	953nl $\eta = 0.05$	954nl $\eta = 0.005$	954nl $\eta = 0.05$
0	5	3	4	8	6	72	34
	10	3	7	15	9	123	31
	15	3	9	19	7	142	141
	20	3	10	31	9	274	31
	25	3	10	2	8	254	46
7.5	5	3	4	14	7	99	14
	10	3	6	13	9	202	105
	15	3	8	21	10	237	61
	20	3	9	2	10	245	131
	25	3	9	21	9	210	99

Table 4.5: This table shows the symbol error rate and the number of epochs required for convergence as a function of SNR and notch location for 16-QAM for the LMS equalizer and the 9 5 3*nl* and 9 5 4*nl* networks. The channel is a 20 dB notch located at either of 0.0 MHz or 7.5 MHz. [1]Note that these networks did not converge properly.

It was also desirable to experiment with the effects of backpropagating the gain term α for both 4 and 16-QAM but for reasons of simplicity only the $0\,0\,1nl$ network was used. These results are shown in Tables 4.6-4.8. In Table 4.6 the results are for 4-QAM with a starting value for α of 1.0 and represent the symbol error rate and convergence time (in epochs) of the network. Three values for the adaptation parameter η_α for α were tried, 0.005, 0.05 and 0.5. In the latter case this resulted in network instability as an η_α of 0.5 is ten times greater than the value which was used for η, 0.05. Therefore, only the results for η_α's of 0.005 and 0.05 are given. The results for 16-QAM are shown in Tables 4.7 and 4.8. Here, due to the results for 4-QAM only two values of η_α were used, 0.005 and 0.05, but three different starting values for α were used, 1, 10 and 25. The output of the network is quite soft for an α of 1 and progresses to a hard quantization function when α is increased to 25. This causes the initial network to display three different characteristics owing to the three different initial values for α. Two tables are presented for 16-QAM as it was discovered that there was a convergence problem with these networks as can be seen in Table 4.7 where there are several cases where networks did not convergence. For these situations a 'not' is entered as the time taken to convergence. What happened was that with each change in α the weights of the network would change and with each of the weights α would change. Even though the network would move towards a solution the constant pull between the change in α and the change in weights would cause the MSE to continually change by an amount greater than 10^{-4} which was the convergence criterion. Hence the network was never seen to converge. The problem that one is faced with is how to alleviate this problem. It was decided to let α adapt for the first 20 epochs and then freeze it for the rest of the training period to allow the weights to adapt. The entire training period was chosen to be 50 epochs as this seemed to allow for adequate convergence time as seen from the results in Table 4.7.

The overall result is that none of these networks provided a significant improvement in SNR performance compared to the LMS equalizer. However, these experiments did bring to light some interesting characteristics and problems when one tries to backpropagate α. When the network is moving towards a solution the MSE gets smaller and α gets larger. The effect of increasing α causes two things to

happen. The first effect is to increase the effective learning rate of the network, that is the effective value of η is increased in proportion to the square of the ratio of the new α to the old α. This is undesirable when the network is nearing a solution as one wishes to decrease η not increase it in order to refine the solution. The other effect is that the network will tend to quantize the signal which may lower or reduce the MSE of the network even though the symbol error rate is not being reduced. This can lead to false convergence when the change in MSE is used as a convergence criteria or when a threshold MSE level is used to indicate convergence.

4.5.2 Outage Performance

To further evaluate the performance of the equalization schemes the outage performance[157] of various equalizers was evaluated. An outage event is said to occur when the bit error rate exceeds a specified threshold value for not more than ten consecutive seconds. If the BER exceeds this threshold for more than ten consecutive seconds the data link is said to be unavailable. The outage threshold chosen here is for a BER of 10^{-3}. Thus, the outage performance of a radio system is based upon those combinations of B and f_0 where the BER is 10^{-3}. To find the outage region for a radio the notch depth is varied for each value of f_0 until a bit error rate of 10^{-3} is achieved. For these simulations the notch depth was varied in steps of 1 dB. As the bit error rate is quite sensitive to the notch depth parameter near the outage region it is possible to count symbol errors rather than individual bit errors to determine at which notch depth the bit error rate will be 10^{-3} (to within 1 dB). If one draws a line through all the points (f_0, B) where the error rate becomes 10^{-3} then the outage region corresponds to the area above this line, assuming that as one moves upwards on the vertical axis the notch depth is increasing.

The outage performance for the networks listed in Table 4.9 is shown in Figures 4.11a, and b, for 4-QAM and Figure 4.11c for 16-QAM where the system signal to noise ratio is 63 dB, a typical value for the DMR channel. Note that here networks of the form 101 are used rather than 001. This was desired so that when the frequency response of the weights in the first hidden layer was evaluated the networks

Symbol Error Rate					
f_0	SNR	lms $\mu = 0.025$	lms $\mu = 0.0025$	001nl $\eta = 0.05$ $\eta_\alpha = 0.005$	001nl $\eta = 0.05$ $\eta_\alpha = 0.05$
0	5	5.7e-1	5.4e-1	5.4e-1	5.4e-1
	10	4.2e-1	4.1e-1	4.0e-1	4.0e-1
	15	2.3e-1	2.0e-1	2.1e-1	2.1e-1
	20	5.5e-2	5.0e-2	4.9e-2	4.9e-2
	25	1.8e-3	1.6e-3	2.4e-3	3.1e-3
7.5	5	5.0e-1	4.8e-1	4.7e-1	4.7e-1
	10	3.3e-1	3.1e-1	3.2e-1	3.2e-1
	15	1.5e-1	1.4e-1	1.5e-1	1.5e-1
	20	2.6e-2	2.3e-2	2.6e-2	2.6e-2
	25	9.5e-4	9.1e-4	8.2e-4	1.6e-3
Convergence (epochs)					
f_0	SNR	lms $\mu = 0.025$	lms $\mu = 0.0025$	001nl $\eta = 0.05$ $\eta_\alpha = 0.005$	001nl $\eta = 0.005$ $\eta_\alpha = 0.05$
0	5	3	9	20	5
	10	3	34	28	15
	15	3	25	30	27
	20	3	63	24	36
	25	3	89	20	10
7.5	5	3	19	25	6
	10	3	32	31	17
	15	3	35	36	31
	20	3	70	14	33
	25	3	65	20	12

Table 4.6: This table shows the symbol error rate and the number of epochs required for convergence as a function of SNR and notch location for 4-QAM for the LMS equalizer and the 001*nl* network. For the neural network both the weights and gain terms of the neurons were allowed to adapt with adaptation parameters of $\eta = 0.05$ for the weights and either an η_α of 0.005 and 0.05 for the gain terms. The channel is a 20 dB notch located at either of 0.0 MHz or 7.5 MHz.

Symbol Error Rate									
f_0	SNR	lms $\mu = 0.025$	lms $\mu = 0.0025$	001nl $\eta = 0.005$	001am1 $\eta = 0.05$	001am1 $\eta = 0.005$	001am10 $\eta = 0.05$	001am25 $\eta = 0.005$	001am25 $\eta = 0.05$
0	5	8.2e-1	8.3e-1	8.2e-1	8.1e-1	8.1e-1	8.1e-1	8.0e-1	8.1e-1
	10	6.7e-1	6.7e-1	6.4e-1	6.4e-1	6.3e-1	6.4e-1	6.4e-1	6.4e-1
	15	4.0e-1	4.0e-1	3.5e-1	3.6e-1	3.7e-1	3.7e-1	3.7e-1	3.7e-1
	20	8.1e-2	7.8e-2	9.0e-2	8.3e-2	7.8e-2	8.3e-2	7.8e-2	7.8e-2
	25	9.1e-4	9.6e-4	2.5e-3	1.3e-3	1.2e-3	1.3e-3	2.1e-3	2.1e-3
7.5	5	7.9e-1	7.8e-1	7.5e-1	7.5e-1	7.5e-1	7.5e-1	7.8e-1	7.6e-1
	10	6.1e-1	6.2e-1	5.6e-1	5.6e-1	5.6e-1	5.6e-1	5.8e-1	5.5e-1
	15	3.0e-1	3.1e-1	2.8e-1	2.7e-1	3.0e-1	2.7e-1	2.8e-1	2.7e-1
	20	5.4e-2	5.8e-2	6.6e-2	5.1e-2	4.5e-2	5.1e-2	5.6e-2	5.6e-2
	25	1.3e-3	1.2e-3	2.6e-3	1.2e-3	1.1e-3	1.2e-3	3.4e-3	3.4e-3
Convergence (epochs)									
f_0	SNR	lms $\mu = 0.025$	lms $\mu = 0.0025$	001nl $\eta = 0.005$	001am1 $\eta = 0.05$	001am1 $\eta = 0.005$	001am10 $\eta = 0.05$	001am25 $\eta = 0.005$	001am25 $\eta = 0.05$
0	5	3	4	18	13	not	23	48	33
	10	3	7	4	12	not	25	47	35
	15	3	9	4	14	3	7	22	24
	20	3	10	4	17	13	10	3	3
	25	3	10	not	20	4	12	3	3
7.5	5	3	4	16	14	not	23	not	32
	10	3	6	4	15	53	30	18	43
	15	3	8	4	14	3	3	26	26
	20	3	9	4	17	20	10	3	3
	25	3	9	not	21	3	13	3	3

Table 4.7: This table shows the symbol error rate and the number of epochs required for convergence as a function of SNR and notch location for 16-QAM for the LMS equalizer and the 001*nl* network with a multi-level nonlinearity. For the neural network both the weights and gain terms of the neurons were allowed to adapt with adaptation parameters of $\eta = 0.05$ for the weights and either an η_α of 0.005 and 0.05 for the gain terms. The initial starting values of α for the multi-level nonlinearity were 1 (denoted as the 001am1 network), 10 (denoted as the 001am10 network), and 25 (denoted as the 001am25 network). The channel is a 20 dB notch located at either of 0.0 MHz or 7.5 MHz.

Symbol Error Rate									
f_0	SNR	lms $\mu = 0.025$	lms $\mu = 0.0025$	001nl $\eta = 0.005$	001am1 $\eta = 0.05$	001am1 $\eta = 0.005$	001am10 $\eta = 0.05$	001am25 $\eta = 0.005$	001am25 $\eta = 0.05$
0	5	8.2e-1	8.3e-1	8.2e-1	8.2e-1	8.1e-1	8.1e-1	8.2e-1	8.1e-1
	10	6.7e-1	6.7e-1	6.4e-1	6.3e-1	6.4e-1	6.3e-1	6.5e-1	6.4e-1
	15	4.0e-1	4.0e-1	3.5e-1	3.6e-1	3.7e-1	3.6e-1	3.8e-1	3.7e-1
	20	8.1e-2	7.8e-2	8.9e-2	8.2e-2	7.8e-2	8.2e-2	7.9e-2	8.2e-2
	25	9.1e-4	9.6e-4	5.1e-3	1.3e-3	1.1e-3	1.6e-3	2.2e-3	2.2e-3
7.5	5	7.9e-1	7.8e-1	7.7e-1	7.6e-1	7.6e-1	7.6e-1	7.9e-1	7.6e-1
	10	6.1e-1	6.2e-1	5.5e-1	5.5e-1	5.6e-1	5.5e-1	5.7e-1	5.6e-1
	15	3.0e-1	3.1e-1	2.6e-1	2.7e-1	2.8e-1	2.8e-1	2.8e-1	2.7e-1
	20	5.4e-2	5.8e-2	7.4e-2	5.3e-2	5.0e-2	7.5e-2	5.6e-2	5.6e-2
	25	1.3e-3	1.2e-3	5.3e-3	1.2e-3	1.3e-3	1.3e-3	3.4e-3	3.5e-3

Table 4.8: This table shows the symbol error rate as a function of SNR and notch location for 16-QAM for the LMS equalizer and the 0 0 1nl network with a multi-level nonlinearity. For the neural network both the weights and gain terms of the neurons were allowed to adapt with adaptation parameters of $\eta = 0.05$ for the weights and either an η_α of 0.005 and 0.05 for the gain terms. However, α was allowed to adapted for only the first 20 epochs and then it was frozen while the network weights were allowed to adapt for an additional 30 epochs. After which time training was stopped and the network's performance was evaluated. The initial starting values of α for the multi-level nonlinearity were 1 (denoted as the 001am1 network), 10 (denoted as the 001am10 network), and 25 (denoted as the 001am25 network). The channel is a 20 dB notch located at either of 0.0 MHz or 7.5 MHz.

4-QAM		16-QAM	
Network	Training Epochs	Network	Training Epochs
Opt. LMS		Opt. LMS	
LMS	200	LMS	200
1011	200	3011	200
101nl	200	303nl[1]	200
102nl	200	404nl	200
951nl	500	953nl[1]	500
1591nl[1]	1000	954nl[1]	500
Opt 9 Tap LMS		Opt 9 Tap LMS	

Training pattern size 4 k symbols
Testing pattern size 16 k symbols

Table 4.9: This is a listing of the neural networks which were evaluated for outage performance and the training information for the networks. All of these networks were trained with a learning constant, η, of 0.05. [1]These networks were simulated but their results are not shown here as they are not significantly different than for the simpler networks

for 4-QAM would have the same number of hidden layers as the networks for 16-QAM. The results for 4-QAM yielded some surprising results. Although the 1011 LNNTE performed similarly to the LTE, the 101nl network shows substantial outage improvement over the LTE and the 1011 LNNTE. The only difference between the 1011 and the 101nl networks is in the output activation function, one being linear, the other nonlinear (tanh). This might suggest that it is better to use a nonlinear output activation function, however, a 9511 LNNTE was evaluated and yielded comparable performance to the 951nl NNTE. Although the 951nl and 9511 networks offer greatly improved outage performance, the cost for this is increased complexity and training times.

The performance of a 9 tap LTE is also shown and it can be readily seen that with the addition of four more taps the performance of the equalizer is greatly improved though still less than that achieved with the 951 NNTE's. However, this improvement is accomplished with far less complexity and training time than that of the NNTE's. Amitay and Greenstein [140] indicate that the performance of a 9 or

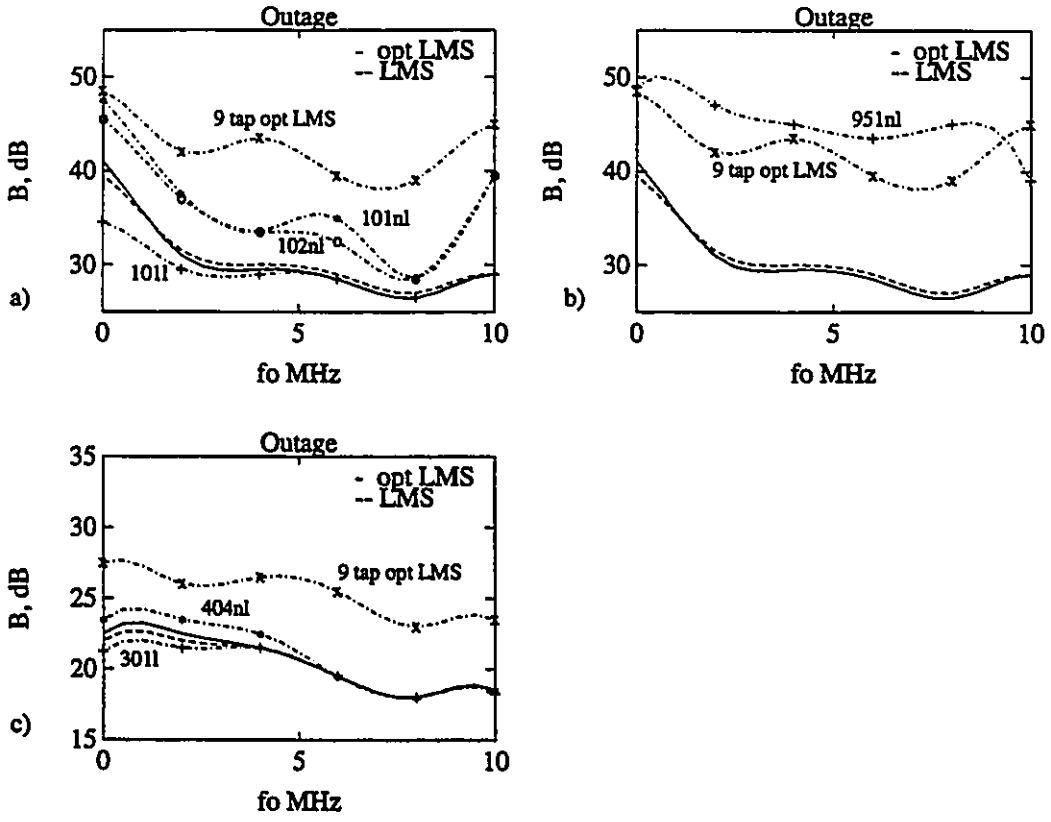

Figure 4.11: Outage performance for 4-QAM (top two graphs) and 16-QAM (bottom left graph). Each curve shows the notch depth, B (dB), as a function of notch frequency, f_0 (MHz), where the bit error rate for the receiver is 10^{-3}. (Notch frequency is measured relative to the center of the channel, 0 MHz). The outage region for a curve is represented by the area above the curve. Performance of the optimum LMS equalizer "opt lms" (5 taps) is shown by the solid curve while the performance of the simulated LMS equalizer "lms" (5 taps) is shown by the dashed curve. These results are accurate to ± 1 dB.

