MULTI-DIMENSIONAL LATTICE EQUALISER FOR Q^{2}PSK

by

JACQUES ÉTIENNE CILLIERS

A dissertation submitted as partial fulfilment
of the requirements for the degree

Master of Engineering (Digital Communication)

in

**ELECTRICAL, ELECTRONIC
and COMPUTER ENGINEERING**

in the

FACULTY OF ENGINEERING

UNIVERSITY OF PRETORIA

SUPERVISOR: Prof. L.P. Linde

February 2002

Key Words:

Phase-Shift Keying, Mobile Digital Communication, Multidimensional Modulation, Synchronisation, Recursive Least Squares, Lattice Algorithms, Adaptive Equalisation, Decision Feedback Equalisation, Mobile Wireless Channel, Multipath, Fading Channels.

SUMMARY

The aim of this dissertation was the design, implementation and performance evaluation of a Recursive Least Squares (RLS), lattice based, adaptive, multidimensional, decision feedback equaliser (DFE) for the spectrally efficient four-dimensional digital modulation technique, referred to as Quadrature-Quadrature Phase-Shift Keying, Q^2PSK. Q^2PSK constitutes a relatively new modulation technique, and the application of adaptive equalisation to this technique has not yet been considered in the open literature.

This dissertation represents an in depth study into the Q^2PSK modulation technique, as well as the optimal implementation, in simulation, of such a modem to aid the inclusion of the adaptive lattice DFE, for application to high speed mobile digital communication over the V/UHF channel in the presence of multipath propagation. Specific aspects of synchronization applicable to this modem platform are also addressed. An in depth study was also conducted into the realisation of a V/UHF channel simulation, capable of producing a Ricean and/or Rayleigh fading multipath propagation environment for the evaluation of the modem platform and adaptive equaliser structure. The theoretical analysis of the effect of multipath on a Q^2PSK signal led to the correct design of the adaptive lattice structure, as well as the correct interfacing of the equaliser to the receiver platform.

The performance of the proposed synchronisation strategies, in tandem with the equalisation technique were evaluated for several static, as well as fading multipath channels. The simulation results obtained show the equaliser operates correctly, and can give large performance gains over the static matched filter (matched to the transmitted waveform) implementation of the modem platform. Several simulations were specifically designed to highlight the performance limitations of the adaptive equalisation technique.

OPSOMMING

Die doel van hierdie verhandeling was die ontwerp, implementering en evaluering van die werkverrigting van 'n Rekursiewe Kleinste Kwadraat (RKK), latwerkgebaseerde, aanpasbare, multidimensionele, beslissingsterugvoereffenaar (BTE) vir toepassing in die spektrale-effektiewe, vierdimensionele, digitale modulasietegniek, wat bekend staan as Kwadratuur-Kwadratuur Faseskuifsleuteling, K^2FSL. Hierdie is 'n betreklik nuwe modulasietegniek, en dus is aanpasbare effenaars vir hierdie tegniek nog nie ondersoek in die literatuur nie.

Hierdie verhandeling is 'n dieptestudie van die K^2FSL modulasietegniek, asook die optimale implementering, in simulasie, van so 'n modem om die integrering van die aanpasbare effenaar vir hoë datatempo mobiele digitale kommunikasie oor die B/UHF kanaal, in die aanwesigheid van multipad radiogolfvoortplantingseffekte te toets. Die sinkroniseringsaspekte van hierdie modem word ook spesifiek aangespreek. 'n Dieptestudie is ook geloods om 'n B/UHF kanaalsimulator daar te stel, wat deining én multipad kan simuleer om die modem en aanpasbare effenaarstruktuur deeglik te evalueer. 'n Teoretiese afleiding van die effek van multipad op die K^2FSL sein het gelei tot die korrekte ontwerp van die aanpasbare latwerkstruktuur, asook die korrekte koppelvlak tussen die modemplatform en die effenaar.

Die werkverrigting van die voorgestelde sinkronisasietegnieke is getoets in samewerking met die vereffeningstegniek vir verskeie statiese asook deinende multipad kanale. Die resultate toon dat die effenaar korrek werk en dat 'n groot verbetering in die modem se werkverrigting bewerkstellig kan word oor die geval waar die ontvanger aangepas is tot die sender golfvorm. Die simulasies illustreer ook die beperkings van die aanpasbare effenaar baie duidelik.

ACKNOWLEDGEMENTS

I would like to thank my Creator and Saviour, to Him all the praise !

I am indebted to many people for their advice and assistance towards the successful completion of this dissertation.

In particular to my supervisor, Professor Louis Linde for his encouragement, guidance, and impeccable proof reading of this manuscript, not to mention the constant supply of the latest publications. Special thanks to both Professor Linde, and Professor van Schalkwyk for helping me to obtain finance for the first two years of this study.

To my friends and colleagues, many thanks for putting up with me while I was working on this project.

A special word of gratitude is expressed to Danie van Wyk, for many hours of discussion, and acting as a bouncing board, concerning Q^{2}PSK, "dimensionality", simulation techniques, and digital communication in general. The library of publications loaned to me was also extremely helpful, as well as the help with some of the finer nuances of Latex.

To Frans Marx, for a four year long fight over SNR, E_b/N_0 and the correct noise calibration of simulations. If either of us had given up we would never have got to the bottom of this ! The many hours of discussion and help, in this and other fields, were invaluable.

To Emlyn Davies for many hours of mathematical discussion, philosophical pondering and programming help.

Many thanks to Momin Jamil for many late night discussions, over tea, pertaining to signal processing, and some of the in depth mathematical and philosophical aspects thereof.

To Brian Burmeister and Danie van Wyk for allowing me to run simulations in the "background" on their computers.

To my parents, for their unselfish support, both emotionally, and financially, as well as for putting up with the eccentricities of raising an engineer. Without their solid grounding, and financing of my undergraduate studies, this dissertation would not have been possible.

To my wife's parents, Mr. and Mrs. Westhof for their support and encouragement.

Finally, I would also express my deepest gratitude to my wife, Therese for her continuous love, and for sacrificing many hours of our time to this study. Special thanks for typing many of the equations in this dissertation, thus allowing me to focus on the technical problems at hand.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Over the last 10 years, spanning 1990 to 2000, we have seen an unprecedented and unequalled growth in the mobile communication market. This is especially true of the cellular telephone concept which has overwhelmed the world to such an extent that it is now taken to be the norm, and few people can imagine how they ever existed without such a device.

The concept of mobile communication is being extended as more and more services are being offered to the end user. This, in combination with the explosion of the "internet" over the same period of time, will be responsible for the following renaissance in the freeing of humans from fixed connections in offices and homes. This will be heralded by the advent of "tetherless" computing, which will enable humans to make use of features such as video conferencing, thus totally changing the way in which interpersonal communication occurs.

The astounding increase in the use of mobile communications, and the advent of "tetherless" computing will both make astronomical demands on the underlying media over which the data for these applications has to be transmitted. This media is the invisible realm in which electromagnetic waves propagate and is referred to as the electromagnetic (EM) spectrum. The use of this spectrum for mobile applications is dependent on the frequency of the underlying sinusoidal waves which can be transmitted. Such transmissions become viable at around 10MHz, and have an upper limit of about 250GHz which is just before the transition from microwave radiation to infrared radiation. The electromagnetic spectrum which is available for such wireless applications is thus a limited resource. Companies in the wireless communications field are charged prices in millions of dollars for the use of frequency bands in the EM spectrum, by the national, state owned, regulatory bodies which govern and police the use of this spectrum. This shows that the EM spectrum is a very scarce natural resource which has to be utilized in an optimum way. The fact that recently, in Australia, a set of 15MHz channels in the 1800MHz band were sold for A$284 million, emphasises the scarcity of the EM spectrum. The inventor of the cellular phone, Martin Cooper, has a slightly different outlook on this point :

"*There is no lack of spectrum, only a lack of spectral efficiency.*"

The communication engineer is responsible for designing techniques for the transmission of end user data at a maximum rate, while minimizing the bandwidth occupied by the signal. These two requirements are contradictory as an increase in data rate generally implies a linear increase in bandwidth for a fixed modulation format. This has the simple implication that modulation techniques have to be found which make more efficient use of the EM spectrum.

The other problem which the communication engineer has to face is the fact that the radio wave propagation environment for mobile communications is so hostile that it is nearly impossible to imagine that reliable communication is possible over this media.

This dissertation will address the second problem for one of the more recent, and more efficient, digital modulation techniques, known as quadrature-quadrature phase shift keying (Q^2PSK). This technique makes use of a higher dimensional signalling space over the traditional two dimensional signalling space. This implies a more complex structure, so the computational complexity of the system has increased for an increase in the bandwidth efficiency of the modulation format.

Trading computational complexity for bandwidth efficiency is actually a very good trade as bandwidth is becoming more expensive, and computing power is becoming cheaper thanks to Moore's Law:

"*Transistor density on integrated circuits doubles every 18 months*"

This implies that techniques which we perceive to be too complex to implement in a realtime system at present, will be possible at some future date.

In contrast to the cost of implementation, one should also consider the cost of a single decibel (dB) of improvement over the operational life span of a system. This is demonstrated by the following quotation W.P. Robins [1], which gives a slightly exaggerated picture :

"*The economic importance of a small degradation in link performance is well illustrated, admittedly in a very crude fasion, by considering an international satellite communications system. The total cost of such a system over a decade including the cost of the space segment and of all ground stations together with staffing costs might well exceed £300 million. A 3dB degradation in performance would halve the traffic capacity. One way to restore the originally required traffic capacity, although admittedly not the most economic, would be to duplicate the complete system at a cost of a further £300 million. Thus, in a very crude sense indeed, we might say that in such a system 3dB is equivalent to £300 million and hence 0.1 dB is very roughly equivalenty to £10 million. This rather absurd example is given merely to emphasise that in such a system a small signal degradation is expensive and that therefore relatively complex steps to reduce link degradation are economically justified.*"

From the above discussion it can be seen that *bandwidth* and *power* are the primary communication resources available to the designer. The cost of bandwidth as well as the cost of system performance are extremely high, and ever increasing. On the other hand, hardware is becoming cheaper, and performance levels are increasing at a rapid rate. It is thus worthwhile developing complex modulation techniques which are bandwidth efficient, and complex algorithms to increase system gain.

1.1 PROBLEM STATEMENT

Multipath propagation over communication channels, especially in a wireless mobile communication environment, can be the most hostile effect encoumtered, degrading the received signal to such an extent that reliable retrieval of the transmitted data stream can be impossible. The main aim of this dissertation is the development of a lattice filter based, multidimensional, adaptive equaliser for the four dimensional modulation format referred to as quadrature-quadrature phase shift keying (Q^{2}PSK). The equaliser is necessary to mitigate the dispersive effects caused by the channel over which communication takes place. To achieve this goal, an accurate multidimensional modem simulation platform for the Q^{2}PSK modulation technique has to be developed, and the multidimensional lattice equaliser has to be properly integrated with the modem receiver section. Finally, to ascertain the performance of the adaptive equaliser in conjunction with the modem platform in realistic conditions, an accurate simulation of the fading mobile channel model also has to be developed.

1.2 LITERATURE SURVEY

This section contains a summary of relevant literature to the above problem statement, as well as a generalised system model for the problem at hand.

1.2.1 System Model

The generalised system model of a digital communication system is depicted in Figure 1.1 below, and gives perspective to the following section which covers the relevant literature.

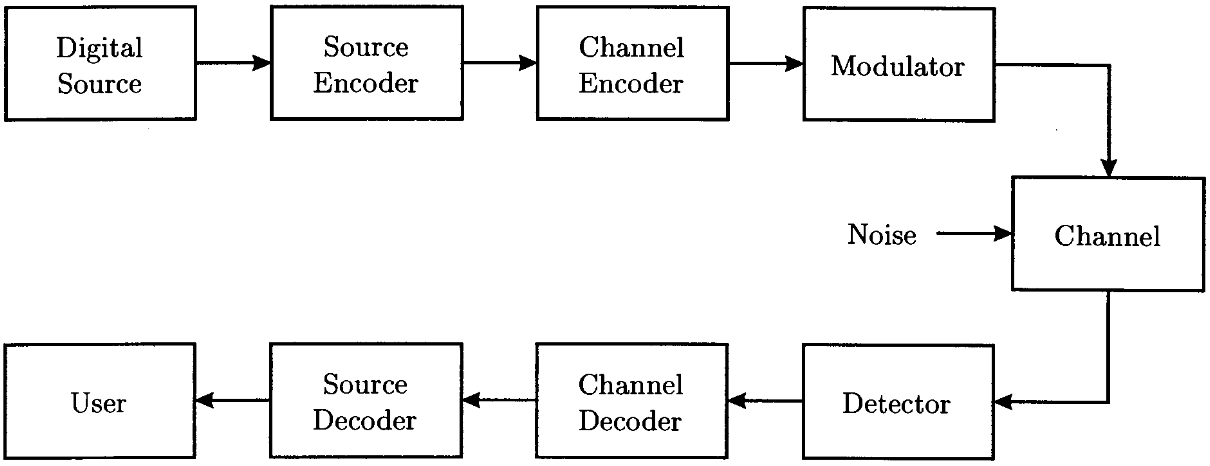

Figure 1.1: General digital communication system.

For the duration of this study, only the modulator, channel and detector will be addressed. It is assumed that the data input into the modulator consists of a stream of equiprobable ones and zeros representing the binary data. The modulator is responsible for the conversion of this stream into an analogue signal, and the transposition of this signal to a radio frequency (RF) carrier.

Three channel models will be used in this study, the additive noise channel, the linear filter channel and the linear time variant filter channel. The noise will always be additive white Gaussian noise (AWGN).

The detector is the beginning of the receiver train, and is responsible for translating the received RF signal to baseband, demodulating it, and maintaining synchronisation with the transmitter. The adaptive equaliser is the final processing step in the detector before the data is converted back to a binary stream, and passed on to the channel decoder.

An overview of publications and other literature pertaining to various aspects of the above mentioned system model will be presented in the following sections.