11 tap equalizer is quite close to that of an equalizer with an infinite number of taps. Thus one would expect that there would not be much more improvement in outage performance through the use of more than 9 taps for an LTE. Hence, any improvement through the use of more taps is likely to be less than that achieved with the 951(n)l networks. The results for the use of NNTE's with higher level signal constellations tell a different story. Figure 4.11c shows the results for 16-QAM. It is clearly indicated here that there is very little improvement in outage performance for the NNTE's. Larger networks were tried, i.e. 951l, 953nl, 954nl, and although their performance is not displayed they were found to yield no improvement over the networks depicted here. Nevertheless, note the performance improvement obtained with the 9 tap LTE over the 5 tap LTE for 16-QAM, approximately a 5 dB improvement in notch depth. This pretty much indicates that the better way to go is to increase the number of taps rather than to consider NNTE's for use with higher level signal sets. Ungerboeck offers an interesting comment on the application of nonlinear equalizers to higher level signal sets in that he says

> The investigation in this paper was restricted to binary antipodal signals. One might think of a generalization to multilevel signals. However, it is felt that then nonlinear methods will lose much of their attraction, not only because of the complexity of such schemes but also because the inter-symbol interference will more and more approach a Gaussian distribution as the number of levels is increased. In the latter case a linear equalizer will be optimum.

This may help to explain the performances differences between 4 and 16-QAM.

4.5.3 Frequency Response

When a multi-layered perceptron is used in conjunction with the tapped delay line it may be seen that the linear parts of the neurons in the first hidden layer appear as a system of linear transversal equalizers with the output of each equalizer being fed to the activation function used in each neuron. The system thus appears to be a linear transversal filter followed by a Bayesian estimator for s_{k-d} in white Gaussian noise.

The purpose of the weights in the first hidden layer is to approximate the inverse channel filter to remove the effects of the channel from the transmitted signal. The rest of the network then may be viewed as the Bayesian estimator or detector for the received signal, s_{k-d}, depending upon how the output of the network is configured. By looking at the weights of the first hidden layer as weights of a FIR filter it is possible to interpret these weights in the frequency domain in exactly the same ways as is possible to do so for the weights of a linear transversal equalizer.

The problem with this interpretation is that the weights of neurons in the first hidden layer will colour the noise on the received signal through their filtering action. Thus the noise on the signal at the input to the neural activation functions will be coloured (correlated) noise rather than white. Hence the optimum Bayesian estimator function is no longer the $tanh(s_{k-d}/\sigma_n^2)$ function, however, for moderate to high signal to noise ratios it is a good approximation. Such is usually the case in practice where usable error rates demand moderate to high signal to noise ratios even when there is no frequency selective fading.

To examine why the NNTE performed so much better for 4-QAM the frequency response of the linear part of the first hidden layer for various networks was plotted. These are shown in Figures 4.12 to 4.19. Figures 4.12, 4.13 and 4.14 show the frequency response of various networks to a 20 dB notch at 0.0 and 7.5 MHz. Obviously the frequency response of the linear part of the first hidden layer compares quite favourably with that of the LTE's. Any magnitude discrepancies may be attributed to the use of nonlinearities and the multiple layers of neurons used in the networks. In trying to determine why the 101nl and 951nl networks exhibited the dramatic improvement in outage performance, their frequency response (first hidden layer) was evaluated for a 40 dB notch at six channel locations equally spaced from $f_0 = 0$ MHz to $f_0 = 10$ MHz for comparison to the response of the LTE, Figures 4.16-4.19. Most of these channel conditions corresponds to an outage condition for the LTE. Notice the dimple in the LTE response at f_0 for notch locations of 0.0, 2.0 and 4.0 MHz which sharply contrasts with the NNTE's response and the response of the channel. The channel's frequency response, which is not shown here, is a smooth

continuous function and has no dimple in it, so why should the response of the equalizer? Obviously the NNTE provides for a better frequency interpolation for the 5 tap delay line than the LMS algorithm for the LTE. This is more clearly seen in Figure 4.16 which depicts the responses of the 101nl network. The 404nl[2] and 3011 networks for 16-QAM are shown in Figure 4.19. Observe that with the use of more than two signal levels per I and Q channel the NNTE is unable to provide a better interpolation of the frequency response than the LTE. This does not fully explain why the NNTE's do not offer any improvement over LTE's for 16-QAM as the outage area for 16-QAM is much lower than the 40 dB notches considered here. Rather the notch depth ranges from 18 to 23 dB where the frequency response of the LTE does not display the 'dimples' associated with the deeper notch depths (40 dB).

[2]A 404 network was used in this case as it was believed that the number of hidden units should be equal to or greater than the number of output units

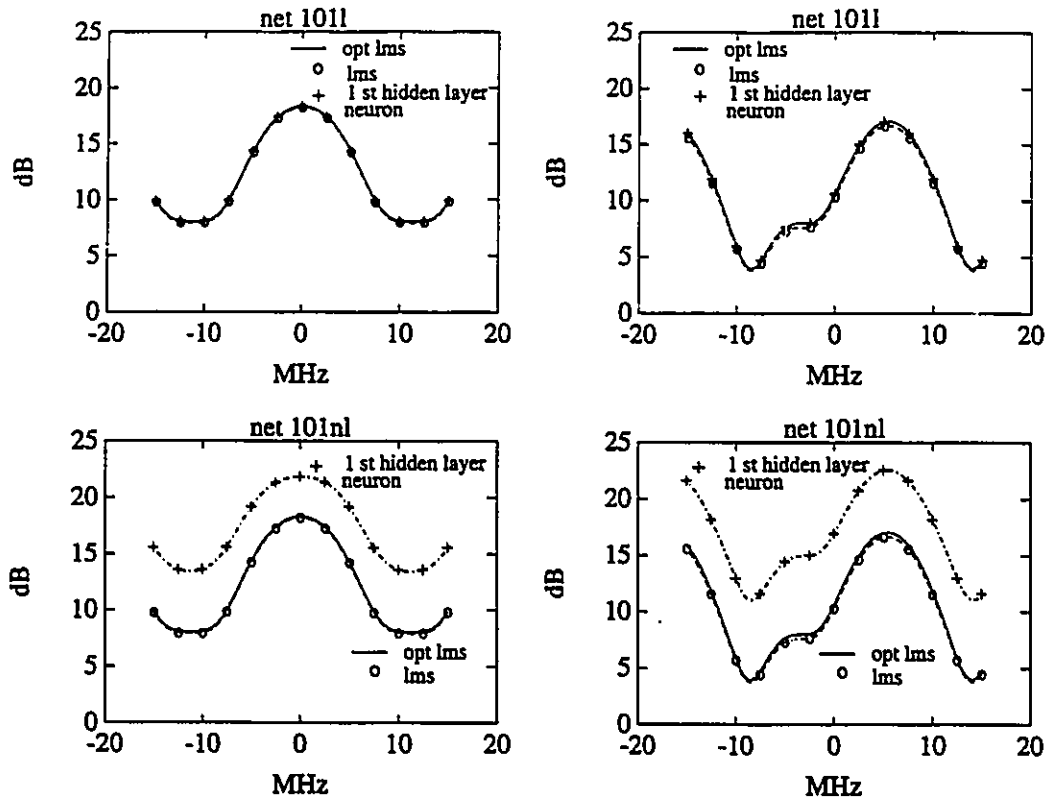

Figure 4.12: Frequency responses for the 4-QAM $101l$ and $101nl$ networks for a 20 dB notch at 0.0 MHz (left) and 7.5 MHz (right).

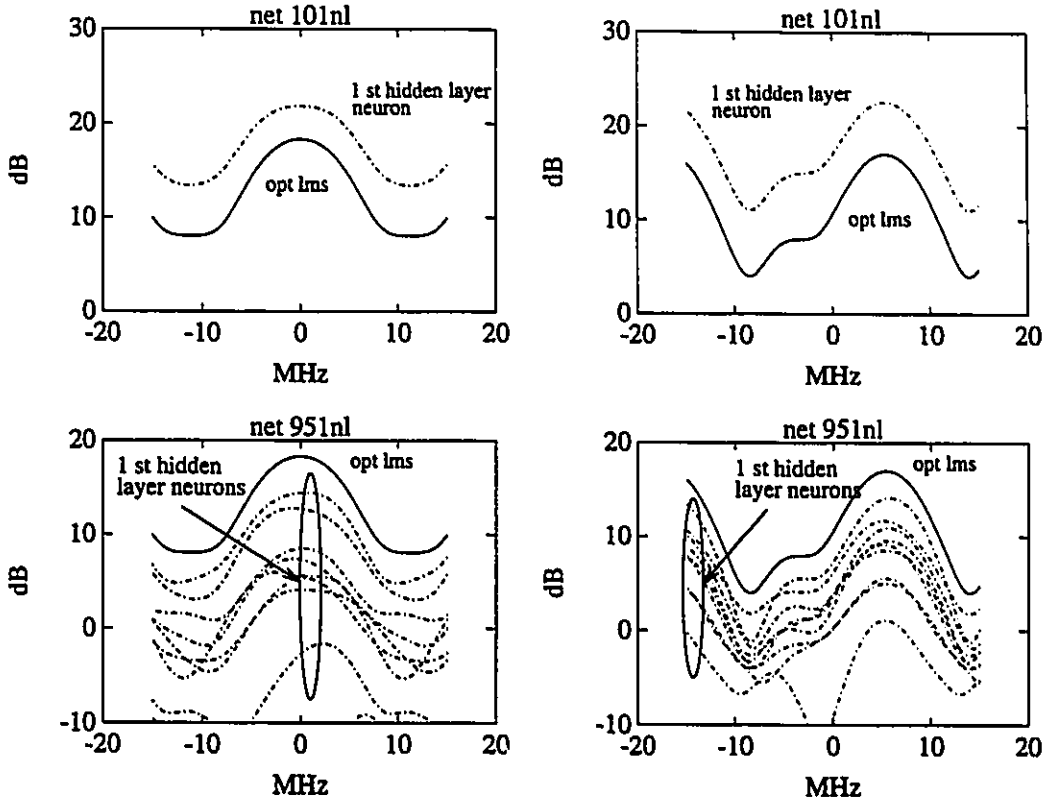

Figure 4.13: Frequency responses for the 4-QAM $101nl$ and $951nl$ networks for a 20 dB notch at 0.0 MHz (left) and 7.5 MHz (right).

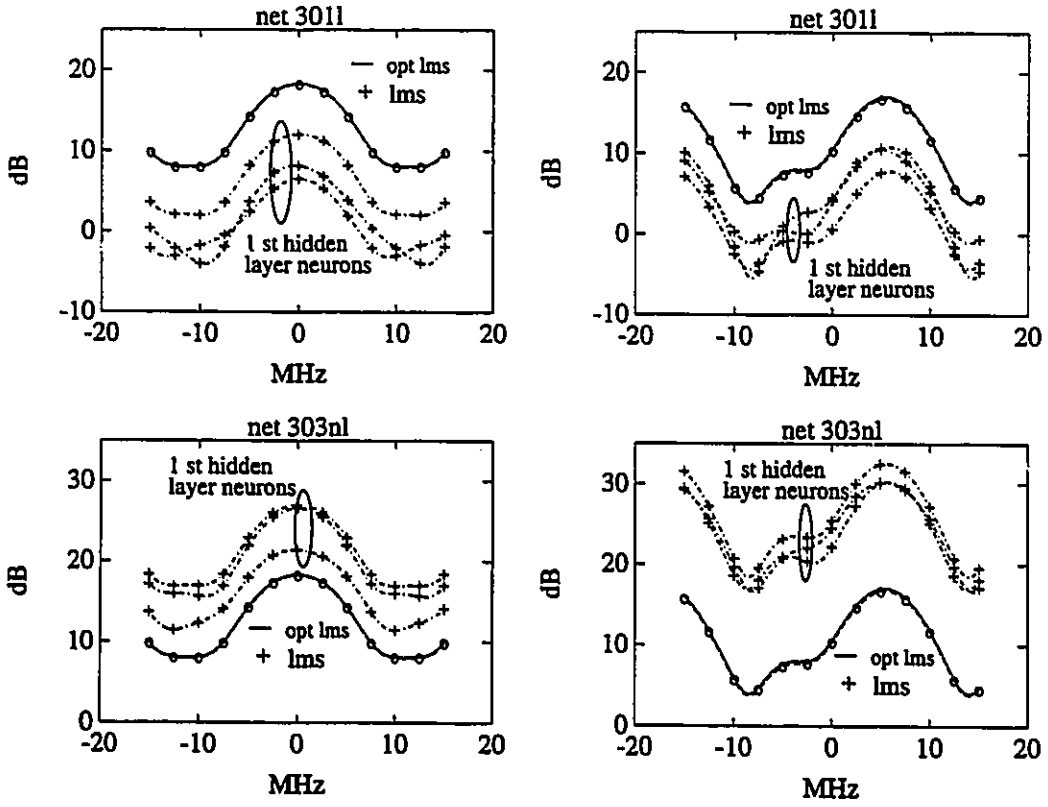

Figure 4.14: Frequency responses for the 16-QAM $301l$ and $303nl$ networks for a 20 dB notch at 0.0 MHz (left) and 7.5 MHz (right).

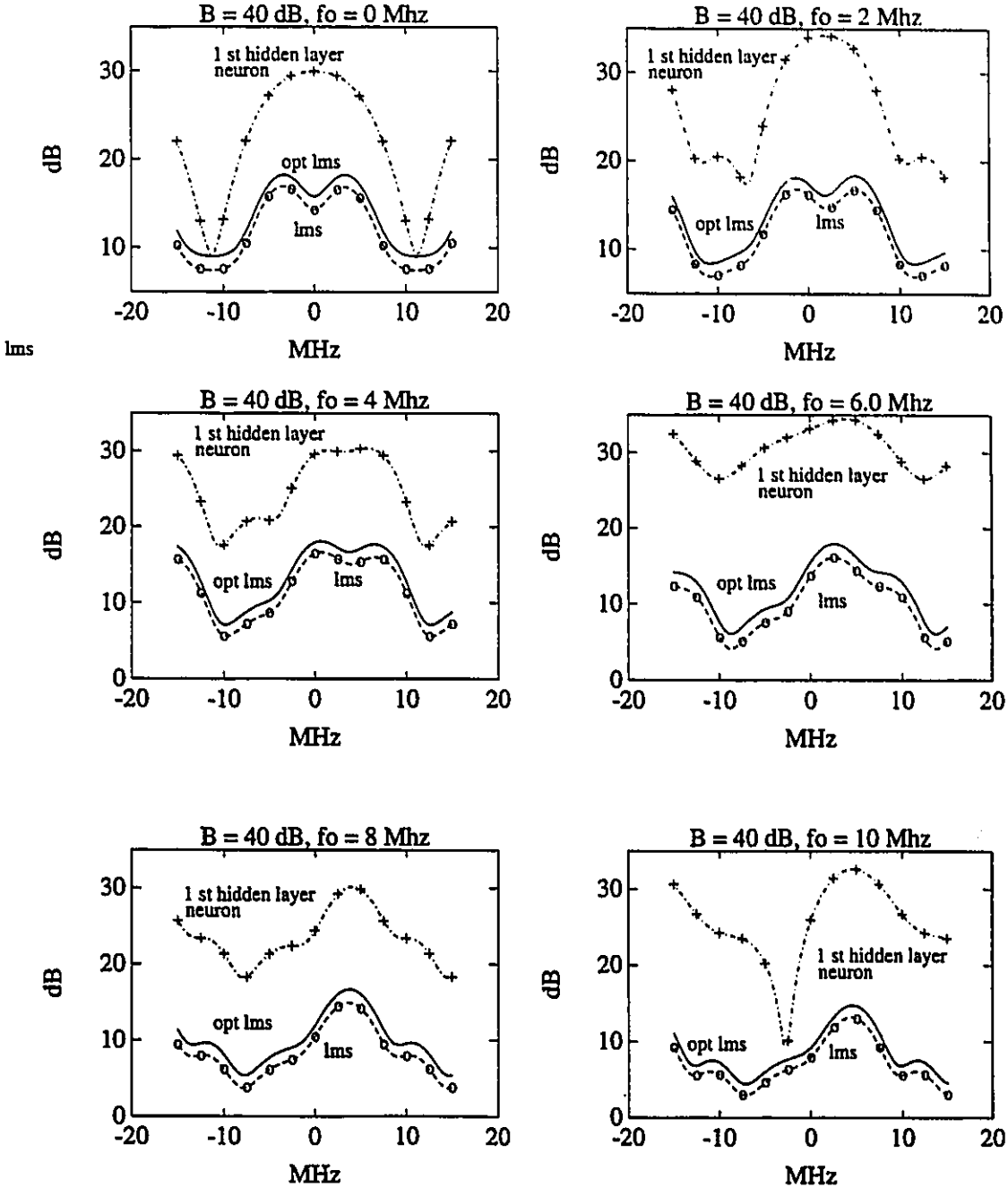

Figure 4.15: Frequency responses for the 4-QAM 1 0 1nl network for a 40 dB notch at 0, 2, 4, 6, 8, and 10 MHz.