1.2.2 Multi-Dimensional Modulation Techniques

Shannon [2] proposed that the signalling waveforms be chosen on the basis of the coordinates of M points located on the surface of a sphere of dimension n, and analysed such a technique in [3]. The maximisation of the minimum distance between these points was considered in [4, 5, 6]. One of the first hardware implementation of a modem making use of multidemensional waveforms was published by Ulstad [7] who gave examples of three and five dimensional constellations. This implementation could be scaled to the megahertz range of the spectrum. Multidimensional signal constellations for voiceband channels were considered in [8].

The concept of a simple four-dimensional transmission technique was pioneered by Saha [9, 10] and Birdsall [11] who named their technique quadrature-quadrature phase shift keying (Q^{2}PSK). This concept was optimised for bandlimited channels by Visintin and Bieglieri [12] and extended to a differential version in [13, 14]. A digital implementation of Q^{2}PSK is considered in [15, 16] and new pulse shapes are proposed by Wei in [17]. Two variants of the technique which fall between QPSK and Q^{2}PSK are given in [18, 19]. Korn and Wei as well as Hughes give a more generalised approach as well as performance bounds in [20, 21, 22, 23].

Coded Q^{2}PSK is discussed by van Wyk and Acha in [15, 24, 25] and by Saha in [26]. Synchronization techniques for Q^{2}PSK are discussed by the originators of the technique, as well as several other authors such as De Gaudenzi and Luise [27, 28]. The application of Kalman based carrier recovery techniques has been considered by van Wyk, Cilliers and Linde in [29, 30]. Linde and van Wyk have also considered frame synchronisation techniques for Q^{2}PSK in [31]. The maximum-likelihood reception of 4-D ISI corrupted signals has been considered by Feiz and Soliman [32].

1.2.3 Channel Modelling

As channel modelling was not the main purpose of this dissertation, standard reference books were used during the development of the mobile channel simulation [33, 34, 35, 36, 37].

General multipath channel characterisation is discussed by Sklar in [38, 39] and Siller in [40]. Specific aspects of mobile channel modelling, and mobile channel simulation are discussed in [41, 42, 43, 44, 45, 46, 47, 48, 49].

1.2.4 Adaptive Filters and Equalisation

The first adaptive equaliser was proposed by Lucky [50, 51] in 1965. This equaliser was designed for voice telephone channels, and was based on an analogue hardware implementation of the Least Mean Squares (LMS) algorithm. The complex version of the LMS algorithm was introduced in 1975 by Widrow et. al. [52]. The performance of this algorithm was again analised by Ungerboeck [53], and recently several variable step versions of the algorithm have been proposed [54, 55, 56, 57].

Due to the relatively slow initial convergence, and steady state tracking capability of the LMS algorithm, faster algorithms were sought. A faster converging algorithm is obtained if a recursive least squares (RLS) criterion for the adjustment of the equaliser coefficients is employed. Algorithms of this type are collectively referred to as RLS algorithms. The Kalman algorithm was the first RLS algorithm proposed for updating the filter coefficients of adaptive equalizers by Godard in 1974 [58]. The stability and implementation of this algorithm have been addressed in [59, 60]. An IIR filter form for the Kalman algorithm was presented by Mulgrew and Cowan [61]. The Kalman algorithm makes use of direct inversion of an estimated covariance matrix, and is thus prone to instability, especially if implemented in limited precision arithmetic, and operating in noisy and/or unstationary conditions. The Kalman algorithm is also one of the most complex algorithms [62] in terms of computational complexity as a function of the number of coefficients, N. The Kalman algorithms computational complexity is proportional to N^2. Various so-called *fast RLS algorithms* have been developed which have a computational complexity proportional to N. The first such algorithm was proposed for adaptive equalisation by Falconer and Ljung [63]. These algorithms are also discussed in [64, 65], and the complex version of the algorithm was derived by Alexander [66].

An equaliser based on the RLS criterion can also be implemented as a lattice structure. The convergence rate for the linear lattice filter is identical to that of the RLS based finite impulse response (FIR) adaptive filter. The computational complexity of the RLS lattice algorithm also proportional to N, but it possesses a slightly higher complexity than the fast RLS algorithms. The adaptive lattice was first introduced by Morf and Lee [67, 68, 69, 70] in 1977 for system identification, and was proposed for adaptive equalisation applications by Satorius and Alexander in 1979 [71, 72]. Even though the adaptive lattice algorithm is slightly more complex than the fast Kalman algorithm, it has certain key advantages:

- less sensitive to quantization errors and roundoff noise [73]
- dynamic adjustment of filter length
- inherently stable due to absence of direct autocorrelation matrix inversion.

A square-root RLS algorithm has also been derived by Lee, Morf and Friedlander in 1981 [74, 75], and it has been shown to have excellent numerical properties, such as being very robust to roundoff noise in the computations [76, 77, 78]. Various other forms of RLS Lattice filters have been described in [79, 80, 81].

The multichannel lattice algorithm, for equal length input vectors was introduced in 1981 by Mueller [82]. This algorithm was generalized by Ling and Proakis [83] to account for non-equal input vector lengths. The same authors then applied this algorithm to adaptive decision feedback equalization of 2-D signals [84, 85, 62, 33]. The implementation considerations for this algorithm have been considered in [86, 87, 88].

1.3 CONTRIBUTION

It should be noted that this is the first time that adaptive filters have been applied to achieve the equalisation of Q^{2}PSK. The only other techniques which has been applied to this problem is maximum likelihood sequence estimation published by Feiz and Soliman [32]. The modified Viterbi algorithm (MVA) was used to implement the maximum likelihood sequence estimator. Unfortunately these authors only derive the theoretical performance of the algorithm for non-fading channels.

The original contributions of this dissertation are listed below in prioritised order

- Simulation results showing the effectiveness of the proposed equaliser structure for the mitigation of multipath propagation in the Q^{2}PSK platform.
- Development of the correct Lattice filter structure for the implementation of a decision feedback equaliser (DFE) for multidimensional digital modulation techniques, and the implementation thereof in software.
- Mathematical analysis of the effect of multipath on a Q^{2}PSK signal, leading to equaliser design.
- Development of an accurate, extensible multidimensional modem simulation platform.
- Interfacing of modem to the equaliser.
- Implementation and testing of a Recursive Least Squares (RLS) based Digital Phase-locked loop (DPLL) for carrier recovery in the Q^{2}PSK platform.
- Development of an accurate mobile channel simulator.
- Development of new Doppler filter design equations, which were shown to be valid over bandwidths spanning five orders of magnitude.
- Identification of an algorithmic instability in the fast filter startup procedure described by Kay in [89].
- Development of a closed form expression for the sum of the squared impulse response of a nth order infinite impulse response (IIR) filter.

This dissertation does not cover the various aspects of source coding, as well as channel coding/error correction strategies. It is assumed that the source has been encoded in an optimal way to produce a digital data stream consisting of equiprobable zeros and ones. Error correction strategies for Q^{2}PSK have been discussed in detail by D. van Wyk in [15], and are thus not addressed in this dissertation. All the techniques proposed in [15] can be added externally to the current system and will only have an impact on the data rate offered to the user, if the bandwidth is kept constant.

1.3.1 Publications

The following papers have been published during research for this dissertation :

- J.E Cilliers and L.P. Linde,"Fractionally spaced lattice equaliser algorithm employing a progressively interlaced joint process error approach", *COMSIG '95*, pp 23-26, November 1995.
- J.E Cilliers and L.P. Linde,"Comparison of data directed estimators and decision feedback lattice equaliser algorithms' performance on fading channels", *AFRICON '96*, pp 1083-1086, September, 1996.
- B. Westra, D.J. van Wyk, J.E. Cilliers and L.P. Linde, "Performance evaluation of multi-level four-dimensional Q^{2}PSK in Gaussian noise", *COMSIG '97*, pp 141-146, 1997.
- D.J. van Wyk, J.E. Cilliers and L.P. Linde "Performance of an adaptive carrier Kalman estimator for Q^{2}PSK", *IWTS '97*, pp 282-287, 1997.
- J.E. Cilliers, L.P. Linde and D.J. van Wyk, "A synchronous Q^{2}PSK DS-CDMA system : system conceptualisation, implementation and performance analysis", *IEEE ISSTA '98*, pp 4-8, September 1998.
- M. Jamil, L.P. Linde, J.E. Cilliers and D.J. van Wyk, "Comparison of complex spreading sequences based on filtering methods and mean square correlation characteristics", *Transactions of the SAIEE*, Vol. 89, No. 3, pp 98-112, November 1998.
- J.E. Cilliers and L.P. Linde, "Application and performance of a dual-loop adaptive RLS Kalman estimator for carrier recovery in a Q^{2}PSK system", *AFRICON '99*, pp 153-158, September 1999.
- J.E. Cilliers and L.P Linde, "Performance of an adaptive phase-locked loop applied to decision directed carrier recovery in a Q^{2}PSK system", *International Federation of Automatic Control (IFAC) Conference on Technology Transfer in Developing Countries*, pp 218-223, July 2000.

1.4 DISSERTATION OUTLINE

The introduction to the topic of study, as well as the general background, and relevant publications in the open literature pertaining to the topic are presented in this chapter. The rest of the dissertation is summarized on a per chapter basis as follows:

Chapter 2 : Theoretical introduction to multidimensional digital communication. This chapter introduces multidimensional digital communication from a theoretical point of view based on a discussion of the available dimensionality of a specified bandwidth. Q^{2}PSK is a 4-D digital modulation technique which is compared in terms of spectral efficiency and bit error probability to existing modulation techniques. The effect of intersymbol interference (ISI) on the Q^{2}PSK signal is derived, and higher dimensional Q^{2}PSK formats are reviewed.

Chapter 3 : Mobile wireless channel characteristics and modelling. This chapter is a survey of relevant topics pertaining to the statistical modelling of the wireless mobile channel, followed by the design and implementation of a software channel simulator. The simulator is specifically designed to give an accurate as possible rendition of the multipath fading environment.

Chapter 4 : Generic multidimensional digital communication simulation platform, describes the implementation of a Q^2PSK digital communication system in simulation. The simulation is designed to be extensible to higher dimensionality than the 4-D system used for this study. The transmitter and receiver are presented, as well as methods of synchronising the receiver to the transmitter to form a coherent system. Key results are given to demonstrate the operation of the system.

Chapter 5 : Channel equalisation for Q^2PSK. This chapter introduces the concept of equalisation from a theoretical discussion of distortionless transmission. The necessity for adaptive equalisation is discussed, and the multidimensional lattice equaliser is proposed as a solution to the adaptive equalisation of Q^2PSK. The interfacing between the receiver structure and the equaliser, as well as the modes of operation of the equaliser are discussed.

Chapter 6 : Results, presents bit error probability results, as a function of bit energy to noise ratio, to demonstrate the following :

- Correct operation of simulation platform in AWGN.
- Performance of the adaptive carrier recovery technique.
- Performance of theadaptive equaliser in static channels.
- Performance of the adaptive equaliser in line of sight (LOS) fading channels as well as frequency selective fading channels.

Chapter 7 : Conclusion. This chapter presents a summary of what has been achieved in this study relative to the initial goals set out in this chapter. A summary of topics for future research, which have been identified during this study, is also presented.

CHAPTER 2

THEORETICAL INTRODUCTION TO MULTIDIMENSIONAL DIGITAL COMMUNICATION

2.1 INTRODUCTION

The purpose of a radio digital communication system is to transmit information in a digital format between two points which may be geographically separated. To achieve this, use is made of a limited portion of the electromagnetic spectrum. As this spectrum is a limited natural resource, it is very expensive. For example the analogue mobile phone bands in Australia were closed down on 31 December 1999. These frequencies in the 800MHz band were sold by means of on online auction for A$350 million. A set of 15MHz channels in the 1800 MHz band were sold for A$284 million. The designer must thus optimise the digital communication system to be as spectrally efficient as possible while still delivering an acceptable level of data integrity, which is application dependant.

The physical constraints on the rate at which data can be transmitted in a given bandwidth in the presence of additive white Gaussian noise (AWGN), for a power constrained transmitter are discussed in the first half of this chapter. The second half of this chapter contains a discussion on the modulation technique to be used in the rest of this study, namely Q^{2}PSK.

2.2 SIGNAL DIMENSIONALITY

This section contains an introduction to the term "signal dimensionality" and the implications thereof. The following summary is based on the more detailed discussion in [90] of sampling and discrete time representation of signals. The principles of this theory are necessary to understand why multidimensional signalling is possible.

2.2.1 Sampling and Discrete Time Representation

Digital signal processing is the art of using digital computers or dedicated hardware to change certain characteristics of signals to give new signals, from which information can be extracted. The process of signal processing on a digital computer is achieved by representing a continuous signal as a discrete, and even finite, set of numbers, and then performing arithmetical operations upon these numbers which are equivalent to various forms of processing on the signal itself. A signal, $g(t)$ which is continuous in time and value, can be expressed as a series in known functions, as

$$g(t) = \sum_n a_n \phi_n(t) \tag{2.1}$$

where the $\phi_n(t)$ are given functions of time and the a_n are scalars which are characteristic of the particular signal $g(t)$. Since the $\phi_n(t)$ are known, the set of a_n distinguish the representation of one signal from another. Desirable properties of (2.1) are

- The a_n and ϕ_n are obtained in a simple way.
- The number of terms to get a 'good' approximation of $g(t)$ should be small enough for economical processing.

Two criterion can be used to measure how well $g(t)$ is represented

- convergence in the mean sense implies

  $$\lim_{N\to\infty} \left| g(t) - \sum_{n=-N}^{N} a_n \phi_n(t) \right|^2 = 0 \quad \text{, almost everywhere} \tag{2.2}$$

  for deterministic signals.

- convergence in the least squares sense is described by

  $$\lim_{N\to\infty} \int_{-\infty}^{\infty} \left| g(t) - \sum_{n=-N}^{N} a_n \phi_n(t) \right|^2 dt = 0 \tag{2.3}$$

  The integral in this expression is the energy in the error for a given N.

These two criteria are limiting criteria, but the goodness of the fit for a finite N is of practical interest to the signal representation problem. A simple method for the determination of the a_n is important. One of the simplest methods to obtain the coefficients of a series expansion is by sampling, so the accuracy of this representation has to be addressed, which in turn leads to the lowpass sampling expansion:

Theorem 2.1 *If the Fourier transform $G(f)$ of a function $g(t)$ is zero outside the frequency interval $(-W, W)$, then $g(t)$ is uniquely determined by the sequence of values $g(n/2W)$, spaced $1/2W$ apart in time.*

Formally: Let $G(f)$ be the Fourier transform of a function $g(t)$ such that

$$G(f) = 0, |f| > W \tag{2.4}$$

Then,

$$\begin{aligned} g(t) &= \sum_{n=-\infty}^{\infty} g(n/2W)\frac{\sin(\pi(2Wt-n))}{\pi(2Wt-n)} \\ &= \sum_{n=-\infty}^{\infty} g(n/2W)\,\mathrm{sinc}\,(2Wt-n) \qquad (2.5) \\ & \qquad (2.6) \end{aligned}$$

If a finite duration of $g(t)$, say the interval $(0,T)$, is considered, it may be approximated by a finite number of terms of (2.5)

$$g(t) \approx \sum_{n=0}^{2TW-1} g(n/2W)\,\mathrm{sinc}\,(2Wt-n) \tag{2.7}$$

This is the basis for the statement that a signal which is approximately duration and bandwidth limited, can be represented by a set of $2WT$ sample values.

2.2.2 Signal Dimensionality

To formally define the signal dimensionality it is necessary to consider the error involved in the approximation of the previous section.

Definition 2.1 *A signal $g(t)$ has dimension M at relative level η in the interval $(-T/2, T/2)$ if:*

1. *There is a collection of $M = M(t,\eta)$ signals, say $\phi_1(t)$,$\phi_2(t)$,$\dots$,$\phi_M(t)$, such that $g(t)$ is essentially indistinguishable at relative level η from a sum of the form*

$$\sum_{i=1}^{M} a_i\phi_i(t) \tag{2.8}$$

That is, a set of a_i's is required such that

$$\int_{-T/2}^{T/2} \left| g(t) - \sum_{i=1}^{M} a_i\phi_i(t) \right|^2 dt \leq \eta \int_{-T/2}^{T/2} |g(t)|^2 dt \tag{2.9}$$

2. *There is no set of $M-1$ functions whose linear combinations can furnish a signal essentially indistinguishable at relative level η from $g(t)$.*

Definition 2.2 *A set of G signals has dimension M at relative level η in the interval $(-T/2, T/2)$ if:*

1. *There is a collection of $M = M(t,\eta)$ signals, say $\phi_1(t)$,$\phi_2(t)$,$\ldots$,$\phi_M(t)$, such that every signal in G is essentially indistinguishable at relative level η from a sum of the form*

$$\sum_{i=1}^{M} a_i\phi_i(t) \tag{2.10}$$

That is, a set of a_i's are required for each $g(t)$ contained in G such that

$$\int_{-T/2}^{T/2} \left| g(t) - \sum_{i=1}^{M} a_i\phi_i(t) \right|^2 dt \leq \eta \int_{-T/2}^{T/2} |g(t)|^2 dt \tag{2.11}$$

2. *There is no set of $M - 1$ functions whose linear combinations can yield signals essentially indistinguishable at relative level η from every member of G.*

When considering duration limited functions the sinc() function does not give the best fit. The best fit for a given number of terms is obtained by an expansion in functions known as prolate spheroidal wave functions (PSWF). A complete discussion of these functions is given by Slepian, Landau and Pollak in [91, 92], and Papoulis in [93].

The use of the sinc() functions for $\phi_i(t)$ will result in a slightly larger value of M than will the use of the prolate spheroidal wave functions [90]. Nevertheless, as the method for obtaining the coefficients for the sinc() functions is simple, use is made of the sinc() functions. The sinc() functions are defined as

$$\phi_i(t) = \mathrm{sinc}\,(2Wt - i) \tag{2.12}$$

Accordingly, dimensionality is defined as :

Definition 2.3 *A signal $g(t)$ has dimension M at relative level η in the interval $(0, T)$ if there is a collection of $M = M(T,\eta)$ sinc() functions, $\mathrm{sinc}\,(2Wt - i)$, $i = 0,\ldots, M-1$, such that $g(t)$ is essentially indistinguishable at relative level η from a sum of the form*

$$\sum_{i=0}^{M-1} g(i/2W)\mathrm{sinc}\,(2Wt - i) \tag{2.13}$$

where W is the essential (lowpass) bandwidth at relative level η. That is, it is required that

$$\int_0^T \left| g(t) - \sum_{i=0}^{M-1} g(i/2W)\mathrm{sinc}\,(2Wt - i) \right|^2 dt \leq \eta \int_0^T |g(t)|^2 dt \tag{2.14}$$

This discussion only covers lowpass signals, but it can be extended to bandpass signals if use is made of the complex envelope representation. The dimensionality of the bandpass signal is then defined as the dimensionality of its complex envelope, where W is then defined as the one sided passband bandwidth occupancy.

In summary, any signal approximately bandwidth limited to W Hz, and approximately duration limited to T seconds can be accurately represented by $M = 2WT$ samples.

2.3 CHANNEL CAPACITY

In this section some of the most fundamental limits on the transmission of digital data over a bandwidth limited channel are presented.

The first limitation is that of limited power. Any radio frequency transmitter is designed to deliver a specified average power, and a specified peak power while causing minimal distortion to the signal being transmitted. The power limitation is thus a hardware limitation. The second limitation is on the maximum rate at which the data can be transmitted, without the transmission exceeding a specified bandwidth. The third limitation is on the correctness of the received data, which is dependent on the noise power at the receiver.

Shannon's channel capacity theorem states that

Theorem 2.2 *The capacity of a channel of bandwidth B Hertz, perturbed by additive white Gaussian noise of power spectral density $N_0/2$ and limited in bandwidth to B hertz, is given by*

$$C = B\log_2\left(1 + \frac{P}{N_0 B}\right)\,[bits/s] \tag{2.15}$$

where P is the average transmitted power.

This result highlights the three key system parameters which constrain the operational performance of all digital communication systems in a single expression. These three parameters are :

1. Channel bandwidth (B).
2. Average transmitted/received signal power (P).
3. Noise power spectral density at the channel output $(N_0/2)$.

The theorem also defines the fundamental limit on the rate of error free transmission for a power-limited, band-limited Gaussian channel.

Given an ideal system that transmits at a bit rate R_b, equal to the channel capacity C, the average transmitted/received power is

$$P = E_b C \tag{2.16}$$

where E_b is the transmitted energy per bit. Equation (2.15) can now be written as

$$\frac{C}{B} = \log_2\left(1 + \frac{E_b}{N_0}\frac{C}{B}\right) \tag{2.17}$$

$$\frac{R_b}{B} = \log_2\left(1 + \frac{E_b}{N_0}\frac{R_b}{B}\right) \tag{2.18}$$

where R_b/B is the bandwidth effeciency, ρ, in $bits/sec/Hz$. Equivalently the energy-per-bit to noise power spectral density ratio, E_b/N_0, can be expressed in terms of the bandwidth effeciency for the ideal system as

$$\frac{E_b}{N_0} = \frac{2^{C/B} - 1}{C/B} \tag{2.19}$$

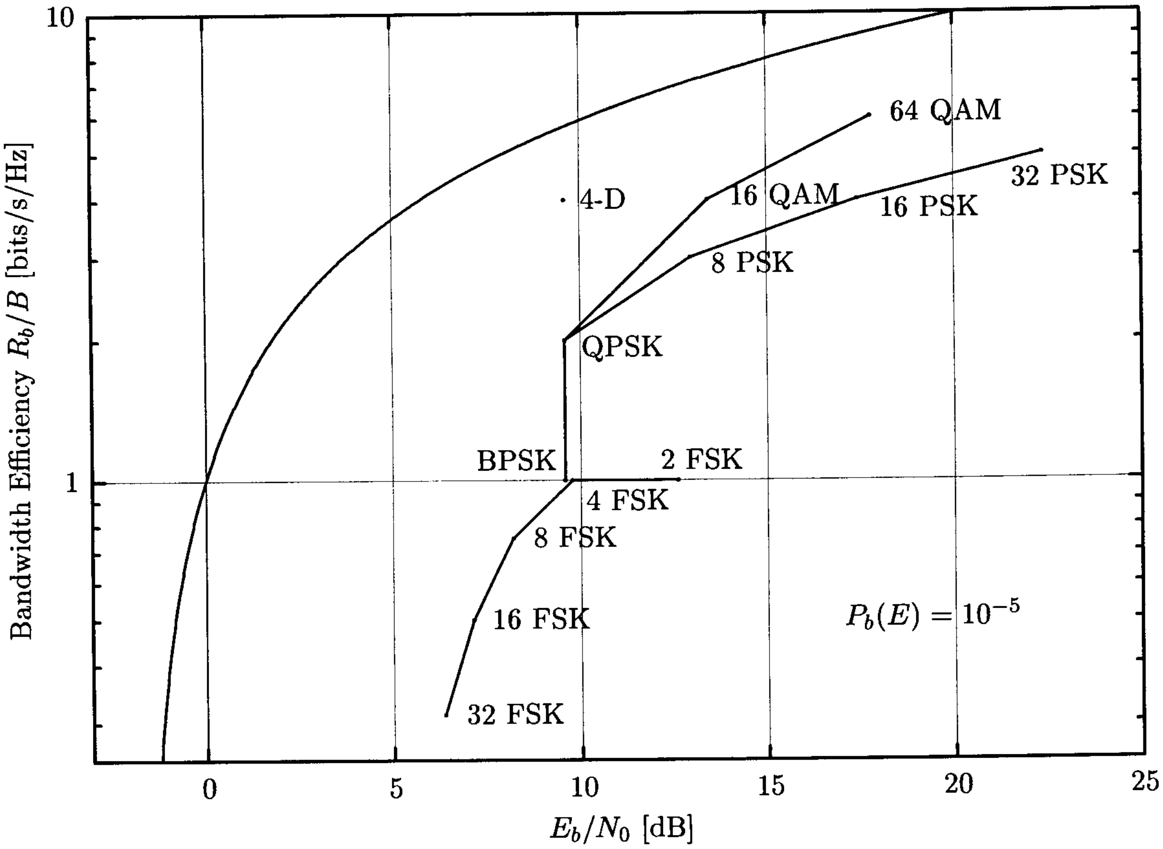

Figure 2.1: Channel Capacity and Various Modulation Formats.

which defines the capacity boundary that separates error free combinations of system parameters from those for which error free transmission is impossible ($R_b > C$). This boundary is depicted as the solid line in Figure 2.1.

The graph gives a more intuitive description of the trade-offs which can be made in the design of a digital communication system between E_b/N_0, R_b/B and P_e. In summary horizontal movement trades P_e versus E_b/N_0 for a fixed R_b/B, and vertical movement trades P_e versus R_b/B for a fixed E_b/N_0.

This graph is an extension of a similar graph (Figure 4.2.29) in [33], where the bandwidth has been defined as the one sided bandwidth occupancy of the signal, and use has been made of bit error probability, instead of symbol error probability. It should be noted that use is made of the zero-to-zero bandwidth occupancy of the signal in Haykin [94].

2.3.1 Comparison of Modulation Techniques

Now that the fundamental limits on signal representation and the capacity of a bandlimited channel in the presence of AWGN have been defined, a closer study can be made of the BPSK and QPSK modulation formats. For this comparison it is first necessary to define a signalling interval, T_s, which is the time used to transmit one symbol, where a symbol consists of a grouping of n bits giving $M = 2^n$ combinations. Each one of these is then assigned to a signal point in the signal constellation. Bandwidth is defined as the one sided baseband bandwidth occupancy necessary to include the first zero in the spectrum of the signal. This bandwidth is chosen to be $B = 1/T_s$. The input bit rate is denoted by $R_b = 1/T_b$, where T_b is the bit duration, which is allowed to vary from one modulation technique to the next. The bandwidth efficiencies of BPSK and QPSK will be compared in terms of these definitions.

In the case of BPSK, $T_s = T_b$ so $R_b = 1/T_s$ ($n = 1$,$M = 2$), and the one sided bandwidth occupancy is given by $B = 1/T_s$, giving a bandwidth efficiency of :

$$\begin{aligned} \rho_{BPSK} &= \frac{R_b}{B} \\ &= 1.0 \qquad [bits/sec/Hz] \end{aligned} \tag{2.20}$$

In the case of QPSK two bits are transmitted per signalling interval ($n = 2$,$M = 4$) so $T_s = 2T_b$ giving $R_b = 2/T_s$, and the one sided bandwidth occupancy is $B = 1/T_s$. The bandwidth efficiency is thus given by :

$$\begin{aligned} \rho_{QPSK} &= \frac{R_b}{B} \\ &= 2.0 \qquad [bits/sec/Hz] \end{aligned} \tag{2.21}$$

It can be seen from Figure 2.1 that by making use of QPSK twice the amount of data can be transmitted as compared to BPSK without incurring a loss in bit error probability (BEP) or equivalently needing more energy per bit. (It should be noted that the BEP of Grey coded QPSK is always better than that of BPSK when expressed in terms of E_b/N_0.) This gain was obtained by increasing the dimensionality of the signal constellation from a 1-dimensional to a 2-D space by making use of orthogonal carriers. The question which now arises is wether it is possible to obtain a further increase in bandwidth efficiency by expanding the signal space to a 3 or higher dimensional signal space, while keeping the bandwidth fixed. This question can be answered by calculation of the maximum number of dimensions which can be accommodated in the bandwidth occupied by BPSK and QPSK. As use must be made of the bandwidth of the complex envelope, $W = 2B = 2/T_s$, and the dimensionality of this bandwidth is given by

$$\begin{aligned} N &= 2WT && (2.22) \\ &= 2\frac{2}{T_s}T_s && \\ N &= 4 && (2.23) \end{aligned}$$

The possibility thus exists to once again double the dimensionality of the signal space, and thus double the bandwidth efficiency of QPSK, so

$$\rho_{NEW} = \frac{R_b}{B} = 4 \qquad [bits/sec/Hz] \tag{2.24}$$

giving

$$T_s = 4T_b \tag{2.25}$$

This implies that input data rate is double that of QPSK, in the same bandwidth.