Figure 4.16:

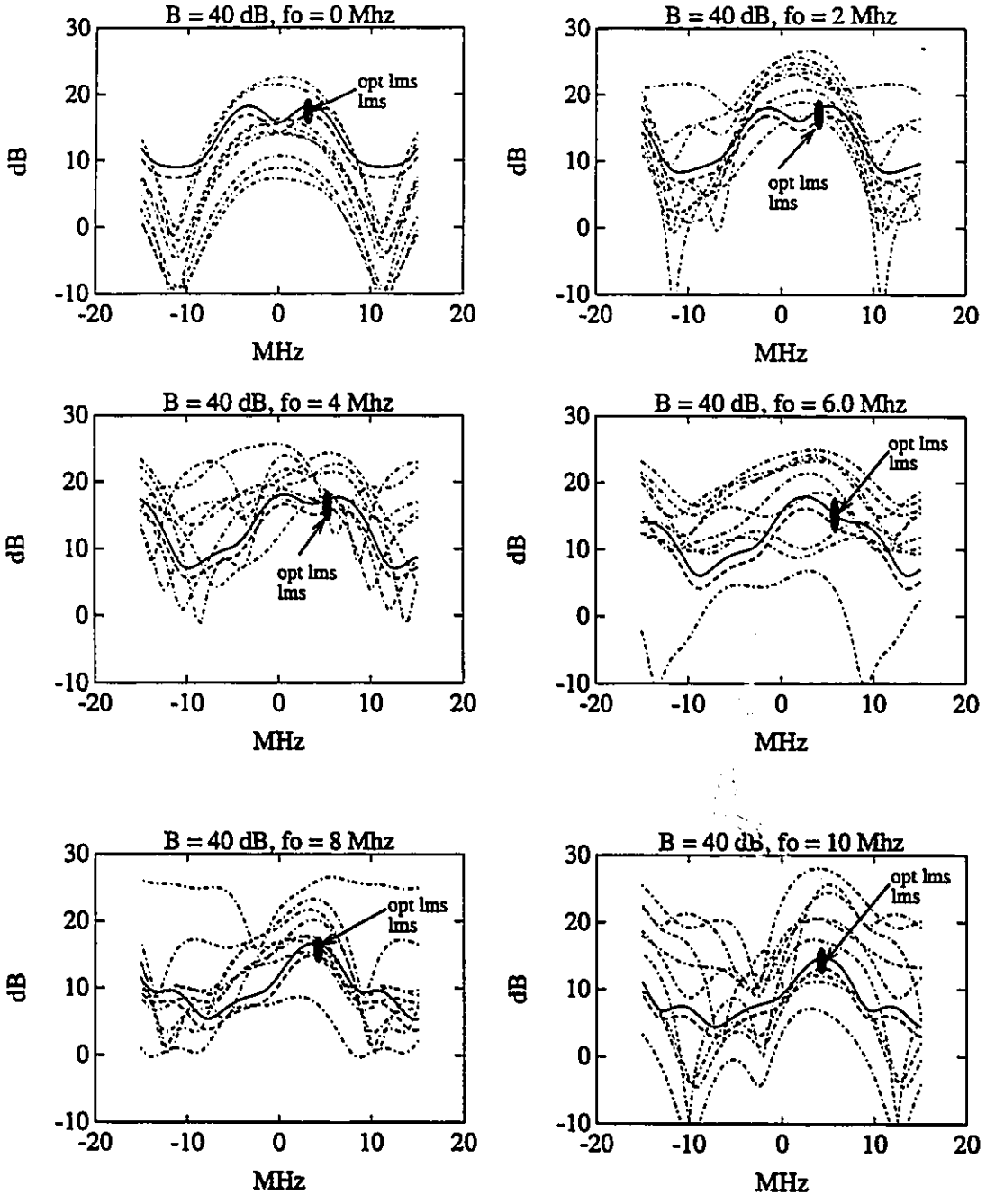

Figure 4.17: Frequency responses for the 4-QAM 951nl network for a 40 dB notch at 0, 2, 4, 6, 8, and 10 MHz. The frequency responses of the optimum LMS (solid line) and simulated LMS (dashed line) equalizers are as shown. The rest of the curves correspond to the frequency responses of the neurons of the first hidden layer of the 951nl network.

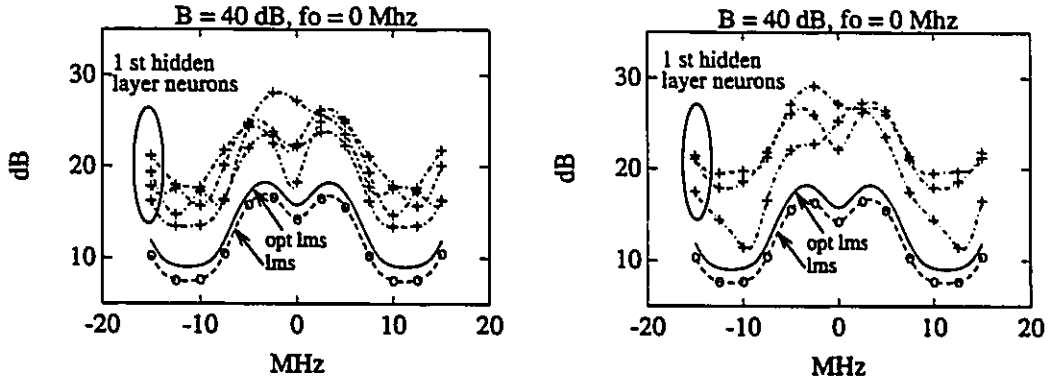

Figure 4.18:

Figure 4.19: Frequency responses for the 16-QAM $404nl$ (left) and $303nl$ (right) networks for a 40 dB notch at 0 MHz.

4.6 Summary

We have shown the application of a complex neuron to neural networks for adaptive channel equalization and have evaluated several neural network structures for both their SNR performance and outage performance. It was found that neural network equalizers offered equivalent SNR performance to the linear transversal equalizer. In terms of outage performance, it was found for 4-QAM (binary signalling in two dimensions) that even very simple neural networks (*e.g.* $1\,0\,1nl$) offered better outage performance than linear transversal equalizers. Unfortunately, this result is not reflected in higher level signalling schemes. Rather, the performance of the neural network equalizer tends to parallel that of the linear equalizer. It is known that the performance of the linear equalizer can be significantly improved (within limits) by increasing its order, i.e. the number of taps and using a suitably chosen decision delay. This is done with very little increase in complexity as compared to an equivalent order neural network. Hence for constellations larger than 4-QAM it would appear to be better to improve system performance through the use of a linear equalizer with a sufficient number of taps. All of this assumes that one makes an appropriate choice for the equalizer decision delay. Typically in the case of an unknown channel response a sufficient number of taps (e.g. 9) is chosen and the decision delay is chosen to correspond to the middle tap. When one is interested in minimizing the decision delay of the equalizer Gibson and Cowan *et al* [28]-[38] have shown that neural network structures offer superior performance over linear equalization schemes through their ability to achieve the functional mappings of the optimal Bayesian detector or estimator. However, much of the performance of the optimum detector/estimator may be achieved without the complexity of neural networks through the use of a linear transversal equalizer or decision feedback equalization by simply increasing the size of the tapped delay line(s) (number of taps) within reasonable limits and using a suitably chosen decision delay. This is particularly so when data pulse shapes are used that extend over numerous symbol intervals such as the spectral raised cosine pulse. Such pulses give rise to a very large number of channel states, reducing the effectiveness of practical Bayesian detection/estimation systems over conventional

equalization systems, especially when the number of signally levels is large.

In extending neural networks to the use of complex data, the use of complex neurons in MLP structures has allowed an interpretation of the weights of the first hidden layer in the frequency domain. This has shown that the frequency response of this section tends to have an inverse channel response that is characteristic of the response of a linear transversal equalizer. The exception is in the case of very deep notches when 4-QAM is used. Here the frequency response of the MLP has a better inverse channel response than the LTE.

This suggests some possibilities for increasing the performance of the NNTE. One way might be to precede the NNTE with a LTE. This has been tried for a fixed optimum 5 tap LTE followed by an adaptive 5 tap LNNTE (1011 and 3011). It was found that this did improve performance but that it was still less than that achieved with a 9 tap LTE. Note that if one convolves the response of the 5 tap LTE with that of the linear section of the 5 tap LNNTE the overall structure is a 9 tap equalizer. Even though both structures are 9 tap equalizers the 9 tap LTE will be superior because it has more degrees of freedom in choosing its weights than the combined structures of the fixed optimum LTE and the adaptive LNNTE (or NNTE).

Another approach would be to initialize the weights of the neurons in the first hidden layer to be those of a LTE (within an amplitude scaling factor) for a given channel. This poses several problems. Such as: the channel response is generally unknown and thus so is its inverse, how to choose the bias levels for the neurons, what should the relative amplitude scaling factor be and how does one organize the other hidden layers and the output layer? This is a risky approach but none the less it may have possibilities.

Chapter 5

Rayleigh Fading

5.1 Introduction

The work in this chapter utilizes work by Dam and Taylor[82]. That work developed an adaptive maximum likelihood receiver for differential quadrature phase shift keying (DQPSK) for rapid Rayleigh fading channels based on sub-sampling the received signal. It was assumed that the Rayleigh fading was correlated and could be modeled as a low pass filtered complex Gaussian noise process, as described in [89].

In this chapter we look at the feasibility of using a neural network as a phase predictor to reduce the error rate for DQPSK signalling in rapid Rayleigh fading. With this network it is possible to remove the error floor that is normally encountered in this channel for average signal to noise ratios of less than 60 dB. By making use of the constant amplitude characteristic of PSK signalling and through oversampling the received signal it is possible to form estimates of the channel fading process. These estimates may then be used by the neural network to predict the phase of the next sample. It is assumed that the band limiting filters used in the transmitter and receiver are wide enough not to significantly affect the signal characteristics. By training a small complex neural network over a group of channel characteristics ranging from slow to fast fading it is possible for the network to act as a generalized phase predictor. Once the network has learned the training set training may be stopped. The network may then be used in a fixed form to act as a generalized phase

predictor over the channel characteristics of interest. Provided the predicted phase jumps are not significantly different from the actual phase jumps due to the channel correct data decisions can be made. This results in a significant reduction in the error rate of the system.

The next section describes the channel model that has been used to simulate the Rayleigh fading process. This is followed by a section on the detection of signals in the Rayleigh fading channel. Under this section the optimum maximum likelihood (ML) quadratic receiver is discussed and the development of the neural network phase predictor is presented. This section is followed by sections on network training and simulation results. A short summary is presented at the end of this chapter.

5.2 Channel Model

The channel model is based on the Rician model presented in [89]. In our case we are only interested in Rayleigh fading, hence the direct signal path has been removed from the Rician model. The resulting complex baseband channel model is a low pass filtered complex Gaussian noise source. For a PSK signal the effect of the channel is to multiply the transmitted baseband signal, $e^{j\,\theta_d(t)}$, by the output of the filtered noise source, $r_c(t)\,e^{j\,\theta_c(t)}$ resulting in a received signal $r(t)$ of the form

$$r\,(t) = r_c(t)\,e^{j\,\theta_c(t)}\,e^{j\,\theta_d(t)} + n\,(t) \tag{5.1}$$

where $r_c(t)$ is the magnitude of the channel fade, $e^{j\,\theta_c(t)}$ is the phase rotation caused by the channel, $e^{j\,\theta_d(t)}$ is the complex baseband PSK signal with phase $\theta_d(t)$ and $n\,(t)$ is the complex Gaussian noise present in the received signal. The filter for the channel is chosen to be a third order Butterworth filter [89] with a bandwidth dependent upon the BT product of the fading process. The chosen filter BT products range from 0.01 to 0.1 which corresponds to the range of fading experienced by a 2400 bps DQPSK signal at 900 MHz in mobile radio communication. The work in [89] is based upon using one sample per baud to simulate the transmission system. The Butterworth filter is an infinite impulse response (IIR) filter whose z-transform function, $H_c(z)$, is

given as:

$$H_c(z) = \frac{1}{1 - \sum_{k=1}^{N} a_k z^{-k}} \tag{5.2}$$

where N is the order of the filter which in our case is 3 and the $a_k's$ are the real valued coefficients of z^{-k}.

For our purposes it was desired to use 8 samples per baud rather than a single sample so that it would be possible to sub-sample the received signal. To simulate this system, third order Butterworth filters (operating at the baud rate) were calculated based on BT products in the range of 0.01 to 0.1. The process used to generate the filters for 8 samples per baud is shown in Figure 5.1. To generate a filter, the frequency response of the third order Butterworth filter, $H_c(e^{j2\pi f_s n/N})$, was evaluated for 512 points ($N = 512$, $n = 0, \ldots, 512-1$) over the entire unit circle in the z-domain. This resulted in samples of the frequency response being taken in increments of $512/f_s$, where f_s is equal to the baud rate. As the bandwidth of the filter is much less than the bandwidth of the signal, the frequency response of $H_c(z)$ very quickly falls off to zero, e.g. approximately 0.01 to 0.1 times f_s. Effectively, by sampling at the baud rate we are sampling at a frequency that is much higher than the highest frequency component of the channel filter $H_c(z)$. By suitably scaling $H_c(e^{j2\pi f_s n/512})$ it is possible to calculate the sampled frequency response of $H_c(z)$ based on 8 samples per baud, e.g. $H_c(e^{j2\pi 8 f_s n/4096})$ by using 8×512 (4096) samples and setting the first 256 samples of $H_c(e^{j2\pi 8 f_s n/4096})$ equal to the first 256 scaled samples of $H_c(e^{j2\pi f_s n/512})$ and setting the last 256 samples of $H_c(e^{j2\pi 8 f_s n/4096})$ equal to the last 256 scaled samples of $H_c(e^{j2\pi f_s n/512})$. The scaling of the samples of $H_c(e^{j2\pi f_s n/512})$ is necessary due to the properties of the discrete Fourier transform used to evaluate the frequency responses of the filters. The rest of the samples of $H_c(e^{j2\pi 8 f_s n/4096})$ are zero as these samples correspond to frequencies that are higher than the bandwidth of $H_c(z)$.

To generate samples of the fading process a 4096 point sample of a complex Gaussian noise process is filtered by taking the FFT of the sample, multiplying it by the 4096 point frequency response of the channel filter, $H_c(e^{j2\pi 8 f_s n/4096})$, and then taking the inverse FFT of the result. Note that it is necessary to scale the variance of the noise so that the mean value for the depth of the fading process is 1.

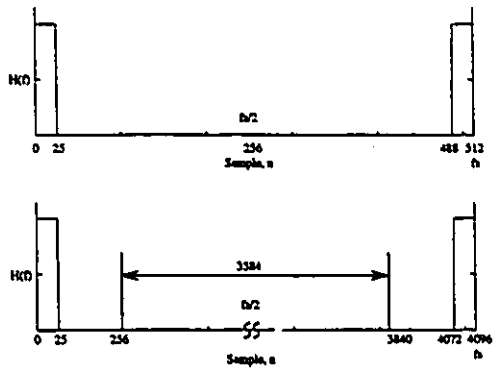

Figure 5.1: Process showing the generation of the frequency response of the channel filter based on 8 samples per baud.

To simulate the transmission system, 512 random signal points of a 4-PSK signal were generated. This signal was then eight times over sampled resulting in a signal with 4096 samples. The resulting samples were then multiplied by the samples from the channel fading model to form the faded signal. Additive white Gaussian noise was added to this signal to simulate the noise in the receiver.

To simulate systems using less than eight samples per baud the received signal was decimated by averaging together the required number of samples. For example, to simulate a system using four samples per baud, the eight samples of the received signal during one baud period are broken down into four pairs which are averaged together to form the four samples the received signal. To use 1 sample per baud, all eight samples during one baud period are averaged together to form a single sample. This is equivalent to using an integrate and dump system that over samples the received signal at a multiple of the baud rate. This is shown in Figure 5.2

5.3 Signal Detection

To remove the effects of rapid amplitude fluctuations due to the channel a constant amplitude signal is desired. In such a signal the information is transmitted in the phase of the signal rather than in its amplitude. The only effect amplitude fluctuations have on such a signal is to change the instantaneous signal to noise ratio.

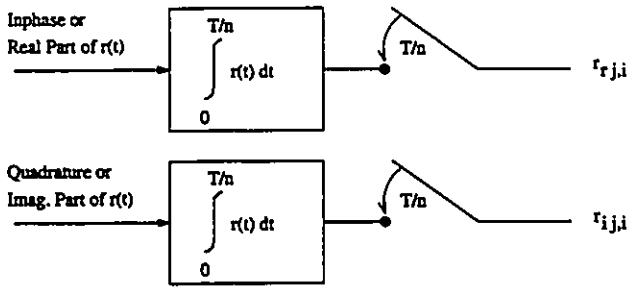

Figure 5.2:

Unfortunately not much can be done about this. As it is difficult to perform coherent detection due to the rapidly varying[1] phase of the signal due to the channel, differential detection can be used. To perform differential detection the receiver effectively integrates or averages the received signal over a symbol period and compares this result with the result obtained during the previous symbol period to determine the phase difference between the two symbols. The receiver makes the assumption that the phase of the signal due to the effects of the channel is relatively constant over this time, i.e. it varies by less than $\pi/4$ radians for DQPSK.

Our fundamental assumption is that the phase change resulting from the channel is not constant over a symbol interval[124][125][126][128][82]. To compensate for this, it is desirable to take multiple samples (say N) of the signal during a symbol period. To perform differential detection a vector $\mathbf{r}$ is formed consisting of these samples taken over a period of two symbols intervals, j to $j+1$ such that $\mathbf{r}^T = [\mathbf{r}_j^T, \mathbf{r}_{j+1}^T]$. That is, $\mathbf{r}$ is a vector of $2N$ elements where the i^{th} element is given as

$$r_i = r_{c_i}\, e^{j\,\theta_{c_i}}\, e^{j\,\theta_{d_i}} + n_i \tag{5.3}$$

The i^{th} elements of $\mathbf{r}_j^T$ and $\mathbf{r}_{j+1}^T$ are given as

$$\begin{aligned} r_{j,i} &= r_{c_{j,i}}\, e^{j\,\theta_{c_{j,i}}}\, e^{j\,\theta_{d_j}} + n_{j,i} && 1 \le i \le N \\ r_{j+1,i} &= r_{c_{j+1,i}}\, e^{j\,\theta_{c_{j+1,i}}}\, e^{j\,\theta_{d_{j+1}}} + n_{j+1,i} && N+1 \le i \le 2N \end{aligned} \tag{5.4}$$

respectively. Note that the energy of the transmitted PSK signal has been normalized to unity over the sampling instant.

[1]By rapidly varying we mean that the phase changes significantly over a few symbol periods.