A relatively new modulation technique, called Q^{2}PSK which makes use of a 4-D signal space to achieve a higher spectral efficiency than QPSK without a degradation in E_b/N_0, is presented in the next section.

2.4 Q^{2}PSK SIGNALLING

This section contains a theoretical description of the generation of a four dimensional signal space and the use thereof to realise the Q^{2}PSK modulation format. The performance of Q^{2}PSK will also be evaluated in terms of spectral properties and bit error probability. A discussion on the effect of bandlimiting is also included, as well as the generalisation of Q^{2}PSK to higher dimensions.

2.4.1 The origins/history of Q^{2}PSK

Q^{2}PSK modulation makes use of a four dimensional signal space instead of the more conventional two dimensional signal space for the transmission of digital information. This modulation technique has its origins in satellite communications, where use was made of two spatially orthogonal electric fields to double the spectral efficiency of standard two dimensional modulation techniques. This method was know as "frequency-reuse" and was first reported by Wilson and Sleeper [95]. The block diagram in Figure 2.2 shows the underlying principle of this method.

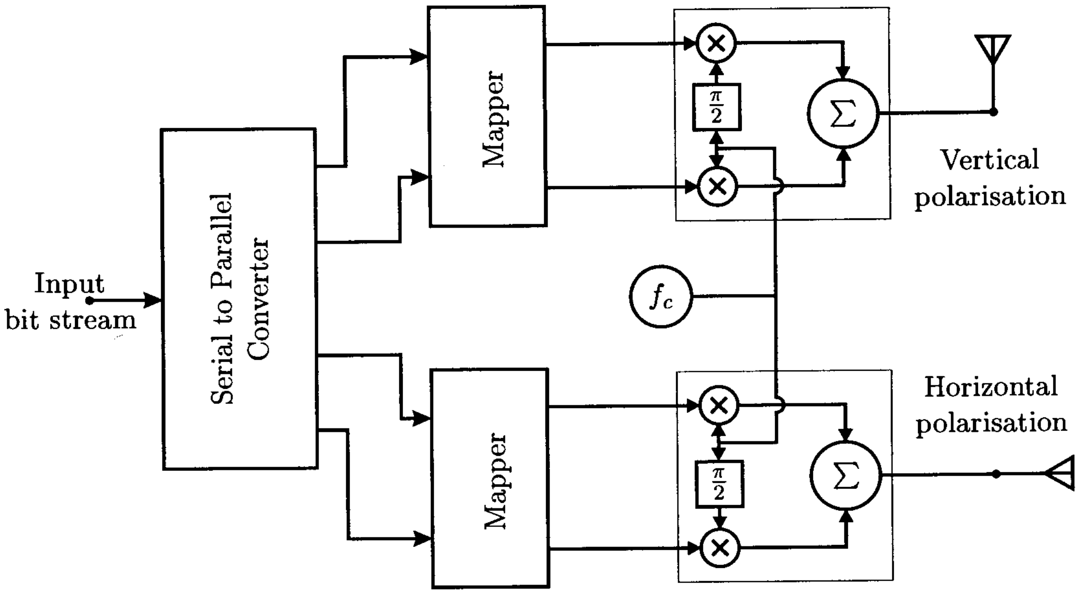

Figure 2.2: Transmitter structure for "frequency- reuse" method.

Q^{2}PSK modulation employs two MSK-like signals, which are modulated in quadrature onto the subcarrier, thus creating a 4-D signal constellation. This scheme has a theoretical spectral efficiency of $4.0 bits/s/Hz$ as compared to MSK ($1.33 bits/s/Hz$) and QPSK ($\alpha = 0.35$) ($2.66 bits/s/Hz$). This scheme thus makes more efficient use of the electromagnetic spectrum, which is a limited natural resource.

2.4.2 Theoretical description of Q^{2}PSK

The Q^{2}PSK modulation technique makes use of a 4-D signal constellation and can thus be described by four mutually orthogonal basis functions as follows

$$
\begin{aligned}
\phi_1(t) &= \frac{2}{\sqrt{T_s}}\cos(2\pi f_d t)\cos(2\pi f_c t) \qquad -\frac{T_s}{2} \le t \le \frac{T_s}{2} && (2.26)\\
\phi_2(t) &= \frac{2}{\sqrt{T_s}}\sin(2\pi f_d t)\cos(2\pi f_c t) \qquad -\frac{T_s}{2} \le t \le \frac{T_s}{2} && (2.27)\\
\phi_3(t) &= \frac{2}{\sqrt{T_s}}\cos(2\pi f_d t)\sin(2\pi f_c t) \qquad -\frac{T_s}{2} \le t \le \frac{T_s}{2} && (2.28)\\
\phi_4(t) &= \frac{2}{\sqrt{T_s}}\sin(2\pi f_d t)\sin(2\pi f_c t) \qquad -\frac{T_s}{2} \le t \le \frac{T_s}{2} && (2.29)\\
\phi_i(t) &= 0 \qquad |t| > \frac{T_s}{2} && (2.30)
\end{aligned}
$$

where $T_s = 1/R_s$ is the symbol period, f_d is the deviation frequency and f_c is the carrier frequency. These four time functions form a set $\phi_i(t)$ of equal-energy orthonormal signals under the restriction that

$$f_c = \frac{n}{2T_s} \qquad n = \text{integer} \ge 2 \tag{2.31}$$

which ensures orthogonality of the carriers. The orthogonality of $\phi_i(t)$ remains invariant under translations of the time origin by multiples of T_s. One bipolar bit is modulated onto each of these basis functions for each symbol period, thus giving $T_s = 4T_b$. The first orthogonality (between ϕ_1,ϕ_3, and ϕ_2,ϕ_4) is derived from the orthogonality of the carriers. The two data streams which are modulated onto each of the carriers (between ϕ_1 and ϕ_3, ϕ_2 and ϕ_4) are made orthogonal to each other by means of orthogonal pulse shaping. From this discussion two quadrature pulse shaping functions can be identified :

$$
\begin{aligned}
p_1(t) &= \cos(\pi t/T_s) \qquad -\frac{T_s}{2} \le t \le \frac{T_s}{2} && (2.32)\\
&= 0 \qquad \text{otherwise} && (2.33)
\end{aligned}
$$

$$
\begin{aligned}
p_2(t) &= \sin(\pi t/T_s) \qquad -\frac{T_s}{2} \le t \le \frac{T_s}{2} && (2.34)\\
&= 0 \qquad \text{otherwise} && (2.35)
\end{aligned}
$$

These two pulse shapes are plotted in Figure 2.3

The two data shaping pulses, $p_1(t)$ and $p_2(t)$ are the shortest possible segments of a cosinusoidal and sinusoidal waveform which have the narrowest frequency content, but still maintain mutual orthogonality. This can be seen by noting that the cross correlation coefficient between a sinusoidal and cosinusoidal pulse of arbitrary length and phase is given by :

$$
\begin{aligned}
\rho &= \int_{t_1}^{t_2} \cos(2\pi f_d t + \theta)\sin(2\pi f_d t + \theta)dt && (2.36)\\
&= \frac{\sin^2(2\pi f_d t_2 + \theta)}{4/pi f_d} - \frac{\sin^2(2\pi f_d t_1 + \theta)}{4/pi f_d} && (2.37)
\end{aligned}
$$

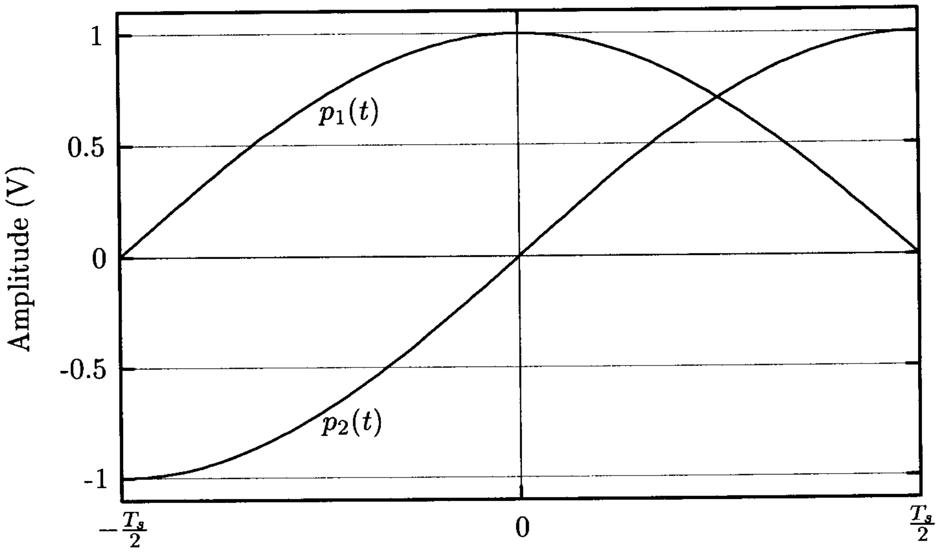

Figure 2.3: Data shaping pulses.

where t_1 and t_2 denote the start and end times of the pulse respectively. This expression is now solved for t_2 giving

$$\begin{aligned} t_2 &= t_1 \pm \frac{n}{2f_d} \qquad n = \text{integer} && (2.38)\\ &= t_1 \pm nT_s && (2.39)\end{aligned}$$

This means that the two shaping pulses remain orthogonal for any phase shift, as long as they are at least T_s long. By setting $\theta = \pi/4$, two shaping pulses which have the same basic waveform result. The one pulse is just a time reversed replica of the other pulse. These two pulses are plotted in Figure 2.4, and are described by

$$\begin{aligned} p_1(t) &= \cos(\pi t/T_s + \pi/4) \qquad -\frac{T_s}{2} \le t \le \frac{T_s}{2} && (2.40)\\ &= 0 \qquad \text{otherwise} && (2.41)\end{aligned}$$

$$\begin{aligned} p_2(t) &= \sin(\pi t/T_s + \pi/4) \qquad -\frac{T_s}{2} \le t \le \frac{T_s}{2} && (2.42)\\ &= 0 \qquad \text{otherwise} && (2.43)\end{aligned}$$

These pulses were first proposed by Westra, Cilliers, van Wyk and Linde [96], and have the advantage that the amplitude discontinuities at intersymbol switching instants are of the same amplitude for both pulse trains. A similar pulse shaping pair, which makes use of complex values pulse shapes has been proposed by Wei [17]. Pulse shapes for 8 dimensional systems have been presented in [97] and [21]. The discussion of these pulse shapes will be postponed until the end of this section.

Q^{2}PSK Modulation

In the previous section the theoretical principles behind the generation of a four dimensional signal space by means of four mutually orthogonal basis functions ($\phi_i(t)$) were described. In this section the mapping of a data stream onto this signal space will be discussed.

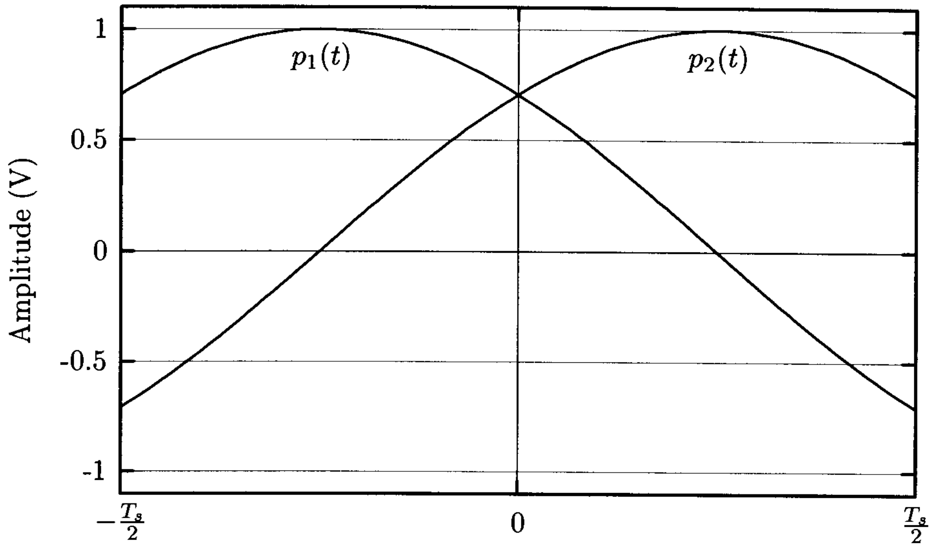

Figure 2.4: Modified data shaping pulses.