For ordinary differential detection the receiver detects the phase difference between the average of the first N elements of $\mathbf{r}$ and the last N elements of $\mathbf{r}$. The maximum likelihood quadratic receiver [82] utilizes the decision variables, l_m

$$\begin{aligned} l_m &= \mathbf{r}^H \mathbf{K}_m^{-1} \mathbf{r} \\ l_m &= \mathbf{r}^H \mathbf{S}_m \mathbf{K}_c^{-1} \mathbf{S}_m^H \mathbf{r} \end{aligned}$$

where H denotes Hermitian transpose, m represents one of the 4 possible phase differences for DQPSK, $\mathbf{K}_c$ is the $2N \times 2N$ channel correlation matrix and

$$\mathbf{S_m} = \begin{bmatrix} \mathbf{I}_{N\times N} & \mathbf{0} \\ \mathbf{0} & e^{j\theta_m}\mathbf{I}_{N\times N} \end{bmatrix}$$

for constant amplitude DQPSK where θ_m is the m^{th} element of $\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$. Note that

$$\mathbf{K}_m^{-1} = \mathbf{S}_m \mathbf{K}_c^{-1} \mathbf{S}_m^H$$

This receiver requires knowledge of the channel correlation matrix, $\mathbf{K}_c$, and the ability to track it over time [82]. We are interested in the application of neural networks to compensate for the phase rotation introduced by the channel. The approach considered here is to use a neural network to predict $e^{j\theta_{c_i}}$ based upon N previous channel samples, $r_{c_{i-N}} e^{j\theta_{c_{i-N}}}, \ldots, r_{c_{i-1}} e^{j\theta_{c_{i-1}}}$. Provided that the predicted phase rotation, $e^{j\hat{\theta}_{c_i}}$, is reasonably accurate, it is possible to remove the effects of the channel phase rotation by multiplying the i^{th} received sample by $e^{-j\hat{\theta}_{c_i}}$.

The predicted phase rotation $e^{j\hat{\theta}_{c_i}}$ may be equivalently represented as $\cos\hat{\theta}_{c_i} + j\sin\hat{\theta}_{c_i}$. As both the *cos* and *sin* functions are nonlinear and bounded by ± 1 the complex backpropagation network with a *tanh* nonlinearity would appear to be an ideal choice for such a predictor. An advantage to using a neural network predictor is that it may be possible to use the generalization capability of neural networks to help track the changes in the fading process, i.e. as the BT product varies within the range 0.01 - 0.1 due to vehicular motion

The use of a predictor is predicated on the assumption that estimates of the channel fading process are available. This posses a problem as the only estimate

of the channel in a conventional DQPSK receiver is the received signal which is a combination of the channel fading process and the desired data signal. As the data signal has constant amplitude, it is then possible to use the received signal to estimate the channel fading process by removing the effects of the transmitted data phase. In a practical situation one does not have prior knowledge of the transmitted signal so that the effects of the data phase can not be removed from the channel estimates. The received signal samples during the j^{th} data symbol, θ_{d_j}, are given as

$$r_{j,i} = r_{c_{j,i}} e^{j\theta_{c_{j,i}}} e^{j\theta_{d_{j,i}}} + n_{j,i}$$

Note that $\theta_{d_{j,i}}$ is constant for $1 \leq i \leq N$ (i.e. $\theta_{d_{j,i}} = \theta_{d_j}$) and for $N+1 \leq i \leq 2N$ (i.e. $\theta_{d_{j+1,i}} = \theta_{d_{j+1}}$). The effect of the data phase on the first N samples may be overcome by removing the average phase, θ_{r_j}, of the first N samples of $\mathbf{r}$ from all of the elements in $\mathbf{r}$. This leaves the relative phase changes from one sampling instant to another.

The reference phase θ_{r_j} is the phase of $\alpha_j e^{j\theta_{r_j}}$ where $\alpha_j e^{j\theta_{r_j}}$ is given as

$$\begin{aligned} \alpha_j e^{j\theta_{r_j}} &= \frac{1}{N}\sum_{i=1}^{N} r_{j,i} \\ &= \frac{1}{N}\sum_{i=1}^{N}(r_{c_{j,i}} e^{j\theta_{c_{j,i}}+\theta_{d_{j,i}}} + n_{j,i}) \end{aligned} \tag{5.5}$$

by letting $n_{j,i} = e^{j\theta_{d_{j,i}}}\tilde{n}_{j,i}$ where $\tilde{n}_{j,i} = e^{-j\theta_{d_{j,i}}} n_{j,i}$ and using the fact that $\theta_{d_{j,i}} = \theta_{d_j}$ then

$$\begin{aligned} \alpha_j e^{j\theta_{r_j}} &= \frac{1}{N}\sum_{i=1}^{N}(r_{c_{j,i}} e^{j\theta_{c_{j,i}}+\theta_{d_{j,i}}} + e^{j\theta_{d_i}}\tilde{n}_{j,i}) \\ &= \frac{e^{j\theta_{d_j}}}{N}\sum_{i=1}^{N}(r_{c_{j,i}} e^{j\theta_{c_{j,i}}} + \tilde{n}_{j,i}) \\ &= \frac{e^{j\theta_{d_j}}}{N}\bar{r}_{c\tilde{n}_j} e^{j\bar{\theta}_{c\tilde{n}_j}} \end{aligned} \tag{5.6}$$

where $\alpha_j = \bar{r}_{c\tilde{n}}/N$ and $\theta_{r_j} = \bar{\theta}_{c\tilde{n}_j} + \theta_{d_j}$.

Upon removing the average phase, θ_{r_j} from the $2N$ samples of $\mathbf{r}$ the first N samples of $\mathbf{r}$ $(1 < i \leq N)$ become

$$r_{j,i} = r_{c_{j,i}} e^{j(\theta_{c_{j,i}}+\theta_{d_j}-\theta_{r_j})} + n_{j,i} e^{-j\theta_{r_j}} \tag{5.7}$$

$$
\begin{aligned}
&= r_{c_{j,i}} e^{j[(\theta_{c_{j,i}} - \bar{\theta}_{c\tilde{n}_j}) + (\theta_{d_j} - \theta_{d_j})]} + n_{j,i} e^{-j\theta_{r_j}} \\
&= r_{c_{j,i}} e^{j(\theta_{c_{j,i}} - \bar{\theta}_{c\tilde{n}_j})} + n_{j,i} e^{-j\theta_{r_j}} \\
&= r_{c_{j,i}} e^{j\Delta\theta_{c_{j,i}}} + n_{j,i} e^{-j\theta_{r_j}}
\end{aligned}
$$

and the last N samples ($N < i \leq 2N$) become

$$
\begin{aligned}
r_{j+1,i} &= r_{c_{j+1,i}} e^{j(\theta_{c_{j+1,i}} + \theta_{d_{j+1}} - \theta_{r_j})} + n_{j+1,i} e^{-j\theta_{r_j}} \\
&= r_{c_{j+1,i}} e^{j[(\theta_{c_{j+1,i}} - \bar{\theta}_{c\tilde{n}_j}) + (\theta_{d_{j+1}} - \theta_{d_j})]} + n_{j+1,i} e^{-j\theta_{r_j}} \\
&= r_{c_{j+1,i}} e^{j[(\theta_{c_{j+1,i}} - \bar{\theta}_{c\tilde{n}_j}) + \Delta\theta_{d_{j+1,j}}]} + n_{j+1,i} e^{-j\theta_{r_j}} \\
&= r_{c_{j+1,i}} e^{j(\Delta\theta_{c_{j+1,i}} + \Delta\theta_{d_{j+1,j}})} + n_{j+1,i} e^{-j\theta_{r_j}}
\end{aligned}
\tag{5.8}
$$

Note that the effect of the data signal is removed from $\mathbf{r}_j$ but not from $\mathbf{r}_{j+1}$. The samples in $\mathbf{r}_{j+1}$ contain the desired channel information offset by $\Delta\theta_{d_{j+1,j}}$, the desired phase difference between $\mathbf{r}_j$ and $\mathbf{r}_{j+1}$. $\Delta\theta_{d_{j+1,j}}$ is determined by searching over the set of possible values for $\Delta\theta_{d_{j+1,j}}$. For each of the m possible values for $\Delta\theta_{d_{j+1,j}}$, its effect $e^{j\Delta\theta_{dm_{j+1,j}}}$ is removed from the last N samples in $\mathbf{r}$ to form the channel samples for the phase predictor. The resulting predicted phase estimate based upon these channel estimates is $e^{j\hat{\theta}_{cm_{j,i}}}$. The performance of the predictor, l_m, based upon the removal of $\Delta\theta_{dm_{j+1,j}}$ is measured by calculating the sum of the squared distances between the predicted phase rotations and the phase rotations of the channel samples weighted by the squared magnitude of the channel samples

$$
\begin{aligned}
l_m = \sum_{i=N+1}^{2N} |r_{c_{j+1,i}}|^2 \; & \{(e^{j\Delta\theta_{c_{j+1,i}} + \Delta\theta_{d_{j+1,j}}} + n_{j+1,i}) e^{-j\Delta\theta_{dm_{j+1,j}}} - e^{j\hat{\theta}_{cm_{j,i}}}\} \\
& \times \{(e^{j\Delta\theta_{c_{j+1,i}} + \Delta\theta_{d_{j+1,j}}} + n_{j+1,i}) e^{-j\Delta\theta_{dm_{j+1,j}}} - e^{j\hat{\theta}_{cm_{j,i}}}\}^*
\end{aligned}
$$

where $*$ denotes complex conjugation and $\Delta\theta_{dm_{j+1,j}} \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$ for DQPSK. $\Delta\theta_{d_{j+1,j}}$ is chosen as the $\Delta\theta_{dm_{j+1,j}}$ with the smallest l_m. The latter constitutes the receiver's decision.

5.3.1 Network Training

To train the neural network predictor a suitable training set must be determined. This poses an interesting problem in this case. It is desired to predict the phase changes

from one sample to another. In doing so it is more important to compensate for the phase jumps from one symbol period to another where the absolute phase changes by more than $\pi/4$ radians than when it changes by less than $\pi/4$ radians as this is when a symbol error will occur. The problem is that the these phase jumps occur relatively infrequently, anywhere from 1 in 10 to 1 in 1000 symbol periods depending upon the signal to noise ratio and the fading rate. If a continuous sample of a channel fading process is used as a training pattern then the number of "good" phase jumps ($< \pi/4$) is much greater than the number of "bad" phase jumps ($> \pi/4$). When a network is trained with this set it will tend to predict in favour of the good phase jumps rather the bad ones. What is required is to skew the training set so that the training set is more evenly weighted between good and bad training patterns. To do this it was decided to split the channel examples evenly between cases where the phase jump between symbols was less than $\frac{\pi}{4}$ (no data decision error) and examples where the phase jump was greater than $\frac{\pi}{4}$, i.e. the channel phase rotation was great enough to produce a data decision error. In order for the network to generalize its training the training set consisted of 10 channels corresponding to BT products from 0.01 to 0.1 in steps of 0.01. For each BT product 200 examples of the channel fading process were taken. Each example consisted of channel samples taken over a period of two symbols. This resulted in 2000 examples of the channel fading process. For each example the average phase over the first symbol interval, θ_r, was calculated and removed from all of the samples by multiplying them by $e^{-j\theta_r}$. This normalized the samples so that the average phase was centered at zero. The sampling rate was determined by the order of the predictor. For an N^{th} order predictor N samples per baud were taken. The sampling rates of interest were 2 and 4 samples per baud.

The neural network was trained upon the presentation of each channel sample. Experimentation showed that the minimum network size is a network consisting of 2 hidden layers of 2 complex neurons and 1 complex (nonlinear) output neuron. This is shown in Figure 5.3. Figures 5.4 and 5.5 show the learning curves for the 2 2 1 networks for 2 and 4 samples per baud with η's of 0.001, 0.01 and 0.1 for an average SNR of 60 dB. Figure 5.6 shows the learning curves for the network with 4 samples per baud at an average SNR of 40 dB. These curves represent the averaged performance

of 50 networks. The figures show that the learning speed increases as η is increased and that the residual mean squared error (MSE) is decreased. The networks trained with the higher η's while showing lower MSE on the training set actually performed worse when used on new channel data. The performance criterion for a network is based upon the symbol error rate of the receiver when the network is used as a phase compensator not on the MSE of the network when used on the training set. It is viewed that the networks with the faster learning times and lower MSE performance on the training set have over trained themselves. Best performance in terms of symbol error rate resulted when this network was trained with an adaptation parameter of $\eta = 0.001$ for 4000 presentations of the training set. Note that the learning rate for the network using 4 samples per baud appears to be faster than the network using 2 samples per baud. This is because there are twice as many training samples in the training set as the sampling rate is double that of the network using 2 samples per baud.

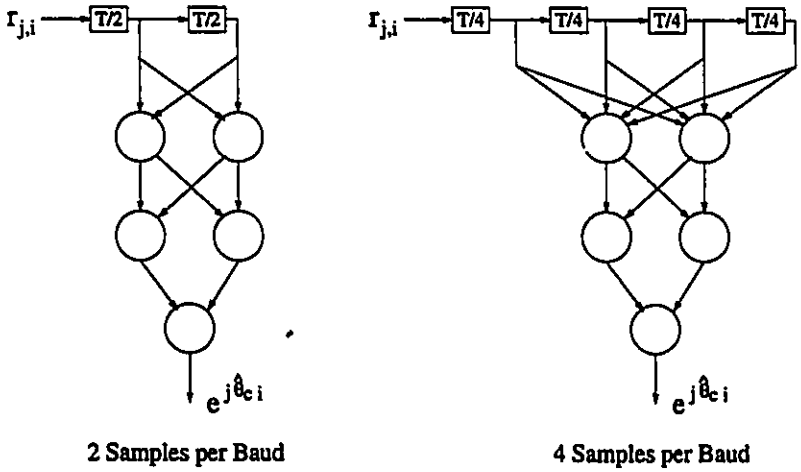

Figure 5.3: Diagram of the neural network phase predictor for 2 and 4 samples per Baud

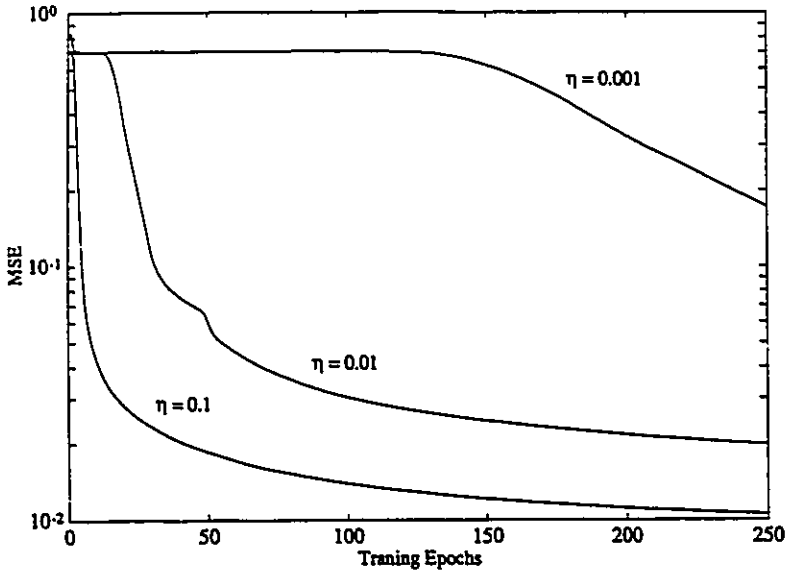

Figure 5.4: Mean squared error performance versus the number of training epochs 2 samples per baud with an SNR of 60 dB.

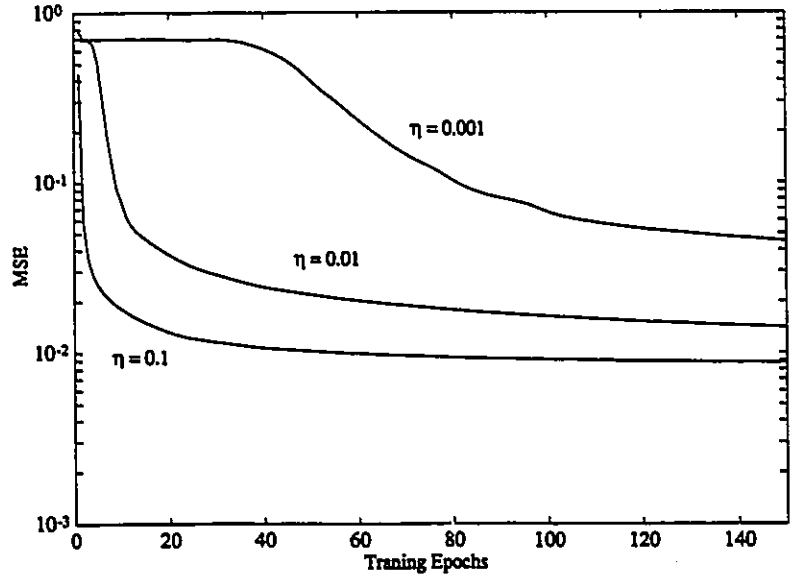

Figure 5.5: Mean squared error performance versus the number of training epochs 4 samples per baud with an SNR of 60 dB.