Data from an i.i.d. (independant, identically distributed) binary (±1) data source at a rate of $4/T_s$ is demultiplexed into four streams $a_i(t)$, where the amplitude of the basic rectangular pulse is ±1, and its duration is T_s. Each of these data streams is multiplied by an associated basis function, thus giving the following passband signal [9, 11, 10]

$$\begin{aligned}
s_{Q^2PSK}(t) &= \frac{\sqrt{T_s}}{2}\sum_{i=1}^{4} a_i(t)\phi_i(t) && (2.44)\\
&= a_1(t)p_1(t)\cos(2\pi f_c t) + a_2(t)p_2(t)\cos(2\pi f_c t) + \\
&\quad + a_3(t)p_1(t)\sin(2\pi f_c t) + a_4(t)p_2(t)\sin(2\pi f_c t) && (2.45)\\
&= [a_1(t)p_1(t) + a_2(t)p_2(t)]\cos(2\pi f_c t) + \\
&\quad + [a_3(t)p_1(t) + a_4(t)p_2(t)]\sin(2\pi f_c t) && (2.46)\\
&= a_1(t)\cos\left(\frac{\pi t}{T_s}\right)\cos(2\pi f_c t) + a_2(t)\sin\left(\frac{\pi t}{T_s}\right)\cos(2\pi f_c t) + \\
&\quad + a_3(t)\cos\left(\frac{\pi t}{T_s}\right)\sin(2\pi f_c t) + a_4(t)\sin\left(\frac{\pi t}{T_s}\right)\sin(2\pi f_c t)\\
&= \cos\left[2\pi\left(f_c + \frac{b_{14}(t)}{2T_s}\right)t + \phi_{14}(t)\right] + \\
&\quad + \sin\left[2\pi\left(f_c + \frac{b_{23}(t)}{2T_s}\right)t + \phi_{23}(t)\right] && (2.47)
\end{aligned}$$

where

$$b_{14}(t) = -a_1(t)a_4(t) \qquad (2.48)$$
$$b_{23}(t) = +a_1(t)a_4(t) \qquad (2.49)$$

and

$$\phi_{14}(t) = 0 \text{ or } \pi \quad \text{as } a_1(t) = +1 \text{ or } -1 \text{ respectively} \qquad (2.50)$$
$$\phi_{23}(t) = 0 \text{ or } \pi \quad \text{as } a_3(t) = +1 \text{ or } -1 \text{ respectively} \qquad (2.51)$$

where $\sqrt{T_s}/2$ is a a power normalisation factor. The above expression has its origins in [10]. An equivalent expression from [15] is also given for reference

$$\begin{aligned} s_{Q^2PSK}(t) &= [a_1(t)p_1(t) + a_2(t)p_2(t)]\cos(2\pi f_c t) + \\ &\quad + [a_3(t)p_1(t) + a_3(t)p_2(t)]\sin(2\pi f_c t) \qquad (2.52) \\ &= a_c(t)\cos\left[(2\pi f_c + b_c(t)2\pi f_d)t\right] + \\ &\quad + a_s(t)\sin\left[(2\pi f_c + b_s(t)2\pi f_d)t\right] \qquad (2.53) \end{aligned}$$

where

$$\begin{aligned} a_c(t) = a_1(t), \quad b_c(t) = \frac{-a_4(t)}{a_1(t)} \\ a_s(t) = a_3(t), \quad b_s(t) = \frac{+a_2(t)}{a_3(t)} \end{aligned} \qquad (2.54)$$

In the light of expression (2.47) and (2.53) the Q^2PSK signal can be described in terms of the well known 2-D modulation schemes from the following two points of view, one of which is based on MSK, the other on QPSK:

- From the above expressions the Q^2PSK signal consists of two signals. The first is a cosinusoidal with a frequency of $(f_c \pm f_d)$, and the second is sinusoidal with a frequency of $(f_c \pm f_d)$. The separation between these two signals is $\Delta f = 2f_d = 1/T_s$, which is the minimum spacing needed to guarantee the orthogonality of two FSK signals, thus giving MSK. The Q^2PSK signal can thus be thought of as consisting of two MSK-like signalling schemes operating in parallel, which are in quadrature with respect to each other.
- The Q^2PSK signal can also be seen as the union of two QPSK signals, both of which are first premodulated onto two synchronous subcarriers at a frequency equal to half the symbol rate. ($f_d = 1/2T_s$) These two signals are then modulated onto the in-phase and quadrature components of the RF carrier respectively [27].

This can be vizualised by making use of a 3-D plot of the frequency axis. The representation of Figure 2.5 is not strictly true, but it gives insight into physical operation of Q^2PSK. The "signalling points" are really only true for one sample of each symbol. Any two of these "signalling points" can occur in each Q^2PSK symbol, but one must be chosen from the four present in the real plane, and the other from the four present in the imaginary plane. This gives $4 \times 4 = 16$ possible symbols.

Amplitude and Phase Properties of Q^2PSK Modulation

In this section the time domain properties of Q^2PSK are discussed in more detail.

Any bandpass signal can be represented in the following form

$$s(t) = \mathrm{Re}\left\{\tilde{s}(t)e^{j2\pi f_c t}\right\} \qquad (2.55)$$

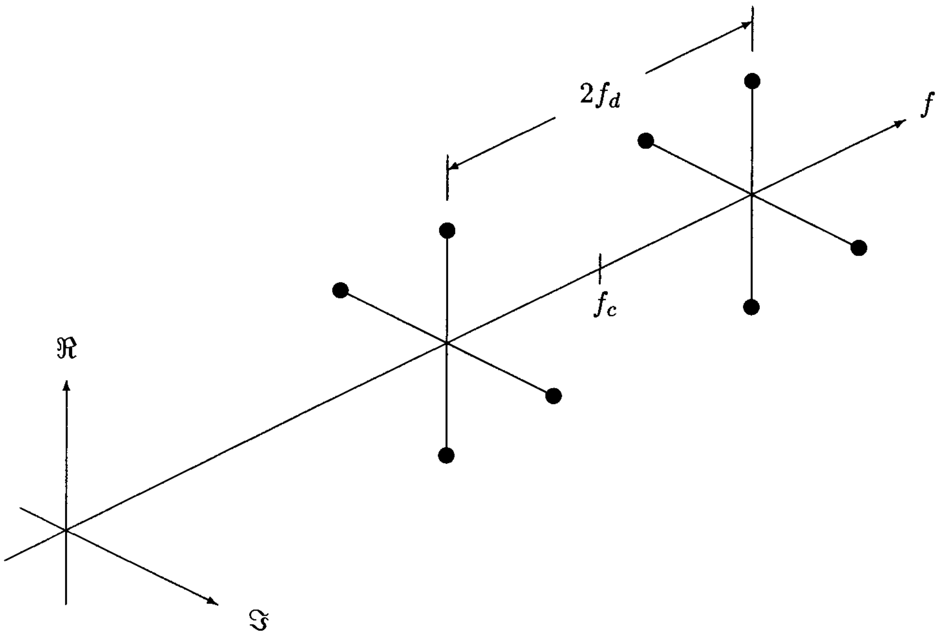

Figure 2.5: Graphical visualisation of Q^{2}PSK in the complex frequency plane

where $\tilde{s}(t)$ is the complex envelope, and is defined as

$$\begin{aligned} \tilde{s}(t) &= a(t)e^{j\theta(t)} \\ &= x(t) + jy(t) \end{aligned} \tag{2.56}$$

Equation (2.55) can be algebraically manipulated to give the following expressions

$$s(t) = x(t)\cos(2\pi f_c t) - y(t)\sin(2\pi f_c t) \tag{2.57}$$
$$\quad = a(t)\cos\left[2\pi f_c t + \theta(t)\right] \tag{2.58}$$

where $a(t)$ denotes the envelope, and $\theta(t)$ the phase of $s(t)$.

The Q^{2}PSK modulation method can also be written in the form of (2.57) by first defining $x(t)$ and $y(t)$, and then using these to find $a(t)$ and $\theta(t)$. By equating expression (2.46) and (2.57), it can be seen that

$$x(t) = a_1(t)p_1(t) + a_2(t)p_2(t) \tag{2.59}$$
$$y(t) = a_3(t)p_1(t) + a_4(t)p_2(t) \tag{2.60}$$

The amplitude of the signal is now obtained from

$$a(t) = \sqrt{x^2(t) + y^2(t)} \tag{2.61}$$

$$\begin{aligned} &= \left\{(a_1(t)p_1(t) + a_2(t)p_2(t))^2 + (a_3(t)p_1(t) + a_4(t)p_2(t))^2\right\}^{\frac{1}{2}} \\ &= \left\{a_1^2(t)p_1^2(t) + 2a_1(t)a_2(t)p_1(t)p_2(t) + a_2^2(t)p_2^2(t) + \right. \\ &\quad \left. + a_3^2(t)p_1^2(t) + 2a_3(t)a_4(t)p_1(t)p_2(t) + a_4^2(t)p_2^2(t)\right\}^{\frac{1}{2}} \end{aligned} \tag{2.62}$$

But $a_i^2 = 1$ as $a_i = \pm 1$. Thus :

$$\begin{aligned} a(t) &= \Big\{p_1^2(t) + 2a_1(t)a_2(t)p_1(t)p_2(t) + p_2^2(t) + \\ &\quad + p_1^2(t) + 2a_3(t)a_4(t)p_1(t)p_2(t) + p_2^2(t)\Big\}^{\frac{1}{2}} \end{aligned} \tag{2.63}$$

Now defining $p_1(t) = \cos(\alpha)$ and $p_2(t) = \sin(\alpha)$, yields :

$$a(t) = \{2 + 2\left[a_1(t)a_2(t) + a_3(t)a_4(t)\right]p_1(t)p_2(t)\}^{\frac{1}{2}} \tag{2.64}$$

and finally, noting that $\sin^2(\alpha) + \cos^2(\alpha) = 1$ and $\sin(\alpha)\cos(\alpha) = \frac{1}{2}\sin(2\alpha)$, we obtain

$$\begin{aligned} a(t) &= \{2 + \left[a_1(t)a_2(t) + a_3(t)a_4(t)\right]p_1(t)p_2(t)\}^{\frac{1}{2}} \\ &= \{2 + \left[a_1(t)a_2(t) + a_3(t)a_4(t)\right]\sin(2\alpha)\}^{\frac{1}{2}} \end{aligned} \tag{2.65}$$

This is a general expression in terms of the shaping frequency α. For the Q²PSK case considered here, $\alpha = \frac{\pi t}{T_s}$, which gives the final expression for the amplitude

$$a(t) = \left\{2 + \left[a_1(t)a_2(t) + a_3(t)a_4(t)\right]\sin\left(\frac{2\pi t}{T_s}\right)\right\}^{\frac{1}{2}} \tag{2.66}$$

Depending on the specific 4-D data vector a_1, a_2, a_3, a_4 to be transmitted, the amplitude of the Q²PSK signal will either be a constant for the symbol period, or a $\sqrt{\sin()}$ function, where the sin() has a frequency of twice the symbol shaping frequency ($f_a = 2f_d$). The three possible amplitude waveforms are plotted in Figure 2.6 below.

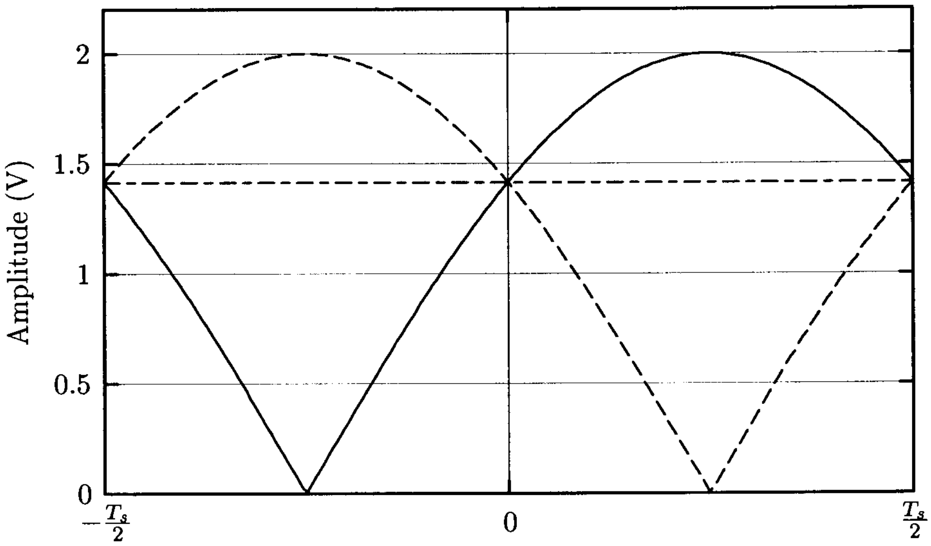

Figure 2.6: Possible amplitude functions for one symbol

A constant amplitude version of Q²PSK may be derived by considering the final expression for $a(t)$. If

$$a_1(t)a_2(t) + a_3(t)a_4(t) = 0 \tag{2.67}$$

then

$$a(t) = 2 \tag{2.68}$$

There are four solutions to (2.67), the most common [28, 11] being

$$a_4(t) = -\frac{a_1(t)a_2(t)}{a_3(t)} \tag{2.69}$$

This coding reduces the throughput rate by 25% as only three bits can be transmitted per symbol. In total this gives eight (2^3) possible data vectors. These vectors are a set of 8 biorthogonal codes with a Hamming distance of $d_{min} = 2$. If use is made of hard decisions at the receiver, this code cannot be used for error correction. The redundant information associated with the fourth data stream can, however, be used to improve the signal to noise ratio of the code.

The phase of the Q^{2}PSK signal is also a function of $x(t)$ and $y(t)$, and is given by

$$\begin{aligned} \theta(t) &= \tan^{-1}\left[\frac{y(t)}{x(t)}\right] && (2.70)\\ &= \tan^{-1}\left[\frac{a_3(t)p_1(t)+a_4(t)p_2(t)}{a_1(t)p_1(t)+a_2(t)p_2(t)}\right] && (2.71) \end{aligned}$$

If $p_1(t) = \cos(\alpha)$ and $p_2(t) = \sin(\alpha)$, then (2.71) can be expanded as

$$\begin{aligned} \theta(t) &= \tan^{-1}\left[\frac{y(t)}{x(t)}\right] && (2.72)\\ &= \tan^{-1}\left[\frac{a_3(t)\cos(\alpha)+a_4(t)\sin(\alpha)}{a_1(t)\cos(\alpha)+a_2(t)\sin(\alpha)}\right] && (2.73)\\ &= \tan^{-1}\left[\frac{\sqrt{2}\cos\left(\alpha+\tan^{-1}\left(-\frac{a_4(t)}{a_3(t)}\right)\right)}{\sqrt{2}\cos\left(\alpha+\tan^{-1}\left(-\frac{a_2(t)}{a_1(t)}\right)\right)}\right] && (2.74)\\ &= \tan^{-1}\left[\frac{\cos\left(\alpha+\frac{\pi}{4}+n\frac{\pi}{2}\right)}{\cos\left(\alpha+\frac{\pi}{4}+m\frac{\pi}{2}\right)}\right] && (2.75) \end{aligned}$$

where

$$m \in 0,1,2,3, \text{ and } n \in 0,1,2,3 \tag{2.76}$$

It can easily be shown that there are only three possible forms of the phase function, which are given by :

$$\theta(t) = \begin{cases} \pm\frac{\pi}{4} \\ \pm(\alpha+\pi) \\ \pm(\alpha+\frac{\pi}{2}) \end{cases} \tag{2.77}$$

It can thus be seen that the phase trajectory of one Q^{2}PSK symbol is either constant or linear, depending on the input bits for that specific symbol, and exhibits abrupt changes at the instant of transition to the following symbol. This means that Q^{2}PSK is not a constant phase modulation (CPM) technique.

The Q^{2}PSK modulation scheme can be extended to maintain constant phase by replacing the half sinusoidal data shaping function with a full sinusoidal function. This unfortunately implies an increase in the bandwidth, and associated loss in spectral efficiency, of the modulation scheme but has the advantage that the spectrum exhibits faster asymptotic fall off due to the smooth phase trajectory between consecutive symbols.

2.4.3 Performance of Q^{2}PSK

In this section the frequency domain properties of Q^{2}PSK are discussed in more detail.

Spectral Properties of Q^{2}PSK Modulation

The Q^{2}PSK signal can be represented as

$$\begin{aligned} s_{Q^2PSK}(t) &= [a_1(t)p_1(t) + a_2(t)p_2(t)]\cos(2\pi f_c t) + \\ &\quad + [a_3(t)p_1(t) + a_4(t)p_2(t)]\sin(2\pi f_c t) \end{aligned} \tag{2.78}$$

where the $\sqrt{2/T_s}$ is an energy normalisation factor for the pulse shapes p_1 and p_2, which are given by

$$p_1(t) = \sqrt{\frac{2}{T_s}}\cos\left(\frac{\pi t}{T_s}\right) \qquad -\frac{T_s}{2} \leq t \leq \frac{T_s}{2} \tag{2.79}$$

$$p_2(t) = \sqrt{\frac{2}{T_s}}\sin\left(\frac{\pi t}{T_s}\right) \qquad -\frac{T_s}{2} \leq t \leq \frac{T_s}{2} \tag{2.80}$$

The four data streams are assumed to be independent, and each may take on a value of +1 or -1 with a probability of $\frac{1}{2}$ respectively for each symbol interval T_s. For each signalling interval the Q^{2}PSK signal can take on one of sixteen possible waveforms, $\{m_i(t)\}_{i=1}^{16}$. The probability of occurrence of a specific waveform $m_i(t)$ is $p_i(t) = \frac{1}{16}$ for all i. The signal set has the following characteristics

- For each waveform $m_i(t)$, there is also a waveform $-m_i(t)$
- The probability of $m_i(t)$ and $-m_i(t)$ is equal.
- The probability of a transition between any two waveforms is the same.

Such a signalling source is said to be negative equiprobable (NEP). The spectral density of such a signal is given by [98]

$$S_{Q^2PSK}(f) = \sum_{i=1}^{16} p_i|M_i(f)|^2 \tag{2.81}$$

where $M_i(f)$ is the Fourier transform of $m_i(t)$, which is given by

$$\begin{aligned} M_i(f) &= \mathcal{F}\{m_i(t)\} \\ &= \int_{-\infty}^{\infty} m_i(t)e^{-j2\pi ft}dt \end{aligned} \tag{2.82}$$

If it is now assumed that $f_c \gg \frac{1}{T_s}$, then the negative portion of the spectrum, which is centered around $-f_c$, will have a minimal effect at $+f_c$. For $f > 0$ each $M_i(f)$ is one of the following 16 combinations.

$$\frac{1}{2}\left[\pm P_1(f-f_c) \pm P_2(f-f_c) \pm jP_1(f-f_c) \pm jP_2(f-f_c)\right] \tag{2.83}$$

where $P_1(f)$ and $P_2(f)$ are the Fourier transforms of the normalised, time limited data shaping pulses $p_1(t)$ and $p_2(t)$ respectively. By substituting this equation for each M_i in (2.81), and cancelling all the cross terms, the following expression is obtained

$$S_{Q^2PSK}(f) = \frac{1}{2}\left[|P_1(f-f_c)|^2 + |P_2(f-f_c)|^2\right] \qquad \text{for } f \geq 0 \tag{2.84}$$

which when translated to the baseband gives

$$S_{Q^2PSK}(f) = \frac{1}{2}\left[|P_1(f)|^2 + |P_2(f)|^2\right] \tag{2.85}$$

The normalised time-limited data pulses can be written as

$$p_1(t) = \sqrt{\frac{2}{T_s}}\cos\left(\frac{\pi t}{T_s}\right)\text{rect}\left(\frac{t}{T_s}\right) \tag{2.86}$$

$$p_2(t) = \sqrt{\frac{2}{T_s}}\sin\left(\frac{\pi t}{T_s}\right)\text{rect}\left(\frac{t}{T_s}\right) \tag{2.87}$$

The Fourier transforms of the pulse shapes are given by

$$P_1(f) = \mathcal{F}\left[p_1(t)\right] \tag{2.88}$$

$$= \frac{2\sqrt{2T_s}}{\pi}\left(\frac{\cos(\pi T_s f)}{1-4T_s^2 f^2}\right) \tag{2.89}$$

$$P_2(f) = \mathcal{F}\left[p_2(t)\right] \tag{2.90}$$

$$= -j\frac{\sqrt{32T_s^3}f}{\pi}\left(\frac{\cos(\pi f T_s)}{1-4T_s^2 f^2}\right) \tag{2.91}$$

These functions are plotted in Figure 2.7. Since $P_1(f)$ is real only, and $P_2(f)$ is imaginary only, only the nonzero parts of the functions are plotted.

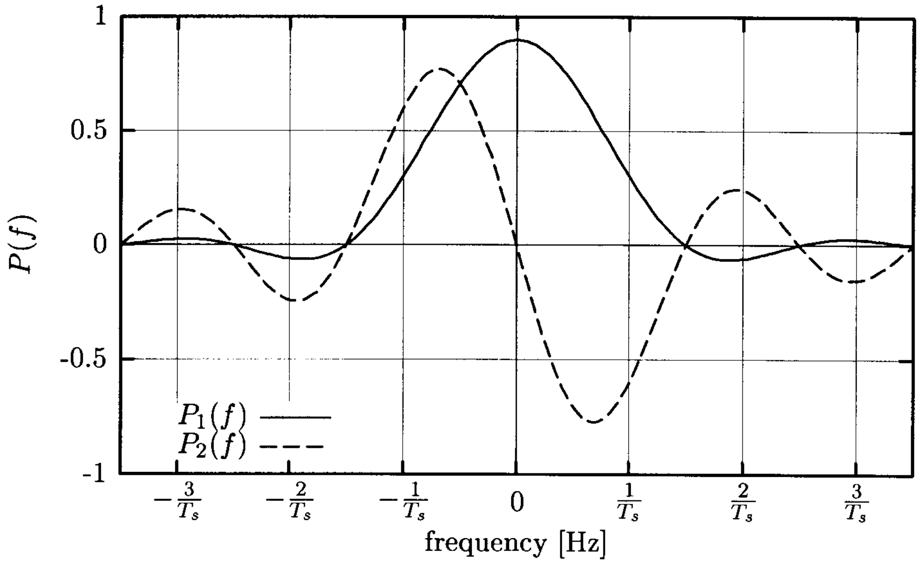

Figure 2.7: Spectra of pulse shaping functions

Now by substitution of (2.89) and (2.91) into (2.85) the following equation is obtained

$$S_{Q^2PSK}(f) = \frac{4T_s}{\pi^2}\left(1+4f^2T_s^2\right)\left(\frac{\cos(\pi f T_s)}{1-4f^2T_s^2}\right) \tag{2.92}$$

To enable this expression to be used for comparison between Q^{2}PSK and other modulation schemes, it should be expressed in terms of T_b, where $T_s = 4T_b$ for the case where the input bit rate is the same as QPSK. This gives the following expression for the PSD of Q^{2}PSK :

$$S_{Q^2PSK}(f) = \frac{16T_b}{\pi^2}\left(1+64f^2T_b^2\right)\left(\frac{\cos(4\pi f T_b)}{1-64f^2T_b^2}\right) \tag{2.93}$$

For the case where the input bit rate is double that of QPSK, $T_s = 2T_b$ giving the following expression for the PSD :

$$S_{Q^2PSK}(f) = \frac{8T_b}{\pi^2}\left(1+16f^2T_b^2\right)\left(\frac{\cos(2\pi f T_b)}{1-16f^2T_b^2}\right) \tag{2.94}$$

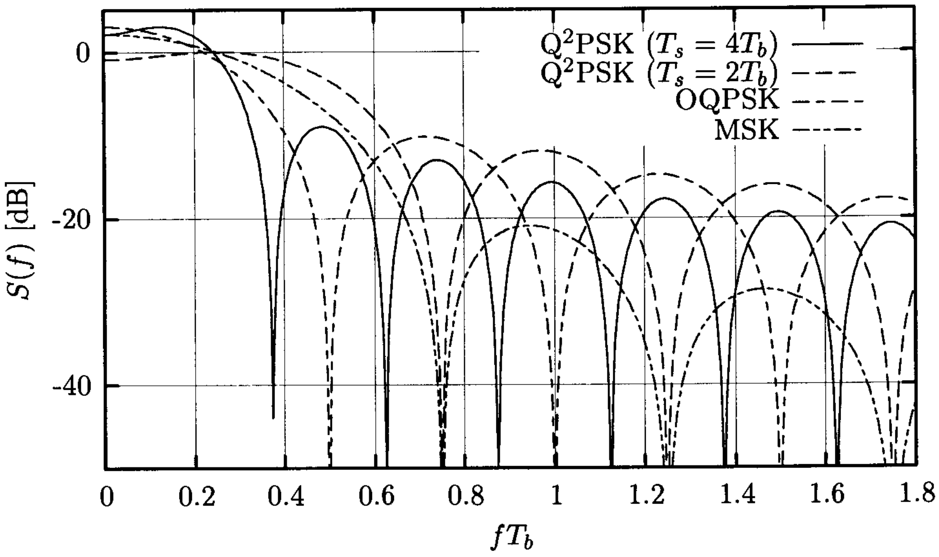

Figure 2.8: Comparative Spectra for various modulation techniques

Comparative plots of the power spectral densities of Q^{2}PSK (for both bit rates), OQPSK and MSK are given in Figure 2.8. From this plot it can be seen that the first zero of Q^{2}PSK is at $fT_b = \frac{3}{8}$, giving a bandwidth efficiency of

$$\rho = \frac{R_b}{B} = \frac{8}{3} = 2.667 \qquad \text{[bits/sec/Hz]} \tag{2.95}$$

This value is lower than the theoretical value of 4 bits/s/Hz due to the choice of pulse shapes. The sinusoidal shapes chosen for $p_1(t)$ and $p_2(t)$ are not optimal, and thus result in a certain amount of spectral broadening. It can easily be shown that this spectral efficiency is the same as that of a QPSK signal, which has been passed through a Nyquist filter with a rolloff factor of $\alpha = 0.5$. In general the spectral efficiency of QPSK as a function of rolloff factor, α is given by :

$$\rho_{QPSK}(\alpha) = \frac{4}{1+\alpha} \tag{2.96}$$

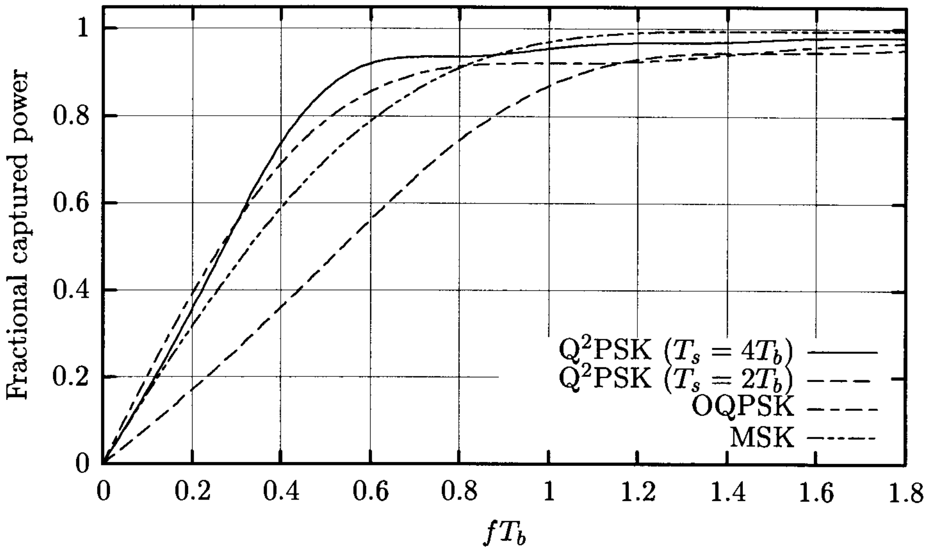

Figure 2.9: Captured power as a function of bandwidth

Figure 2.9 gives a plot of the power captured within a specified bandwidth centred around the carrier frequency, implying a baseband frequency interval of $(-B, B)$. This graph represents a measure of the spectral compactness of the various modulation techniques. This graph is in agreement Fig. 1 of Visintin and Biglieri [12], and thus implies an error in Saha's Figure 3.3 [10].

Visualisation of 4-D Signalling Space for Q²PSK

In this section two methods of visualisation of the 4-dimensional signalling space will be discussed. The first method is a means of visualising the 4-D space in 3-D, and the second projects combinations of two of the orthogonal dimensions at a time onto a two dimensional plane.

Each of the basis functions $\phi_i(t)$ represents one of the four co-ordinate axes. When a i.i.d. (± 1) information stream is modulated onto each dimension, the vertices of a 4-D hypercube become the signalling points in the signal constellation. Each signal point can be described by its coordinates, $\{a_i\}_{i=1}^{4}$, which is a vector consisting of the 4 bits being modulated. When visualising a 4-D signalling space in 3-D the best that one can do is to represent the points as 2 linked 3-D diagrams. The signalling points are denoted by a circle or point, and each of these is labelled by a vector representing its coordinates. The nearest neighbours of each point are then joined to the point by links, and each link is labelled with the Euclidean distance between the two points. This distance is given by

$$d_e = \sqrt{\sum_{i=1}^{N} x_i^2} \qquad N = 4 \tag{2.97}$$

where N is the number of dimensions. By using this method, a picture of the 4-D signalling space can be generated. It is obvoius that there are several topological forms, the first of which is depicted in Figure 2.10. If the second 3-D cube is now scaled, and moved to within the first cube, the topological form of Figure 2.11 is obtained.