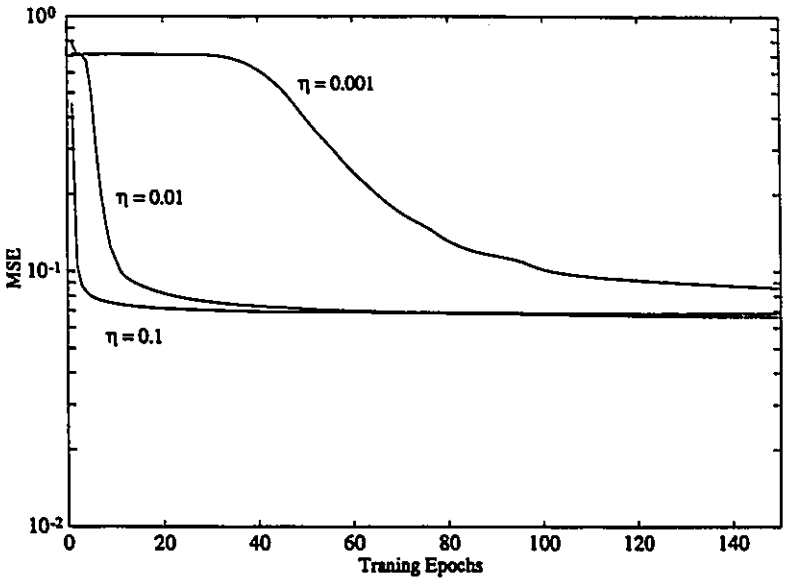

Figure 5.6: Mean squared error performance versus the number of training epochs 4 samples per baud with an SNR of 40 dB.

5.4 Simulation Results

To determine the performance of the phase predictor a network was first trained at a specific average SNR and then evaluated at that SNR with the training mode disabled. This was done for average SNR's ranging from 20 to 60 dB. These networks were evaluated for BT products of 0.1 and 0.04. Evaluation was accomplished by simulating the transmission of DQPSK through the channel at a fixed average SNR. At the receiver the affect of the data phase was removed from the channel estimates. The resulting channel estimates were then used by the predictor to compensate for channel phase rotation experience by the next signal sample. As the assumption was that the effects of the data signal were removed from the channel estimates it was not necessary to actually generate a data signal in this simulation. All that was necessary was to generate the channel fading process and to add in the appropriate amount of noise. A symbol error would only occur if after phase compensation the average residual phase of the signal (channel fading process + noise + phase compensation) during a symbol period was greater than $\pi/4$ radians.

This first system assumes that one has prior knowledge of the data signal so that its effects can be removed from the channel estimates. In general, this is not so but this allows us to form a lower bound on the performance of the system and to determine the accuracy of the phase predictor by comparing its performance with that of a receiver with perfect phase recovery. Figures 5.7 and 5.8 show the performance of the receiver with the neural network phase predictor compared to the performance of a receiver that has complete knowledge of the channel phase but which is still subjected to the magnitude of the fading process, i.e. a receiver with complete carrier recovery. Obviously the performance of the receiver with perfect phase recovery is superior to the neural network phase predictor but both receivers are capable of moving the error to below 10^{-5}. It can also be seen that the use of 4 samples per baud results in better predictor performance than for the use of 2 samples per baud. This is to be expected as the predictor is predicting over a smaller sampling time interval. One might expect that by increasing the number of samples per baud even further, e.g. 8, that there would be a further improvement in predictor

performance but what must be remembered is that for every doubling of the sampling rate the signal to noise ratio of the samples decreases by 3 dB as the noise is averaged over a smaller amount of time. There is obviously a trade off between the number of samples per baud and the signal to noise ratio of the samples.

The problem with implementing this type of system is how does one obtain accurate estimates of the fading process? One way to look at it is to assume that there is a "black box" that is capable of providing channel estimates for the last N samples but not for the present sample. This is, however, not a very realistic assumption but these results do serve to show the capability of the neural network predictor given accurate estimates of the channel fading process and hence form a lower bound on the performance of the predictor.

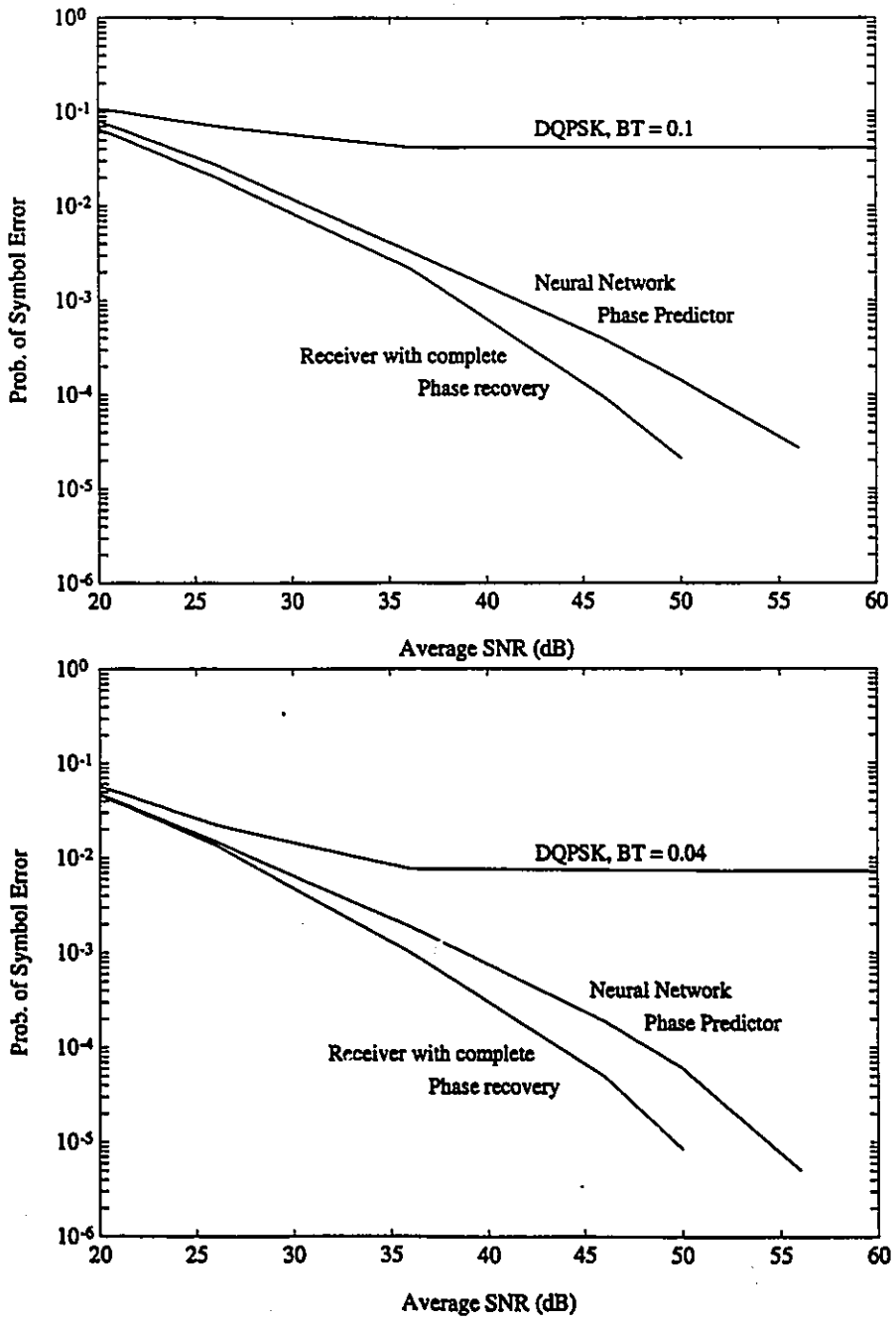

Figure 5.7: Performance of a receiver with perfect phase recovery and performance of the neural network with knowledge of the channel phase for BT's of 0.1 and 0.04 for 2 samples per baud.

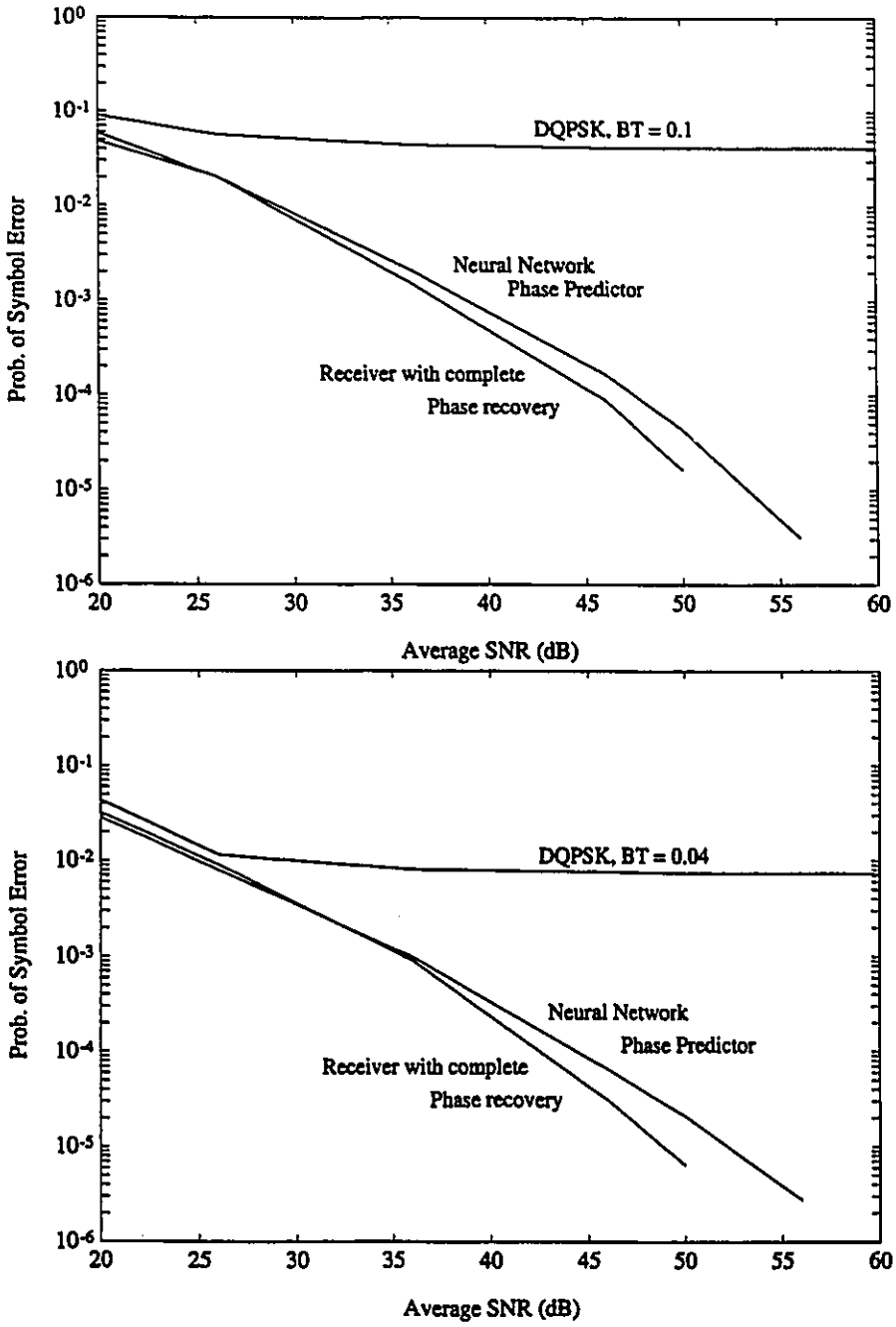

Figure 5.8: Performance of a receiver with perfect phase recovery and performance of the neural network with knowledge of the channel phase for BT's of 0.1 and 0.04 for 4 samples per baud.

The performance of the predictor based upon the search technique described in section 5.3 is shown in Figures 5.9 and 5.10. Also shown in these figures is the performance of the optimum quadratic receiver[82] from section 5.3. During the simulation neither the neural network or the maximum likelihood receiver was allowed to adapt as they were being evaluated at a specific BT product and signal to noise ratio. Under normal conditions the channel characteristics would be changing due in part to vehicular motion. Under such circumstances the maximum likelihood receiver would have to adapt to track the channel but a receiver using the neural network phase predictor would not have to adapt due to the generalization capabilities of the neural network. For the simulation $\mathbf{K}_c$ was calculated for each BT product at a specific signal to noise ratio by averaging over a large number of channel samples for that channel. The neural network, however, was trained at a specific signal to noise level over the range of BT products from 0.1 to 0.01.

As can be seen from the figures the performance of the neural network is quite close to that of the maximum likelihood receiver but tends to diverge slightly as the signal to noise ratio increases. For both the ML-receiver and the neural network the performance of the 4 samples per baud systems is better than that for 2 samples per baud. What can also be seen is that the performance of the neural network phase predictor is not as close to the performance of the ML-receiver for a BT of 0.04 as it is for 0.1. This suggests that the performance of the neural network predictor deteriorates from that of the ML-receiver as the BT product decreases. The important point here is that even with a fixed neural network (no training) it is possible to significantly lower the error floor of these systems. These results are consistent with results in [109] which show a bit error of 10^{-3} in the range of 35 to 40 dB for binary level partial response continuous phase modulation at BT products from 0.005 to 0.001.

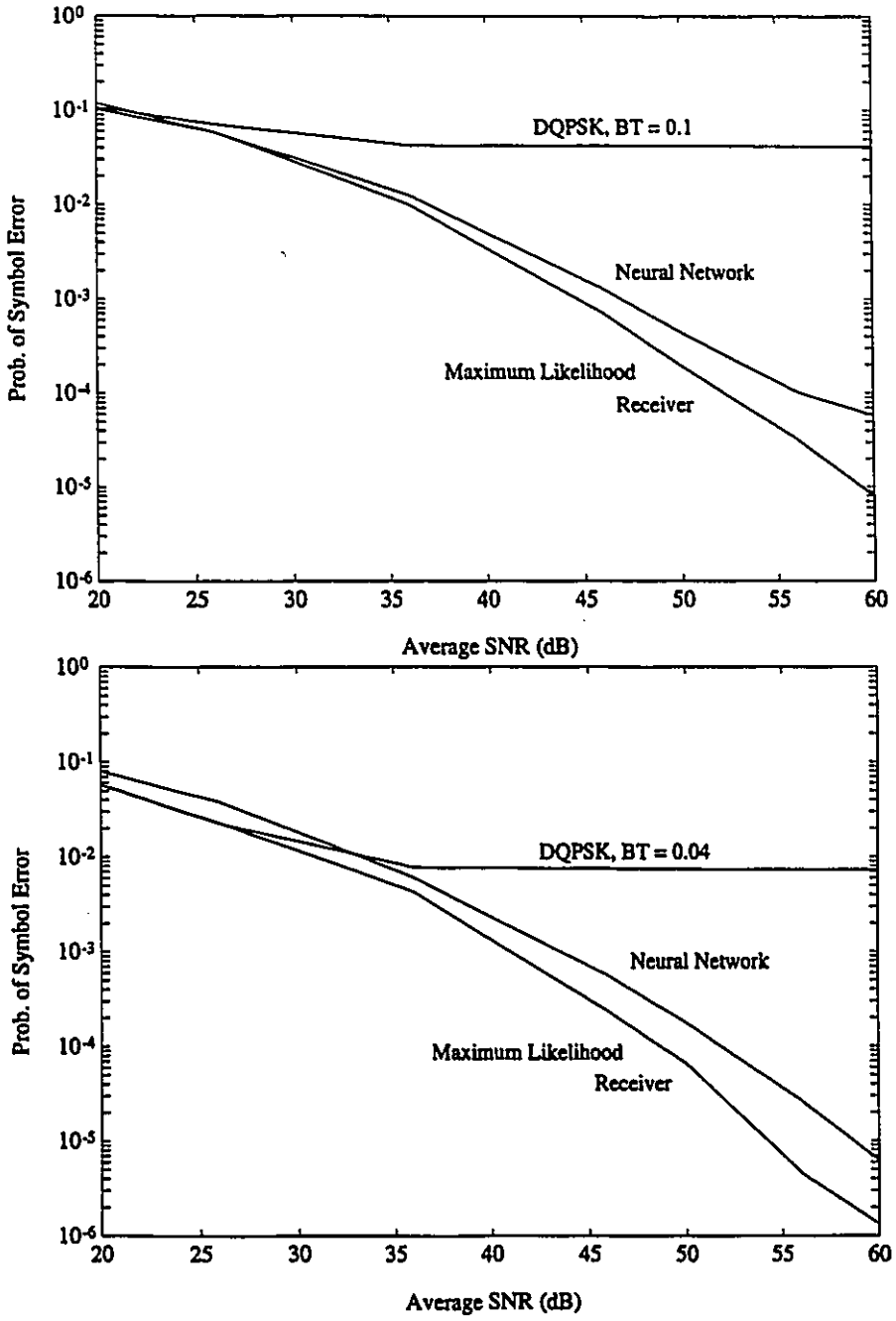

Figure 5.9: Performance of the maximum likelihood receiver and the neural network phase predictor for BT's of 0.1 and 0.04 for 2 samples per baud.