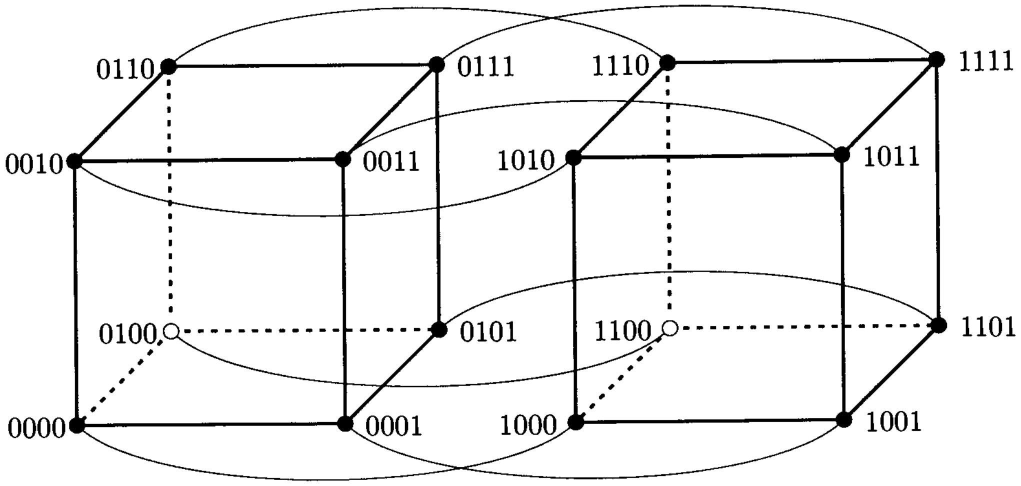

Figure 2.10: 3-D representation of 4-D Q^{2}PSK signal space (Form 1).

The difficulty of representing a four-dimensional space leads to an alternative method which makes use of decoupled two-dimensional subspaces.

The problem of which sets of the basis functions to group together for each two-dimensional plot has been discussed in several papers [13], [32], [29] and [28], without consensus.

In the 2-D subspace method the two demodulated streams which have the same pulse shape, but orthogonal carriers are plotted against each other on two X-Y plots. The basis functions which are plotted against each other are $\phi_1(t)$ against $\phi_3(t)$ and $\phi_2(t)$ against $\phi_4(t)$ This method shows carrier effects clearly and is thus used throughout this dissertation.

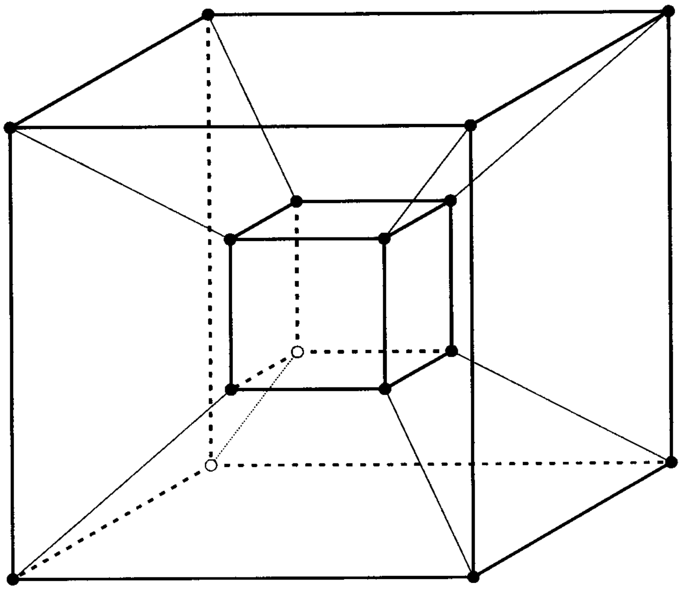

Figure 2.11: 3-D representation of 4-D Q^{2}PSK signal space (Form 2).

Error Rate Properties of Q^{2}PSK Modulation

The efficiency of any modulation scheme can be evaluated relative to other schemes by comparing the amount of energy needed by each scheme to achieve a given probability of bit error, say $P_b(E) = 10^{-5}$. The ratio of the bit energy (E_b) to the noise spectral density(N_0), E_b/N_0, necessary to achieve this specified BER is the standard quantitative parameter used for performance comparisons between digital modulation strategies.

In this section the theoretical error rate of Q^{2}PSK will be derived for the case of an ideal signal geometry, i.e. not bandlimited, where the received signal is corrupted by additive white Gaussian noise (AWGN). The receiver structure is a correlation receiver, which is optimal in the sense that it maximizes the probability of a correct decision in the presence of AWGN.

As the signal set $\phi_i(t)_{i=1}^4$ spans the 4-D signal space, the Q^{2}PSK signal can be represented with respect to this set of axis as

$$s(t) = [a_1(t), a_2(t), a_3(t), a_4(t)] \tag{2.98}$$

where $a_i(t) = \pm 1$, and $+1$ and -1 are equally probable. As discussed previously the signal space geometry is the vertices of a hyper cube of dimension $N = 4$, giving $2^4 = 16$ possible equally likely signals. These signals also have equal energy, E_s. For the case where the vertices of a hyper cube are used for signalling, the probability of a symbol error for any N, where the hyper cube is centered around the origin, is given in [99, 100]. Let $N_0/2$ be the two sided spectral density of the Gaussian noise. The probability of a symbol error is given by

$$P_s(E) = 1 - (1-p)^N \tag{2.99}$$

where

$$\begin{aligned} p &= Q\left(\sqrt{\frac{E_s}{2N_0}}\right) && (2.100)\\ &= \frac{1}{2}\operatorname{erfc}\left(\sqrt{\frac{E_s}{4N_0}}\right) && (2.101)\\ &= \frac{1}{2}\operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) && (2.102)\end{aligned}$$

therefore

$$P_s(E) = 1 - \left(1 - \frac{1}{2}\operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)\right)^N \tag{2.103}$$

The probability of symbol error can be used to provide upper and lower bounds on the bit error probability of Q^{2}PSK by noting that if a symbol error occurs, then at least one bit is in error, and at most all 4 bits are in error. The bounds can thus be expressed as

$$\frac{1}{4}P_s(E) \leq P_b(E) \leq P_s(E) \tag{2.104}$$

where $P_b(E)$ is the probability of a bit error.

Saha [10] has proved the following theorem :

Theorem 2.3 *In the presence of additive white Gaussian noise (AWGN) any modulation scheme which uses all the vertices of some hyper cube, centred at the origin of the signal space geometry, and an optimum receiver for detection has a bit error probability given by*

$$P_b(E) = \frac{1}{2}\,\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \tag{2.105}$$

where E_b is the average bit energy, and $N_0/2$ is the two sided noise spectral density. This probability holds for a hypercube of any dimension N.

From (2.99) it should be noted that, for any fixed p, as N tend towards infinity, the symbol error probability tends to unity. Equation (2.105) on the other hand, shows that the bit error probability is independent of the dimension N. This apparent contradiction can be resolved by noting that the theorem assumes that the bit energy E_b is fixed, which means that the symbol energy E_s is actually increasing with the dimension N.

This theorem illustrates that the hyper cube signal space geometry coupled with the equiprobable use of all vertices is equivalent to an antipodal binary geometry. Thus BPSK, QPSK, OQPSK, MSK and Q^{2}PSK all belong to the same class of signalling schemes, as each makes use of the vertices of some hypercube to transmit data. Each of these schemes has the same bit error probability which is given by (2.106), and plotted in Figure 2.12.

$$P_b(E) = \frac{1}{2}\,\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \tag{2.106}$$

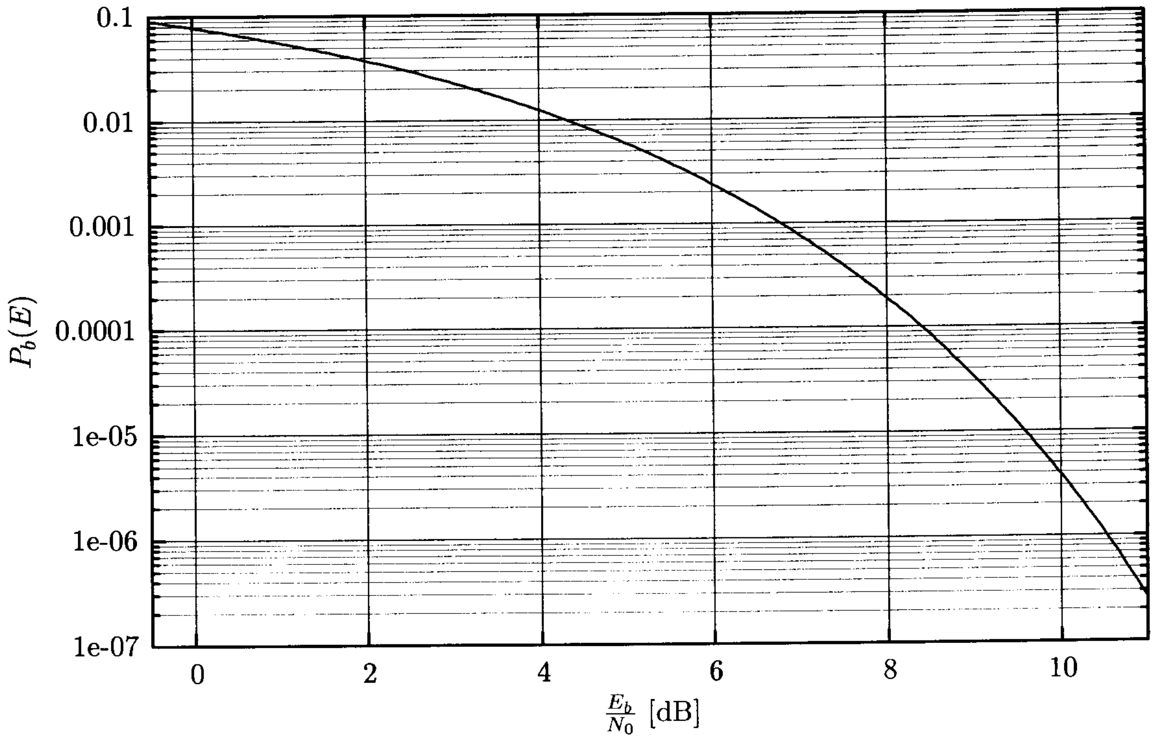

Figure 2.12: Theoretical bit error rate performance of Q^{2}PSK, BPSK, QPSK and MSK

The preceding discussion pertains to channels which are not bandlimited, but in most practical situations the channel is bandlimited. Bandlimiting in the frequency domain affects each modulation scheme by causing intersymbol interference (ISI) in the time domain. Due to the ISI the signal space geometry no longer remains a hyper cube, and as a result the bit error probability is degraded.

Effect of Bandlimiting

In this section the effects of bandlimitng on the Q^{2}PSK signal will be analysed in more detail. The results of this section will be indicative of the type of equaliser structure which will be necessary.

This analysis will be approached in the time domain by modelling the effect of ISI on the Q^{2}PSK signal. ISI is the interference of past and future symbols on the current symbol of interest. This will firstly be modelled as a scaled down, time delayed replica of the Q^{2}PSK signal arriving at the receiver at the same time as the current symbol. Use will be made of a FIR filter to achieve this effect. The impulse response of this filter is given by

$$h(t) = \delta(t) + \alpha\delta(t-\tau) \tag{2.107}$$

where $\delta(t)$ is the Kronecker delta distribution [101], α is relative strength of the delayed replica, and τ is the time delay of the replica. The transmitted Q^{2}PSK signal is given by

$$\begin{aligned} s_{Q^2PSK}(t) &= [a_1(t)p_1(t) + a_2(t)p_2(t)]\cos(2\pi f_c t) + \\ &\quad + [a_3(t)p_1(t) + a_4(t)p_2(t)]\sin(2\pi f_c t) \end{aligned} \tag{2.108}$$

It is assumed that use is made of a correlative receiver as shown in Figure 2.13 below.

The received signal can be written as follows

$$\begin{aligned} r(t) &= s_{Q^2PSK}(t) + \alpha s_{Q^2PSK}(t-\tau) \qquad (2.109)\\ &= [a_1(t)p_1(t) + a_2(t)p_2(t)]\cos(2\pi f_c t) + \\ &\quad + [a_3(t)p_1(t) + a_4(t)p_2(t)]\sin(2\pi f_c t) + \\ &\quad + \alpha\left\{[a_1(t-\tau)p_1(t-\tau) + a_2(t-\tau)p_2(t-\tau)]\cos\left(2\pi f_c(t-\tau)\right) + \right. \\ &\quad \left. + [a_3(t-\tau)p_1(t-\tau) + a_4(t-\tau)p_2(t-\tau)]\sin\left(2\pi f_c(t-\tau)\right)\right\} \\ &= [a_1(t)p_1(t) + a_2(t)p_2(t)]\cos(2\pi f_c t) + \\ &\quad + [a_3(t)p_1(t) + a_4(t)p_2(t)]\sin(2\pi f_c t) + \\ &\quad + \alpha\,[a_1(t-\tau)p_1(t-\tau) + a_2(t-\tau)p_2(t-\tau)]\cos(2\pi f_c t - 2\pi f_c \tau) \\ &\quad + \alpha\,[a_3(t-\tau)p_1(t-\tau) + a_4(t-\tau)p_2(t-\tau)]\sin(2\pi f_c t - 2\pi f_c \tau) \qquad (2.110) \end{aligned}$$

Now defining

$$\theta = 2\pi f_c \tau \tag{2.111}$$

$$A = \alpha\,[a_1(t-\tau)p_1(t-\tau) + a_2(t-\tau)p_2(t-\tau)]\cos(2\pi f_c t - \theta) \tag{2.112}$$

$$B = \alpha\,[a_3(t-\tau)p_1(t-\tau) + a_4(t-\tau)p_2(t-\tau)]\sin(2\pi f_c t - \theta) \tag{2.113}$$

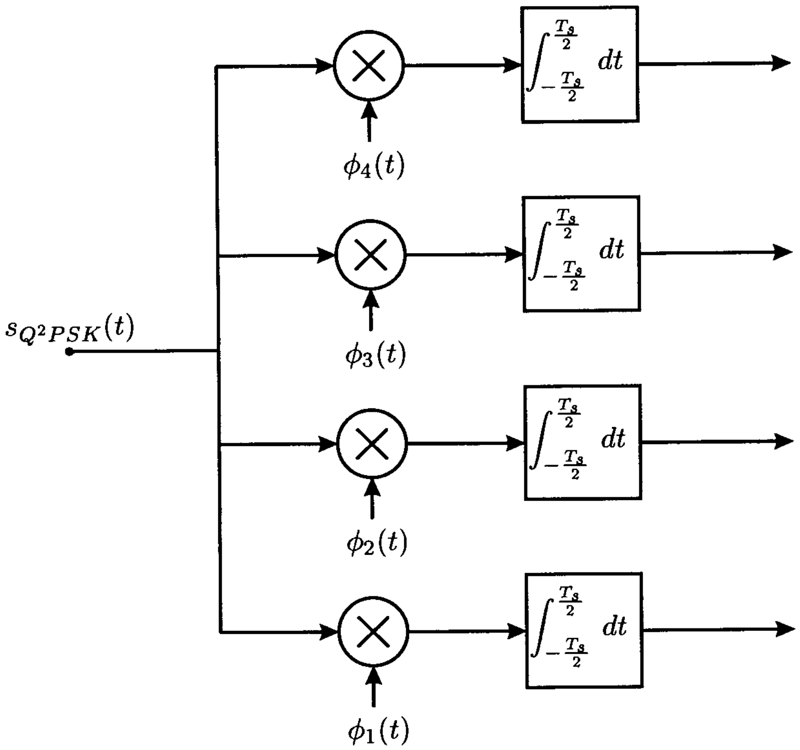

Figure 2.13: Correlative receiver structure.