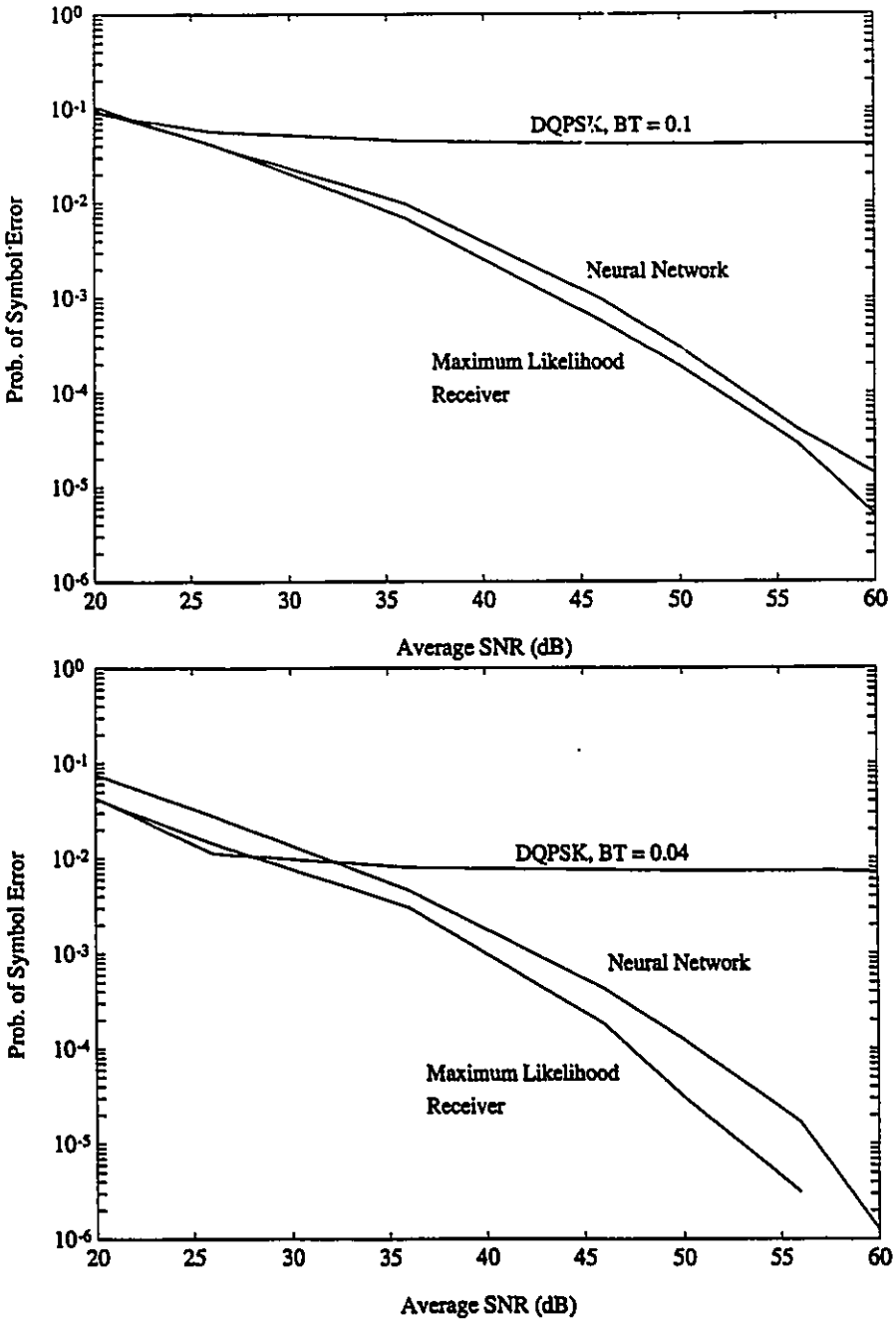

Figure 5.10: Performance of the maximum likelihood receiver and the neural network phase predictor for a BT's of 0.1 and 0.04 for 4 samples per baud.

5.5 Summary

Lowering the error floor for signalling in a rapid Rayleigh fading channel is a difficult problem. The work presented within this chapter has shown that with the use of a $2 \times 2 \times 1$ complex network it is possible to remove the error floor that is normally encountered in this channel for average signal to noise ratios of less than 60 dB. The performance of this predictor has been evaluated both in the case where the neural network has knowledge of the channel fading process and where estimates of the fading process must be obtained from the received signal. In the former case the performance of the neural network has been compared to that of a receiver with perfect phase recovery. In the latter case the performance of the neural network has been compared to that of the optimum maximum likelihood quadratic receiver and has been shown to be within 3 dB of the ML-receiver at a symbol error rate of 10^-3. The neural network phase predictor has demonstrated that for the differential detection of a QPSK signal it is capable of lowering the error floor normally encountered in this channel by several orders of magnitude. The advantage of the neural network is its simplicity and its generalization capabilities. Once the network has been trained it does not need to adapt to track changing channel characteristics whereas the maximum likelihood quadratic receiver must track changing channel conditions and continuously update its estimate of the inverse channel correlation matrix. However, the problem with a neural network phase predictor is that the network must be trained and used at a specific SNR. This is not a serious problem as it is possible to train a group of networks at various SNR's and to store their configuration information in memory. By determining the SNR of the signal the appropriate network configuration could be used. Determination of SNR could easily be achieved by looking at the automatic gain control characteristics of the receiver.

Chapter 6

Conclusions

The purpose of this thesis has been to study the application of multi-layered perceptrons to channel equalization/compensation schemes for two types of channels: the pure frequency selective fading channel and the correlated Rayleigh fading channel, a pure time selective fading channel. The goal of this research was not to propose neural networks as superior structures to present systems but to explore the possibilities of their use, to identify suitable network architectures, evaluate their performance and to lay a solid ground work for their application to channel equalization/compensation schemes. For this it was decided to look at multi-layered perceptron (MLP) structures using the standard back-propagation (BP) algorithm.

6.1 The AWGN Channel - Bayesian Estimation

To lay the ground work for work in the frequency selective fading channel the first part of this thesis considered the problem of Bayesian estimation of a digital signal in AWGN. While this is a simple problem it has been shown that that it has some important implications in the area of neural networks. Within this section, the link between a single neuron and the Bayesian estimator for 2-PAM in AWGN has been established. Provided that the weight of a single neuron (with one input) is appropriately chosen, the neuron is equivalent to the Bayesian estimator for 2-PAM. The choice of this weight is related to the noise present on the signal such that the weight

is proportional to the inverse of the noise variance.

By considering the Bayesian estimator function for N-PAM an activation function was presented that exhibits multiple (> 2) quantization levels when its gain parameter, α, is chosen sufficiently large yet when it is small it behaves as a soft limiter much like the $tanh(x)$ function. The use of this function may have significant implications in the area of multi-level nonlinear signal processing. However, to make full use of this function it is necessary to adapt the gain parameter of the function. This led to the idea of back propagating α. In regards to this, there has been some work done for binary neurons, $tanh(\alpha x)$, that indicates this leads to faster learning times[142]. This thesis establishes the exact link between the gain term α and the adaptation parameters for the back propagation algorithm. It is shown that adapting the gain term is equivalent to adapting the adaptation parameter η for each neuron in the neural network. In the case of a multi-level nonlinearity, back propagating the gain term leads to an effective change in the adaptation parameter as well as changing the characteristics of the nonlinearity function.

Further research needs to be done in determining the usefulness of applying the modified activation function, $tanh(\alpha x)$, and the multi-level activation function to more complicated problems. An application of the Bayesian estimator function and hence possibly neural networks, is in the area of blind equalization[158]. In blind equalization a transmitted data sequence is convolved with an unknown channel disturbance and it is the job of the equalizer to de-convolve the receive sequence to obtain the transmitted sequence without any knowledge of transmitted sequence other than its probability distribution. In this case the use of Bayesian estimator functions falls under the heading of Bussgang techniques[158, 159]. In addition, other application areas for neural networks are in image processing, multi-valued logic[146] and in analogue to digital conversion[160].

6.2 Neural Network Channel Equalization

In studying the application of MLPs to the problem of nonlinear equalization in the frequency selective fading channel, previously published work [28]-[38] has been

extended from 2-PAM to 4 and 16-QAM (from two levels in one dimension to multiple levels in two dimensions). Of particular importance is that the work presented within this thesis includes the effects of pulse shaping and makes use of a widely accepted more realistic channel model than has been used in published work on neural network equalization.

It was found that neural network equalizers offered equivalent SNR performance to the linear transversal equalizer. In terms of outage performance, it was found for 4-QAM (binary signalling in two dimensions) that even very simple neural networks (e.g. $1\,0\,1nl$) offered better outage performance than linear transversal equalizers. Unfortunately, this result is not reflected in higher level signalling schemes. Rather, the performance of the neural network equalizer tends to parallel that of the linear equalizer. It is known that the performance of the linear equalizer can be significantly improved (within limits) by increasing its order, i.e. the number of taps and using a suitably chosen decision delay. This is done with very little increase in complexity as compared to an equivalent order neural network. Hence for constellations larger than 4-QAM it would appear to be better to increase system performance through the use of a linear equalizer with a sufficient number of taps. All of this assumes that one makes an appropriate choice for the equalizer decision delay. Typically in the case of an unknown channel response a sufficient number of taps (e.g. 9) is chosen and the decision delay is chosen to correspond to the middle tap. When one is interested in minimizing the decision delay of the equalizer Gibson and Cowan *et al* [28]-[38] have shown that neural network structures offer superior performance over linear equalization schemes through their ability to achieve the functional mappings of the optimal Bayesian detector or estimator. However, much of the performance of the optimum detector/estimator may be achieved without the complexity of neural networks through the use of a linear transversal equalizer or decision feedback equalization by simply increasing the size of the tapped delay line(s) (number of taps) within reasonable limits and using a suitably chosen decision delay. This is particularly so when data pulse shapes are used that extend over numerous symbol intervals such as the spectral raised cosine pulse. Such pulses give rise to a very large number of channel states, reducing the effectiveness of practical Bayesian

detection/estimation systems over conventional equalization systems, especially when the number of signally levels is large.

In extending neural networks to the use of complex data, the use of complex neurons in MLP structures has allowed an interpretation of the weights of the first hidden layer in the frequency domain. This has shown that the frequency response of this section tends to have an inverse channel response that is characteristic of the response of a linear transversal equalizer. The exception is in the case of very deep notches when 4-QAM is used. Here the frequency response of the MLP has a better inverse channel response than the LTE.

Overall, neural network equalizers seem particularly well suited to binary signalling schemes, 2-PAM, 4-QAM and 4-PSK (binary in the in-phase and quadrature channels) or systems where the data pulse shape exists for only a single symbol period or a small number of periods. Such pulse shapes lead to significantly fewer and more defined channel states. Hence they are more suitable for Bayesian detection and thus the application of neural network equalizers. It is apparent that what really needs to be studied are the performance differences between linear equalization, decision feedback equalization, Bayesian detection/estimation and maximum likelihood sequence estimation in a realistic transmission model in order to accurately define the benefits of using Bayesian detection/estimation and hence neural network based equalizers. An important aspect of this work will be to determine the affects of pulse shaping on equalizer performance.

Neural network structures will likely be significant in their application to nonlinear channels. In many cases nonlinearities in the transmission system limit the efficiency of the system. Neural networks with their associated nonlinear properties would seem to be likely candidates to achieve improved system performance. Such applications are in the linearization of power amplifiers and in the areas of magnetic recording media and fiber optics. In looking towards the future, neural networks may be of use in a combined coding-modulation-equalization or equalization-demodulation-decoding system. Ungerboeck has already shown the benefits of combined coding and modulation. It is only a natural step forward to add further systems to this. This will obviously lead to increased computational complexity and require

more detailed theory, but neural networks may offer a desirable short cut.

6.3 Rayleigh Fading

The application of multi-layered perceptrons in the fast correlated Rayleigh fading channel environment has led to the development of a neural network phase predictor which for the differential detection of a QPSK signal is capable of significantly reducing the error floor that is normally encountered in this channel. This predictor relies on the fact that for fast correlated Rayleigh fading the phase rotation introduced by the channel changes significantly over one symbol interval. By over sampling the received signal and by making use of the constant amplitude characteristic of PSK signalling it is possible to form estimates of the channel fading process. These estimates when used with the neural network phase predictor yield a receiver whose performance is close to that of the optimum maximum likelihood quadratic receiver. The advantage of the neural network lies in its simplicity and its generalization capabilities. Only a $2 \times 2 \times 1$ complex network is needed, i.e. two hidden layers consisting of two complex neurons and an output layer consisting of one complex neuron. The maximum likelihood quadratic receiver on the other hand, requires the calculation of the inverse channel correlation matrix which for a sampling rate N times the baud rate is a $2N \times 2N$ matrix. Once the neural network phase predictor has been trained it does not need to adapt to track changing channel characteristics whereas the maximum likelihood quadratic receiver must do so and it must continuously update its estimate of the inverse channel correlation matrix. The advantage of the ML quadratic receiver is that it can track changing SNR conditions as opposed to the neural network which is trained at a specific SNR. This could easily be overcome by allowing the neural network to adapt when in use and developing a suitable training strategy. An alternative would be to train a group of neural networks to cover a range of SNRs and to store them in memory. A specific neural network could be retrieved for use depending on an estimate of the present SNR of the system. The neural network is small enough that a large number of networks could easily be stored in a small memory chip, e.g. a $2 \times 2 \times 1$ complex network using 16 bytes for each complex

weight (8 bytes per real number) requires only 304 bytes.

This work has shown that it is possible to remove the error floor that is encountered in Rayleigh fading channel conditions for SNRs of less than 60 dB. Additional work needs to be done to determine how feasible it is to actually implement such a structure. In particular, both the application of the neural network phase predictor and the ML quadratic receiver need to be studied under real channel fading conditions, not simulated ones. Further work could be done in studying the application of these techniques to the case where the transmitted signal experiences both Rayleigh fading and frequency selective fading such as is the case when the signal bandwidth is much larger than that considered here, e.g. greater than 30 or 50 kHz. Work also needs to be done to extend this work to the more general case where the amplitude of the signal is not constant and the length of the data pulse is not limited to the duration of a single symbol interval. Under these circumstances the estimation of the channel fading process is a much more difficult problem as more unknowns are introduced into the problem. Further, allowing the data pulse to extend outside of the symbol interval removes the assumption that the samples taken during a symbol interval are uncorrelated with those taken from another symbol interval, adding to the complexity of the problem.

Appendix A

Requirements for Equivalent Networks

In order to determine the necessary conditions for two networks based upon two different values of α to be equivalent during the learning process consider two networks with the same topology but with differing gain terms. With no loss of generality we may denote the gain term of the first network as $alpha_1$ and we may set the gain term of the second network to unity. The variables of the first network are denoted with a subscript of 1 and those of the second network with a subscript of 2. The outputs of the j^{th} neurons for the two networks x_{1j}, x_{2j}, are given as

$$\begin{aligned} x_{1_j} &= f_{\alpha_1}(net_{1_j}) \\ x_{2_j} &= f(net_{2_j}) \end{aligned}$$

where

$$f_{\alpha_1}(x) = \frac{1}{(1+e^{-\alpha x})} \tag{A.1}$$

$$f(x) = \frac{1}{(1+e^{-x})} \tag{A.2}$$

and

$$net_{1_j} = \sum_i x_{1_i} w_{1_{ij}} - \theta_{1_j}$$

$$net_{2_j} = \sum_i x_{2_i} w_{2_{ij}} - \theta_{2_j}$$

and where x_i is the output of the i^{th} neuron, w_{ij} is the weight linking the i^{th} neuron to the j^{th} neuron and θ_j is the bias term.

In order for the networks to be equivalent, that is they achieve the same functional mapping from the input space to the output space, we specify

$$w_{2_{ij}} = \alpha_1 \, w_{1_{ij}} \tag{A.3}$$

$$\theta_{2_j} = \alpha_1 \, \theta_{1_j} \tag{A.4}$$

hence

$$net_{2j} = \alpha_1 \, net_{1j} \tag{A.5}$$

$$x_{2_j} = x_{1_j} \tag{A.6}$$

Note that the activation functions are of the logistic (sigmoidal) type. This is done for clarity and simplicity. These results are easily extended to the $tanh\,(x)$ function.

In taking the derivative of the activation functions we obtain

$$f'_{\alpha_1}(x) = \frac{\alpha_1 \, e^{-\alpha_1 x}}{(1 + e^{-\alpha_1 x})^2}$$

$$f'(x) = \frac{e^{-x}}{(1 + e^{-x})^2}$$

hence

$$f'_\alpha(x) = \alpha_1 f'(\alpha x) \tag{A.7}$$

$$f'(\alpha x) = 1/\alpha_1 f'_\alpha(x) \tag{A.8}$$

The adaptation equations for the networks (no momentum term) are given as:

$$\Delta w_{1_{ij}} = \eta_1 \, \delta_{1_j} x_{1_i}$$

$$\Delta w_{2_{ij}} = \eta_2 \, \delta_{2_j} x_{2_i}$$

If the j^{th} neuron is an output neuron then δ_{1_j} and δ_{2_j} are defined as

$$\delta_{1_j} = f'_{\alpha j}(net_{1j})(d_{1_j} - x_{1_j}) \tag{A.9}$$

$$\delta_{2_j} = f'(net_{2j})(d_{2_j} - x_{2_j}) \tag{A.10}$$

where d_{1_j} and d_{2_j} are the desired responses for the j^{th} output neurons. As the two networks are considered to be equivalent, then for identical inputs to the networks

$$d_{1_j} = d_{2_j} \tag{A.11}$$

$$x_{1_j} = x_{2_j} \tag{A.12}$$

Substituting equations (A.11) and (A.12) into equation (A.10) and by making use of equations (A.5) and (A.8) we obtain

$$\delta_{2_j} = f'(\alpha_1 net_{1_j})(d_{1_j} - x_{1_j}) \tag{A.13}$$

$$= 1/\alpha_1 \, f'_{\alpha_1}(net_{1_j})(d_{1_j} - x_{1_j}) \tag{A.14}$$

$$= 1/\alpha_1 \, \delta_{1_j} \tag{A.15}$$

These equations can be recursively applied to the hidden layers of the networks. In this case

$$\delta_{1_j} = f'_{\alpha j}(net_{1j}) \sum_k \delta_{1_k} w_{1_{jk}} \tag{A.16}$$

$$\delta_{2_j} = f'(net_{2j}) \sum_k \delta_{2_k} w_{2_{jk}} \tag{A.17}$$

where k ranges over all neurons that are connected to the output of the j^{th} neuron. Substituting equations (A.15) and (A.3) into (A.17) yields

$$\delta_{2_j} = f'(net_{2j}) \sum_k 1/\alpha_1 \delta_{1_k} \alpha_1 w_{1_{jk}} \tag{A.18}$$

$$= 1/\alpha_1 \, \delta_{1j} \tag{A.19}$$

From equations (A.15) and (A.19) we may write

$$\begin{aligned} \Delta w_{1_{ij}} &= \eta_1 \, \delta_{1_j} x_{1_i} \\ \Delta w_{2_{ij}} &= 1/\alpha_1 \, \eta_2 \, \delta_{1_j} x_{2_i} \\ &= \frac{\eta_2}{\alpha_1 \eta_1} \Delta w_{1_{ij}} \end{aligned} \tag{A.20}$$

which gives the following update equations:

$$\begin{aligned} w_{1_{ijn+1}} &= w_{1_{ijn}} - \Delta w_{1_{ij}} \quad \text{at time } n+1 \\ w_{2_{ijn+1}} &= w_{2_{ijn}} - \Delta w_{2_{ij}} \quad \text{at time } n+1 \end{aligned} \tag{A.21}$$

In order for the networks to be equivalent at time $n+1$ we need to have

$$\begin{aligned} w_{2_{ij}n+1} &= \alpha_1 w_{1_{ij}n+1} \\ &= \alpha_1 w_{1_{ij}n} - \alpha_1 \Delta w_{1_{ij}} \end{aligned} \tag{A.22}$$

hence we need

$$\Delta w_{2_{ij}} = \alpha_1 \Delta w_{1_{ij}} \tag{A.23}$$

From equation (A.20) this will be satisfied if

$$\eta_2 = \alpha_1^2 \eta_1 \tag{A.24}$$

If we choose to include a momentum term, μ, then

$$\begin{aligned} \Delta w_{1_{ij}n+1} &= \eta_1 \delta_{1_j} x_{1_i} + \mu_1 \Delta w_{1_{ij}n} \\ \Delta w_{2_{ij}n+1} &= \eta_2 \delta_{2_j} x_{2_i} + \mu_2 \Delta w_{2_{ij}n} \end{aligned} \tag{A.25}$$

For the two networks to remain equivalent at time $t = n+1$ we require from equation (A.23)

$$\Delta w_{2_{ij}} = \alpha_1 \Delta w_{1_{ij}}$$

hence

$$\Delta w_{2_{ij}n} = \alpha_1 \Delta w_{1_{ij}n}$$

Thus, by choosing $\mu_1 = \mu_2$ we satisfy (A.23).

In general, for two networks with α_1 and α_2 to be equivalent, we require

$$\alpha_2^2 \eta_2 = \alpha_1^2 \eta_1 \tag{A.26}$$

$$\mu_2 = \mu_1 \tag{A.27}$$

Appendix B

Back Propagation of α

The change in the gain parameter, α, of a neuron is directly proportional to the negative of the gradient of the error power, that is

$$\Delta\alpha_j \propto -\frac{\partial E}{\partial \alpha_j} \tag{B.1}$$

where E is given as

$$E = \frac{1}{2}\sum_j (d_j - x_j)^2 \tag{B.2}$$

$\partial E/\partial \alpha_j$ may be written as

$$\frac{\partial E}{\partial \alpha_j} = \frac{\partial E}{\partial x_j}\frac{\partial x_j}{\partial \alpha_j} \tag{B.3}$$

where

$$\frac{\partial x_j}{\partial \alpha_j} = \frac{\partial f_j(\alpha_j net_j)}{\partial \alpha_j} \tag{B.4}$$

When the j^{th} neuron is an output node, $\partial E/\partial x_j$ is given as

$$-\frac{\partial E}{\partial x_j} = -(d_j - x_j) \tag{B.5}$$

In the case where the j^{th} neuron is in a hidden layer let us consider $\partial E/\partial net_j$ where

$$-\frac{\partial E}{\partial net_j} = -\frac{\partial E}{\partial x_j}\frac{\partial x_j}{\partial net_j} \tag{B.6}$$

$$= \delta_j \tag{B.7}$$

where

$$\delta_j = \frac{\partial f_j(\alpha_j net_j)}{\partial net_j} \sum_k \delta_k w_{jk} \tag{B.8}$$

where k ranges over all nodes that connect to the output of node j. If the j^{th} neuron is an output node then δ_j is defined as

$$\delta_j = \frac{\partial f_j(\alpha_j net_j)}{\partial net_j}(d_j - x_j) \tag{B.9}$$

where d_j is the desired output of node j and x_j is the actual output.

By rearranging the terms in equations B.6 and B.7 we obtain

$$\frac{\partial E}{\partial x_j} = -\frac{\delta_j}{\partial x_j / \partial net_j} \tag{B.10}$$

$$= -\frac{\delta_j}{\partial f_j(\alpha_j net_j)/\partial net_j} \tag{B.11}$$

Substituting equations B.4 and B.11 into B.1 we obtain

$$\Delta\alpha_j \propto \frac{\delta_j}{\partial f_j(\alpha_j net_j)/\partial net_j} \frac{\partial f_j(\alpha_j net_j)}{\partial \alpha_j} \tag{B.12}$$

If the j^{th} neuron is an output node then from equations B.9 and B.12

$$\Delta \alpha_j \propto (d_j - x_j) \frac{\partial f_j(\alpha_j net_j)}{\partial \alpha_j} \tag{B.13}$$

otherwise, if the j^{th} neuron is an internal, hidden node, then from equations B.8 and B.12

$$\Delta \alpha_j \propto \left(\sum_k \delta_k w_{kj} \right) \frac{\partial f_j(\alpha_j net_j)}{\partial \alpha_j} \tag{B.14}$$

where k ranges over all nodes that connect to the output of node j.

Bibliography

[1] Y. LeCun, "Models connexionistes de l'apprentissage," Master's thesis, Université Pierre et Marie Curie, 1987.

[2] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, pp. 2–2, 1989.

[3] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, pp. 183–192, 1989.

[4] A. R. Gallant and H. White, "There exists a neural network that does not make avoidable mistakes," in *Proceedings of the IEEE Second International Conference on Neural Networks*, pp. 657–664, 1988.

[5] R. Hect-Nielsen, "Theory of the back propagation neural network," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 593–606, 1989.

[6] K. Hornik, M. Stinchombe, and H. White, "Multilayer feed forward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[7] B. Irie and S. Miyake, "Capabilities of three layer perceptrons," in *Proceedings of the IEEE Second International Conference on Neural Networks*, pp. 641–648, 1988.

[8] C. K. Chui and L. Xin, "Approximation by ridge functions and neural netowrks with one hidden layer," *Journal of Approximation Theory*, vol. 70, pp. 131–141, 1992.

[9] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Networks*, vol. 6, pp. 861–867, 1993.

[10] S. U. H. Qureshi, "Adaptive equalization," *Proc. IEEE*, vol. 73, pp. 1349–1387, Sept. 1985.

[11] W. D. Rummler, R. P. Coutts, and M. Liniger, "Multipath fading channel models for microwave digital radio," in *Microwave Digital Radio*, pp. 67 –77, IEEE Press, 1988.

[12] R. W. Lucky, "A survey of the communication theory literature," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 725–739, Nov. 1973.

[13] R. Price, "Nonlinearly feedback-equalized pam vs. capacity for noisy filter channels," in *ICC'81*, pp. 22–12 to 22–17, IEEE, June 1972.

[14] S. Haykin, *Adaptive Filter Theory*. Prentice-Hall, 1986.

[15] B. Widrow, *Adaptive Signal Processing*. Prentice-Hall, 1985.

[16] C. A. Belfiore and J. Park Jr., "Decision feedback equalization," *Proceedings of the IEEE*, vol. 67, pp. 1143–1156, Aug. 1979.

[17] B. Widrow and M. E. Hoff Jr., "Adaptive switching circuits," in *IRE WESCON Conv. Rec.*, vol. pt. 4, pp. 96–104, Aug. 1960.

[18] B. Widrow, J. McCool, and M. Ball, "The complex lms algorithm," *Proc. IEEE*, vol. 63, pp. 719–720, Apr. 1975.

[19] R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, pp. 547–588, Apr. 1965.

[20] R. W. Lucky, "Techniques for adaptive equalization of digital communication systems," *Bell Syst. Tech. J.*, vol. 45, pp. 255–286, Feb. 1966.

[21] R. W. Lucky, J.Salz, and E. J. Weldon Jr., *Principles of Data Communication.* New York:McGraw-Hill, 1968.

[22] R. L. Plackett, "Some theorems in least squares," *Biometrika*, vol. 37, p. 149, 1950.

[23] D. D. Falconer and L. Ljung, "Applications of fast Kalman estimation to adaptive equalization," *IEEE Trans. Commun.*, vol. COM-26, pp. 1439–1446, 1978.

[24] J. M. Cioffi and T. Kailath, "Fast, recursive-least-squares transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-32, pp. 304–337, 1984.

[25] M. Morf and D. T. Lee, "Recursive least squares ladder forms for fast parameter tracking," in *Proc. 1978 Conf. decision and Control*, pp. 1362–1367, 1978.

[26] G. D. Forney Jr., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 363–378, May 1972.

[27] G. Ungerboeck, "Nonlinear equalization of binary signals in gaussian noise," *IEEE Trans. Comm.*, vol. COM-19, pp. 1128–1137, Dec. 1971.

[28] G. Gibson and C. Cowan, "Multilayer perceptron structures applied to adaptive equalisers for data communications," *IEEE Proceedings, ICASP*, pp. 1183–1186, May 1989.

[29] C. F. N. Cowan, P. M. Grant, S. Chen, and G. J. Gibson, "Non-linear classification and adaptive structures," in *Advanced Signal-Processing Algorithms, Architectures, and Implementations*, vol. 1348, (Bellingham, WA, USA.), pp. 62–72, Proceedings of SPIE - The International Society for Optical Engineering, Int Soc for Optical Engineering, July 1990.

[30] G. J. Gibson, S. Siu, S. Chen, C. F. N. Cowan, and P. M. Grant, "The application of nonlinear architectures to adaptive channel equalisation," in *ICC '90*, (IEEE Service Center, Piscataway, NJ, USA), pp. 649–653, IEEE, 1990.

[31] S. Chen, G.J.Gibson, and C. Cowan, "Adaptive channel equalisation using a polynomial-perceptron structure," *IEE Proceedings*, vol. 137, pp. 257–264, Oct. 1990.

[32] S. Chen, G. Gibson, C. Cowan, and P. Grant, "Adaptive equalization of finite non-linear channels using multilayer perceptrons," *Signal Processing 20*, pp. 107–119, 1990.

[33] S. Siu, G. Gibson, and C. Cowan, "Decision feedback equalisation using neural network structures and performance comparison with standard architecture," *IEE Proceedings*, vol. 137, no. 4, pp. 221–225, 1990.

[34] S. Chen, G. Gibson, C. Cowan, and P. Grant, "Reconstruction of binary signals using an adaptive radial-basis-function equalizer," *Signal Processing 22*, pp. 77–93, 1991.

[35] G. J. Gibson, S. Siu, and C. F. N. Cowan, "The application of nonlinear structures to the reconstruction of binary signals," *IEEE Transactions on Signal Processing*, vol. 39, pp. 1877–1884, Aug. 1991.

[36] S. Chen and B. Mulgrew, "Overcoming co-channel interference using an adaptive radial basis function equaliser," *Signal Processing*, vol. 28, pp. 91–107, 1992.

[37] S. Chen, P. M. Grant, and C. F. Cowan, "Orthogonal least-squares algorithm for training multioutput radial basis function networks," *IEE Proceedings-F*, vol. 139, pp. 378–384, Dec. 1992.

[38] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive bayesian decision feedback equaliser based on a radial basis function network," *ICC*, pp. 343.3.1 – 343.3.5, 1992.

[39] M. E. Austin, "Decision-feedback equalization for digital communication over dispersive channels," tech. rep. 437, MIT Lincoln Lab., Lexington, MA, Aug. 1967.

[40] W. R. Kirkland and D. P. Taylor, "The application of feed forward neural networks to channel equalization," in *IJCNN*, pp. II 919–924, June 1992.

[41] W. Rummler, R. Coutts, and M. Liniger, "Multipath fading channel models for microwave digital radio," *IEEE Communications Mag.*, vol. 24, pp. 30–42, Nov. 1986.

[42] N. W. K. Lo and H. M. Hafez, "Neural network channel equalization," in *IJCNN*, pp. II 981–986, June 1992.

[43] S. Arcens, J. Cid-Sueiro, and A. R. Figueiras-Vidal, "Pao networks for data transmission equalization," *IJCNN*, pp. II–963 – II–968, June 1992.

[44] T. Kohonen, K. Raivio, O. Simula, O. Venta, and J. Henriksson, "Combining linear equalization and self-organizing adaptation in dynamic discrete-signal detection," in *IJCNN 90*, pp. 223–228, 1990.

[45] D. D. Falconer, "Adaptive equalization of channel nonlinearities in QAM data transmission systems," *Bell Syst. Tech. J.*, vol. 57, pp. 2589–2611, Sept. 1978.

[46] J. F. Michaelides and P. Kabal, "Nonlinear adaptive filtering for echo cancellation," in *ICC'88*, (IEEE Service Cent Piscataway, NJ, USA), pp. 951–956, IEEE, 1988.

[47] E. Biglieri, A. Gersho, R. D. Gitlin, and T. L. Lim, "Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission," *IEEE J. Selected Areas Commun.*, vol. SAC-2, pp. 765–777, Sept. 1984.

[48] M. Bellafemina and S. Benedetto, "Identification and equalization of nonlinear channels for digital transmission," in *1985 International Symposium on Circuits and Systems - Proceedings*, (IEEE Service Center, Piscataway, NJ, USA), pp. 1477–1480, IEEE, 1985.

[49] Q. Zhang, "Adaptive equalization using the backpropagation algorithm," *IEEE Transactions on Circuits and Systems*, vol. 37, pp. 848–849, June 1990.

[50] N. Benvenuto, M. Marchesi, F. Piazza, and A. Uncini, "Non linear satellite radio links equalized using blind neural networks," in *ICASSP 91*, pp. 1521–1524, IEEE, 1991.

[51] K. Konstantinides and K. Yao, "Modelling and computationally efficient time domain linear equalisation of nonlinear bandlimited QPSK satellite channels," *IEE Proceedings, Part I: Communications, Speech and Vision*, vol. 137, pp. 438–442, Dec. 1990.

[52] G. Karam and H. Sari, "Analysis of predistortion, equalization, and ISI cancellation techniques in digital radio systems with nonlinear transmit amplifiers," *IEEE Transactions on Communications*, vol. 37, pp. 1245–1253, Dec. 1989.

[53] A. Baranyi, "Nonlinear equalizer for controlling AM-PM conversion and AM compression," in *1988 IEEE International Symposium on Circuits and Systems, Proceedings*, (IEEE Service Cent, Piscataway, NJ, USA), pp. 2545–2548, 1988.

[54] A. Said, A. Elrefaie, and L. Kurz, "Equalizaion for QPSK signals over nonlinear channels," in *IEEE/IEICE Global Telecommunications Conference 1987*, (IEEE Service Cent, Piscataway, NJ, USA), p. 2153, 1987.

[55] H. Sari and G. Karam, "Cancellation of power amplifier nonlinearities in digital radio receivers," in *ICC'87*, (IEEE Service Cent Piscataway, NJ, USA), pp. 1809–1814, 1987.

[56] A. P. Kalcina, D. J. Skellern, and S. Howe, "Equalisation of a TDMA QPSK satellite channel," *Journal of Electrical and Electronics Engineering, Australia*, vol. 7, pp. 98–103, June 1987.

[57] S. Pupolin and L. J. Greenstein, "Performance analysis of digital radio links with nonlinear transmit amplifiers," *IEEE Journal on Selected Areas in Communications*, vol. SAC-5, pp. 533–545, Apr. 1987.

[58] K. S. Shanmugan and M. J. Ruggles, "An adaptive linearizer for 16-QAM transmission over non-linear satellite channels," in *GLOBECOM'86*, (IEEE Service Center, Piscataway, NJ, USA), pp. 126–132, IEEE, 1986.

[59] K. Konstantinides and K. Yao, "Modeling and equalization of nonlinear bandlimited satellite channels," in *ICC'86*, (IEEE Service Center, Piscataway, NJ, USA), pp. 1622–2626, IEEE, 1986.

[60] G. Junyent, R. Agusti, and F. Casadevall, "Nonlinear equalization in optical fiber at 140 and 560 mbit/sec," in *Canadian Communications & Energy Conference*, (IEEE Service Center, Piscataway, NJ, USA), pp. 324–327, IEEE, Oct. 1982.

[61] N. L. Swenson, B. L. Shoop, and J. M. Cioffi, "Precluding nonlinear ISI in direct detection long-haul fiber optic systems," *IEEE Photonics Technology Letters*, vol. 3, pp. 182–184, Feb. 1991.

[62] J. H. Winters and R. D. Gitlin, "Electrical signal processing techniques in long-haul, fiber-optic systems," in *Supercomm ICC '90*, (IEEE Service Center, Piscataway, NJ, USA), pp. 397–403, IEEE, 1990.

[63] J. H. Winters and R. D. Gitlin, "Electrical signal processing techniques in long-haul fiber-optic systems," *IEEE Transactions on Communications*, vol. 38, pp. 1439–1453, Sept. 1990.

[64] J. L. Grubb, "The traveler's dream come true," *IEEE Communication Magazine*, vol. 29, pp. 48–51, Nov. 1991.

[65] IEEE, "Satellite and terrestrial systems and services for travelers," *IEEE Communications Magazine*, vol. 29, Nov. 1991.

[66] M. L. M. J. H. Lodge and S. N. Crozier, "A comparison of data modulation techniques for land mobile satellite channels," *IEEE Trans. Veh. Techno.*, vol. VT-36, pp. 28–35, Feb. 1987.

[67] J. H. Lodge, "Mobile satellite communications systems: Toward global personal communications," *IEEE Comm. Mag.*, vol. 29, pp. 24–31, Nov. 1991.

[68] A. D. Kucar, "Mobile radio: An overview," *IEEE Comm. Mag.*, vol. 29, pp. 72–85, Nov. 1991.

[69] R. Steel, "Towards a high-capacity digital cellular mobile radio system," *IEE Proceedings, Part F*, vol. 132, pp. 405–415, Aug. 1985.

[70] W. LeBlanc, S. Hənna, and S. Mahmoud, "Performance of a low complexity celp speech coder under mobile channel fading conditions," *39 th IEEE Vehicular Technology Conference*, p. 647, May 1989.

[71] S. Gupta, r. Viswanathan, and R. Muammar, "Land mobile radio systems-a tutorial exposition," *IEEE Communications Mag.*, vol. 23, pp. 34–45, June 1985.

[72] A. Clark, "Digital modems for land mobile radio," *IEE Proc. F, Commun., Radar & Signal Processing*, vol. 132, pp. 348–362, Aug. 1985.

[73] J. D. Oetting, "A comparison of modulation techniques for digital radio," *IEEE Trans. on Com.*, vol. COM-27, pp. 1752–1762, Dec. 1979.

[74] J. G. D. Forney, R. G. Gallager, G. R. Lang, F. M. Longstaff, and S. U. Qurehi, "Efficient modulation for band-limited channels," *IEEE Journ. on Selected Areas in Communications*, vol. SAC-2, pp. 632–646, Sept. 1984.

[75] C.-L. Liu and K. Feher, "Performance of non-coherent pi/4-QPSK in a frequency-selective fast raleigh fading channel," *ICC'90*, pp. 1369–1373, 1990.

[76] C.-L. Liu and D. K. Feher, "Noncoherent detection of pi/4-QPSK systems in a CCI-AWGN combined interference environment," *39 th IEEE Vehicular Technology Conference*, pp. 83–93, May 1989.

[77] S. Sampei and T. Sunaga, "Rayleigh fading compensation method for 16-QAM in digital land mobile radio channels," *39 th IEEE Vehicular Technology Conference*, pp. 640–646, May 1989.

[78] J. K. Cavers, "An analysis of pilot symbol assisted 16QAM for digital mobile communications," in *IEEE Vehicular Technology Conference*, pp. 380–385, 1991.

[79] W. T. Webb and R. Steele, "Equaliser techniques for QAM transmissions over dispersive mobile radio channels," *IEE Proceedings*, vol. 138, pp. 566–576, Dec. 91.

[80] L. H. W.T. Webb and R. Steele, "Bandwidth efficient QAM schemes for Rayleigh fading channels," *IEE Proceedings-I*, vol. 138, pp. 169–174, June 91.

[81] S. Stein, "Fading channel issues in system engineering," *IEEE Journal on Selected Areas in Com.*, vol. SAC-5, pp. 68–88, Feb. 87.

[82] W. C. Dam and D. P. Taylor, "An adaptive maximum likelihood receiver for Rayleigh fading channels," Tech. Rep. CRL Report 224, CRL, McMaster University, Nov. 90.

[83] C. Loo and N. Secord, "Computer models for fading channels with applications to digital transmission," *IEEE Trans. on Vehicular Technology*, vol. 40, pp. 700–707, Nov. 91.

[84] W. H. C. G. A. Arredondo and E. H. Walker, "A multipath fading simulator for mobile radio," *IEEE Trans. Commun.*, vol. COM-21, pp. 1325–1328, Nov. 73.

[85] K. E. M. E. L. Caples and T. R. Minor, "A uhf channel simulator for digital mobile radio," *IEEE Trans. Veh. Technol.*, vol. VT-29, pp. 281–289, May 80.

[86] J. D. Ralphs and F. M. E. Sladen, "An hf channel simulator using a new Rayleigh fading method," *Radio Electron. Eng.*, vol. 46, pp. 579–587, Dec. 76.

[87] J. I. Smith, "A computer generated multipath fading simulation for mobile radio," *IEEE Trans. Veh. Technol.*, vol. VT-24, pp. 39–40, Aug. 75.

[88] E. Lutz and E. Plochinger, "Generating rice processes with given spectral properties," *IEEE Trans. on Veh. Technol.*, vol. VT-34, pp. 178–181, Nov. 85.

[89] P. J. McLane, P. H. Wittke, and P. K.-M. Ho, "A study on combined channel coding and modulation for mobile satellite communications," *DSS Contract 0ST85-00197 Final Report*, Mar. 86.

[90] C. Loo, "A statistical model for a land mobile satellite link," *ICC'84*, pp. 588–594, 1984.

[91] C. Loo, "Measurements and models of a mobile-satellite link with applications," *Proc. GLOBECOM'85*, pp. 1177–1180, Dec. 1985.

[92] C. Loo, "Measurements and models of a mobile-satellite link with applications to msk signals," *IEEE Trans. on Veh. Techno.*, vol. VT-35, pp. 114–121, Aug. 1988.

[93] G.D'Aria, L. Stola, and V. Zingarelli, "Modelling and simulation of the propagation characteristics of the 900 mhz narrowband-tdma cept/gsm mobile radio," *39 th IEEE Vehicular Technology Conference*, pp. 631–639, May 1989.

[94] M. Lecours and F. Marceau, "Design and implementation of a channel simulator for wideband mobile radio transmission," *39 th IEEE Vehicular Technology Conference*, pp. 652–655, May 1989.

[95] J. C.-I. Chuang, "The effects of delay spread on 2-PSK, 4-PSK, 8-PSK and 16-QAM in a portable radio environment," *IEEE Tran. Veh. Techno.*, vol. 38, pp. 43–45, May 1989.

[96] J. C.-I. Chuang, "The effects of time delay spread on portable radio communications channels with digital modulation," *IEEE Journal on Selected Areas in Communications*, vol. SAC-5, pp. 879–889, June 1987.

[97] S. Gurunathan and K. Feher, "Fade compensation in QPRS mobile radio modems," in *IEEE Vehicular Technology Conference*, pp. 144–148, 1991.

[98] P. J. McLane, P. H. Wittke, P. K. M. Ho, and C. Loo, "PSK and DPSK trellis codes for fast fading, shadowed mobile satellite communication channels," *IEEE Trans. Comm.*, pp. 1242–1245, Nov. 88.

[99] D. Divsalar and M. K. Simon, "The design of trellis coded MPSK for fading channels: performance criteria," *IEEE Trans. Com.*, vol. 36, pp. 1004–1012, Sept. 1988.

[100] D. Divsalar and M. K. Simon, "The design of trellis coded MPSK for fading channels: set partitioning for optimum code design," *IEEE Trans. Com.*, vol. 36, pp. 1013–1021, Sept. 1988.

[101] C. Schlegel and J. D. J. Costello, "Bandwidth efficient coding for fading channels: code construction and performance analysis," *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 1356–1368, Dec. 1989.

[102] Y.-J. Liu and E. Biglieri, "Fully transparent phase/doppler invariant trellis-coded modulation," *IEE Proceedings-I*, vol. 138, pp. 105–116, Apr. 1991.

[103] K. Y. Chan, A. Bateman, and J. P. McGeehan, "Trellis-coded reference-based MPSK in Rayleigh fading," in *IEEE GLOBECOM'92*, pp. 102–105, 1992.

[104] J. K. Cavers, "An analysis of pilot symbol assisted modulation for Rayleigh fading channels," *IEEE Trans. on Vehicular Technology*, vol. 40, pp. 686–693, Nov. 91.

[105] A. Clark and S. Jayasinghe, "Channel estimation for land mobile radio systems," *IEE Proc. F, Commun., Radar & Signal Processing*, vol. 134, pp. 383–393, July 1987.

[106] D. D. F. N. W. K. Lo and A. U. H. Sheikh, "Adaptive equalization and diversity combining for mobile radio using interpolated channel estimates," *IEEE Trans. on Vehicular Technology*, vol. 40, pp. 636–645, Aug. 91.

[107] S. Crozier, D. Falconer, and S. Mahmoud, "Short-block equalization techniques employing channel estimation for fading time-dispersive channels," *39 th IEEE Vehicular Technology Conference*, pp. 142–146, May 1989.

[108] J. F. Doherty, "Equalization for mobile communications channels," *40 th IEEE Vehicular Technology Conference*, pp. 643–647, 1990.

[109] S. M. Elnoubi, "BER improvement of PRCPM in mobile radio channels using decision feedback equalization," *40 th IEEE Vehicular Technology Conference*, pp. 516–520, 1990.

[110] H. B. S. M. Elnoubi and E.-S. A. Youssef, "BER improvement of PRCPM mobile radio channels with discriminator detection using decision feedback equalization," *IEEE Trans. Veh. Techno.*, vol. 40, pp. 516–520, Nov. 1991.

[111] T. Nakai, S. Ono, Y. Shimazaki, and N. Kondoh, "Adaptive equalizer for digital cellular radio," in *IEEE Vehicular Technology Conference*, pp. 13–16, 1991.

[112] K. E. a. W. S. Samejima, "Differential PSK system with nonredundant error correction," *IEE Journal On Selected Areas In Communications*, vol. SAC-1, pp. 74–81, Jan. 83.

[113] Y. M. T. Masamura, S. Samejima and H. Fuketa, "Differential detection of msk with nonredundant error correction," *IEEE Trans. Comm.*, vol. COM-27, pp. 912–918, June 1979.

[114] D. Wong and P. Mathiopoulos, "Nonredundant error correction analysis and evaluation of differentially detected pi/4-shift DQPSK systems in a combined cci and awgn environment," *IEEE Trans. on Vehicular Technology*, vol. 41, pp. 34–47, Feb. 1992.

[115] P. Ho and D. Fung, "Error performance of multiple-symbol differential detection of PSK signals transmitted over correlated Rayleigh fading channels," *IEEE Trans. Com.*, vol. 40, pp. 1566–1569, Oct. 1992.

[116] P. Ho and D. K. P. Fung, "Error performance of interleaved trellis coded PSK modulations in correlated Rayleigh fading channels," *IEEE Trans. Com.*, vol. 40, pp. 1800–1809, Dec. 1992.

[117] D. Divsalar and M. K. Simon, "Multiple symbol differential detection of MPSK," *IEEE Trans. Com*, vol. 38, pp. 300–308, Mar. 1990.

[118] D. Divsalar and M. K. Simon, "Multiple symbol differential detection of uncoded and trellis coded MPSK," *Tech. Rep. JPL 89-98*, Nov. 89.

[119] J. F. B. S. G. Wilson and C. Marshal, "Multiple-symbol detection of M-DPSK," *Proc. IEEE GLOBECOM'89*, pp. 1692–1697, Nov. 1989.

[120] J. H. Lodge and M. L. Moher, "Maximum likelihood sequence estimation of cpm signals transmitted over Rayleigh flat-fading channels," *Trans. Com.*, vol. 38, pp. 787–794, June 1990.

[121] D. Fung, "Error performance of PSK signals transmitted over correlated Rayleigh fading channels," Master's thesis, Simon Fraser University, July 91.

[122] P. Bune, "Effective low-effort adaptive equalizers for digital mobilephone systems," *39 th IEEE Vehicular Technology Conference*, pp. 147–154, May 1989.

[123] R. Haeb and H. Meyr, "A systematic approach to carrier recovery and detection of digitally phase modulated signals on fading channels," *IEEE Trans. on Communications*, vol. COM-37, pp. 748–754, July 1989.

[124] A. Clark and F. McVerry, "Channel estimation for an hf radio link," *IEE Proc. F, Commun., Radar & Signal Processing*, vol. 128, pp. 33–42, Feb. 1981.

[125] A. Clark, S. N. Abdullah, S. Jayasinghe, and K. Sung, "Pseudobinary and pseudoquaternary detection processes for linearly distorted multilevel qam signals," *IEEE Trans. on Communications*, vol. COM-33, pp. 639–645, July 1985.

[126] A. Clark and C. Soon, "Pseudobinary detection processes for a 9600 bit/s modem," *IEE Proc. F, Commun., Radar & Signal Processing*, vol. 129, pp. 305–314, Oct. 1982.

[127] D. Makrakis and P. Mathiopoulos, "Performance evaluation of a fading estimation/cancellation receiver for trellis coded cpm signals," *39 th IEEE Vehicular Technology Conference*, pp. 101–105, May 1989.

[128] M. J. Barrett, "Error probability for optimal and suboptimal quadratic receivers in rapid Rayleigh fading channels," *IEEE Journal on Selected Areas in Communications*, vol. SAC-5, pp. 302–304, Feb. 87.

[129] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, madaline and backpropagation," *Proceedings of the IEEE*, vol. 78, pp. 1415–1441, Sept. 1990.

[130] B. Widrow and R. Winter, "Neural nets for adaptive filtering and adaptive pattern recognition," *Computer*, vol. 21, pp. 25–39, Mar. 1988.

[131] P. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* PhD thesis, Harvard University, Cambridge, Aug. 1974.

[132] D. E. Rummelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," ics report 8506, Institute for Cognitive Science, University of California at San Diego, La Jolla, CA, Sept. 1985.

[133] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, *Parallel Distributed Processing*, vol. 1. MIT Press, 1986.

[134] R. Lippman, "An introduction to computing with neural nets," *IEEE Acoust. Speech Signal Process. Mag.*, vol. 4, 1987.

[135] D. R. Hush and B. G. Horne, "Progress in supervised neural networks," *IEEE Signal Processing Mag.*, Jan. 1993.

[136] P. Hartmann and D. Taylor, "Telecommunications by microwave digital radio," *IEEE Communications Mag.*, vol. 24, pp. 11–16, Aug. 1986.

[137] T. Noguchi, Y. Daido, and J. Nossek, "Modulation techniques for microwave digital radio," *IEEE Communications Mag.*, vol. 24, pp. 21–30, Oct. 1986.

[138] S. Haykin, *Communication Systems.* New York: John Wiley & Sons, second edition ed., 1983.

[139] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of Communication Systems.* New York: Plenum Press, 1992.

[140] N. Amitay and L. Greenstein, "Multipath outage performance of digital radio receivers using finite-tap adaptive equalizers," *IEEE Trans. Comm.*, vol. COM-32, pp. 597–608, Aug. 1984.

[141] W. Rummler, "A new selective fading model: Application to propagation data," *Bell System Technical Journal*, vol. 58, pp. 1037–1071, May/June 1979.

[142] J. K. Kruschke and J. R. Movellan, "Benefits of gain: Speeded learning and minimal hidden layers in back-propagation networks," *IEEE Transactions on Systems, Man and Cybernetics*, pp. 273–280, 1991.

[143] G. M. Georgiou and C. Koutsougeras, "Complex domain backpropagation," *IEEE Trans. on Circuits and Systems*, vol. 39, pp. 330–334, may 1992.

[144] J. Hopfield, "Neurons with graded response have collective properties like those of two state neurons," *Proc. Natl. Acad. of Science, USA*, vol. 81, 1984.

[145] J. J. Shynk, "Performance surfaces of a single layer perceptron," *IEEE Transactions on Neural Networks*, vol. 1, Sept. 1990.

[146] J. dong Yuh and R. W. Newcomb, "Circuits for multi-level neuron nonlinearities," *IJCNN*, pp. 27–32, June 1992.

[147] D. J. Amit, *Modelling Brain Function.* Cambridge University Press, 1989.

[148] J. Si and A. N. Michel, "Analysis and synthesis of discrete-time neural networks with multi-level threshold-functions," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 1461–1464, June 1991.

[149] W. Banzhaf, "A network of multistate units capable of associative memory and pattern classification," *Physica D*, pp. 418–426, Mar. 1989.

[150] D. P. Taylor, *Nonlinear Feedback Equalization of Digital Signals Transmitted Over Dispersive Channels.* PhD thesis, McMaster Univ., May 1972. Appendix A.

[151] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets part i: Introduction," *IEEE Comm. Mag.*, vol. 25, pp. 12–21, Feb. 1987.

[152] H. L. V. Trees, *Detection, Estimation, and Modulation Theory.* John Wiley & Sons, 1968. Part I.

[153] M. D. Richard and R. P. Lipmann, "Neural network classifiers estimate bayesian a posteriori probabilities," *Neural Computation*, vol. 3, pp. 461–483, 1991.

[154] F. Kanaya and S. Miyake, "Bayes statistical behavior and valid generalization of pattern classifying neural networks," *IEEE Trans. on Neural Networks*, vol. 2, pp. 471–475, July 1991.

[155] E. Barnard, "Comments on "bayes statistical behavior and valid generalization of pattern classifying neural networks"," *IEEE Trans. on Neural Networks*, vol. 3, pp. 1026–1027, Nov. 1992.

[156] F. Kanaya and S. Miyake, "Author's reply," *IEEE Trans. on Neural Networks*, vol. 3, pp. 1027–1028, Nov. 1992.

[157] L. J. Greenstein and M. Shafi, "Outage calculation methods for microwave digital radio," *IEEE Communications Mag.*, vol. 25, pp. 30–38, Feb. 1987.

[158] S. Bellini and F. Rocca, "Blind deconvolution: Polyspectra or bussgang techniques?," in *Digital Communications*, pp. 251–263, Elsevier Science Publishers B. V. (North-Holland), 1986.

[159] Bussgang, "Tech, rep. 216," tech. rep., M.I.T. R.L.E., 1952.

[160] J. dong Yuh and R. W. Newcomb, "A multilevel neural network for A/D conversion," *IEEE Trans. Neural Networks*, vol. 4, pp. 470–483, May 1993.