By expanding the sin() and cos(), A and B can now be written as

$$\begin{aligned} A &= \alpha\left[a_1(t-\tau)p_1(t-\tau)+a_2(t-\tau)p_2(t-\tau)\right] \\ &\qquad [\cos(2\pi f_c t)\cos(\theta)+\sin(2\pi f_c t)\sin(\theta)] \quad (2.114) \\ B &= \alpha\left[a_3(t-\tau)p_1(t-\tau)+a_4(t-\tau)p_2(t-\tau)\right] \\ &\qquad [\sin(2\pi f_c t)\cos(\theta)-\cos(2\pi f_c t)\sin(\theta)] \quad (2.115) \end{aligned}$$

Now by factoring $\cos(2\pi f_c t)$ and $\sin(2\pi f_c t)$ out, the expression for the received signal can be written in the form

$$r(t) = I(t)\cos(2\pi f_c t) + Q(t)\sin(2\pi f_c t) \quad (2.116)$$

where

$$\begin{aligned} I(t) &= a_1(t)p_1(t)+a_2(t)p_2(t)+\alpha\left[a_1(t-\tau)p_1(t-\tau)+a_2(t-\tau)p_2(t-\tau)\right]\cos(\theta) - \\ &\quad -\alpha\left[a_3(t-\tau)p_1(t-\tau)+a_4(t-\tau)p_2(t-\tau)\right]\sin(\theta) \quad (2.117) \\ Q(t) &= a_3(t)p_1(t)+a_4(t)p_2(t)+\alpha\left[a_1(t-\tau)p_1(t-\tau)+a_2(t-\tau)p_2(t-\tau)\right]\sin(\theta) + \\ &\quad +\alpha\left[a_3(t-\tau)p_1(t-\tau)+a_4(t-\tau)p_2(t-\tau)\right]\cos(\theta) \quad (2.118) \end{aligned}$$

Equations (2.117) and (2.118) are also the In phase and Quadrature phase low pass outputs after coherent demodulation. A signal on which a decision can be made, is formed by sampling at the optimal sampling point in time and applying a binary decision. The $I(t)$ and $Q(t)$ signals are each split into two branches, and then passed into filters matched to $p_1(t)$ and $p_2(t)$. The impulse response of these matched filters are given by

$$\begin{aligned} h_{I_{p_1}}(t) &= p_1\left(\frac{T_s}{2}-t\right) && (2.119)\\ h_{I_{p_2}}(t) &= p_2\left(\frac{T_s}{2}-t\right) && (2.120)\\ h_{Q_{p_1}}(t) &= p_1\left(\frac{T_s}{2}-t\right) && (2.121)\\ h_{Q_{p_2}}(t) &= p_2\left(\frac{T_s}{2}-t\right) && (2.122)\end{aligned}$$

After the matched filtering process, the output of the filters, $y_{I_{p_1}}(t)$,$y_{I_{p_2}}(t)$,$y_{Q_{p_1}}(t)$,$y_{Q_{p_2}}(t)$, respectively, is sampled at the optimum position relative to the original pulse train. The output of each of these samplers is now derived.

Firstly the convolution of the In phase branch with the filter matched to $p_1(t)$ yields :

$$\begin{aligned} y_{I_{p_1}}(t) &= I(t)\otimes h_{I_{p_1}}(t) && (2.123)\\ &= I(t)\otimes p_1\left(\frac{T_s}{2}-t\right) \end{aligned}$$

where $\otimes$ is the convolution operator. The above equation can be split up into a set of constituent convolutions due to the distributive property of convolution.

$$\begin{aligned} y_1(t) &= [a_1(t)p_1(t)+a_2(t)p_2(t)]\otimes p_1\left(\frac{T_s}{2}-t\right) && (2.124)\\ &= \int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} a_1(u)p_1(u)p_1\left(\frac{T_s}{2}-t+u\right)+a_2(u)p_2(u)p_1\left(\frac{T_s}{2}-t+u\right)du && (2.125)\end{aligned}$$

Now by sampling at $t=T_s/2$ and keeping in mind that $a_1(u)$ and $a_2(u)$ are constant over the interval of interest, and by making use of the following definitions

$$\begin{aligned} \int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} p_1^2(u)du &= E_{p_1} \quad \text{and} && (2.126)\\ \int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} p_1(u)p_2(u)du &= 0 && (2.127)\end{aligned}$$

we obtain

$$y_1\left(\frac{T_s}{2}\right)=a_1E_{p_1} \qquad (2.128)$$

where E_{p_1} is the energy contained in $p_1(t)$ over one symbol period.

Next the multipath component related to the delayed version of $p_1(t)$ is considered. The second constituent convolution can thus be expressed as

$$\begin{aligned} y_2(t) &= \alpha\left[a_1(t-\tau_m)p_1(t-\tau_m)\cos(\theta)-a_3(t-\tau_m)p_1(t-\tau_m)\sin(\theta)\right]\otimes\\ &\quad p_1\left(\frac{T_s}{2}-t\right) && (2.129)\\ &= \alpha\cos(\theta)\int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} a_1(u-\tau_m)p_1(u-\tau_m)p_1\left(\frac{T_s}{2}-t+u\right)du-\\ &\quad -\alpha\sin(\theta)\int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} a_3(u-\tau_m)p_1(u-\tau_m)p_1\left(\frac{T_s}{2}-t+u\right)du && (2.130)\end{aligned}$$

which when sampled at $t = T_s/2$ gives

$$\begin{aligned} y_2\left(\frac{T_s}{2}\right) &= \alpha\cos(\theta)\int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} a_1(u-\tau_m)p_1(u-\tau_m)p_1(u)du - \\ &\quad -\alpha\sin(\theta)\int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} a_3(u-\tau_m)p_1(u-\tau_m)p_1(u)du \qquad (2.131) \end{aligned}$$

Due to the fact that the multipath delay τ_m is not necessarily an integer multiple of the symbol period T_s, the above integrals will usually span two adjacent symbols. To deal with this problem, each integral is split into two integrals, one for each symbol which the matched filter spans in time. To this end it is necessary to make the following definitions:

$$n = \left\lceil \frac{\tau_m}{T_s} \right\rceil \qquad (2.132)$$

$$z = \left(n - \frac{1}{2}\right)T_s - \tau_m \qquad (2.133)$$

where z is the symbol transition spanned by the matched filter under consideration. It is also necessary to define the partial autocorrelation and cross-correlation coefficients as follows

$$c'_{ij}(z,\tau_m) = \int_{-\frac{T_s}{2}}^{z} p_i(u)p_j(u-\tau_m)du \qquad (2.134)$$

$$c''_{ij}(z,\tau_m) = \int_{z}^{\frac{T_s}{2}} p_i(u)p_j(u-\tau_m)du \qquad (2.135)$$

which can be simplified by elimination of the dependence on τ_m as follows

$$c'_{ij}(z) = \int_{-\frac{T_s}{2}}^{z} p_i(u)p_j\left(u - z + \frac{T_s}{2}\right)du \qquad (2.136)$$

$$c''_{ij}(z) = \int_{z}^{\frac{T_s}{2}} p_i(u)p_j\left(u + z - \frac{T_s}{2}\right)du \qquad (2.137)$$

where the accents are used to denote the integral position. The first delayed bit is denoted as a'_i and the second bit as a''_i to differentiate between the data bits spanned by the matched filter. $y_2(t)$ can now be written as follows

$$y_2\left(\frac{T_s}{2}\right) = \alpha\cos(\theta)\left[a'_1c'_{11}(z) + a''_1c''_{11}(z)\right] - \alpha\sin(\theta)\left[a'_3c'_{11}(z) + a''_3c''_{11}(z)\right] \qquad (2.138)$$

Similarly

$$y_3\left(\frac{T_s}{2}\right) = \alpha\cos(\theta)\left[a'_2c'_{12}(z) + a''_2c''_{12}(z)\right] - \alpha\sin(\theta)\left[a'_4c'_{12}(z) + a''_4c''_{12}(z)\right] \qquad (2.139)$$

The final output of the matched filter is given by

$$\begin{aligned} y_{I_{p1}}\left(\frac{T_s}{2}\right) &= y_1\left(\frac{T_s}{2}\right) + y_2\left(\frac{T_s}{2}\right) + y_3\left(\frac{T_s}{2}\right) \qquad (2.140) \\ &= a_1E_{p1} + \alpha\cos(\theta)\left[a'_1c'_{11}(z) + a''_1c''_{11}(z) + a'_2c'_{12}(z) + a''_2c''_{12}(z)\right] \\ &\quad -\alpha\sin(\theta)\left[a'_3c'_{11}(z) + a''_3c''_{11}(z) + a'_4c'_{12}(z) + a''_4c''_{12}(z)\right] \qquad (2.141) \end{aligned}$$

This calculation is then repeated for the remaining matched filter on the In phase channel, giving

$$\begin{aligned} y_{I_{P2}}\left(\frac{T_s}{2}\right) &= a_2E_{p2} + \alpha\cos(\theta)\left[a_1'c_{21}'(z) + a_1''c_{21}''(z) + a_2'c_{22}'(z) + a_2''c_{22}''(z)\right] \\ &\quad -\alpha\sin(\theta)\left[a_3'c_{21}'(z) + a_3''c_{21}''(z) + a_4'c_{22}'(z) + a_4''c_{22}''(z)\right] \qquad (2.142) \end{aligned}$$

By repeating these calculations the matched filter outputs for the Quadrature channel are given by

$$\begin{aligned} y_{Q_{P1}}\left(\frac{T_s}{2}\right) &= a_3E_{p1} + \alpha\sin(\theta)\left[a_1'c_{11}'(z) + a_1''c_{11}''(z) + a_2'c_{12}'(z) + a_2''c_{12}''(z)\right] \\ &\quad +\alpha\cos(\theta)\left[a_3'c_{11}'(z) + a_3''c_{11}''(z) + a_4'c_{12}'(z) + a_4''c_{12}''(z)\right] \qquad (2.143) \end{aligned}$$

and

$$\begin{aligned} y_{Q_{P2}}\left(\frac{T_s}{2}\right) &= a_4E_{p2} + \alpha\sin(\theta)\left[a_1'c_{21}'(z) + a_1''c_{21}''(z) + a_2'c_{22}'(z) + a_2''c_{22}''(z)\right] \\ &\quad +\alpha\cos(\theta)\left[a_3'c_{21}'(z) + a_3''c_{21}''(z) + a_4'c_{22}'(z) + a_4''c_{22}''(z)\right] \qquad (2.144) \end{aligned}$$

By grouping the outputs of the two matched filters which have the same pulse shape as complex numbers, the following expressions can be obtained

$$\begin{aligned} y_{G_{P1}} &= (a_1 + ja_3)E_{p1} + \alpha(W + jX)e^{j\theta} \qquad (2.145) \\ y_{G_{P2}} &= (a_2 + ja_4)E_{p2} + \alpha(Y + jZ)e^{j\theta} \qquad (2.146) \end{aligned}$$

where

$$\begin{aligned} W &= a_1'c_{11}'(z) + a_1''c_{11}''(z) + a_2'c_{12}'(z) + a_2''c_{12}''(z) \qquad (2.147) \\ X &= a_3'c_{11}'(z) + a_3''c_{11}''(z) + a_4'c_{12}'(z) + a_4''c_{12}''(z) \qquad (2.148) \\ Y &= a_1'c_{21}'(z) + a_1''c_{21}''(z) + a_2'c_{22}'(z) + a_2''c_{22}''(z) \qquad (2.149) \\ Z &= a_3'c_{21}'(z) + a_3''c_{21}''(z) + a_4'c_{22}'(z) + a_4''c_{22}''(z) \qquad (2.150) \end{aligned}$$

From these expressions the effect of the multipath can be established. Equation (2.145) and (2.146) give the received complex data symbols separately from the effect of the received multipath components. To aid in the visualisation of the signal constellation under multipath conditions the probability density function for the interference term will be calculated. The interference terms, W,X,Y and Z have a similar form, which can be generalised as

$$V = \alpha(a_i'c_i' + a_i''c_i'' + a_j'c_j' + a_j''c_j'') \qquad (2.151)$$

It is known that the a_i's and a_j's have PDF's given by

$$p_a(x) = \frac{1}{2}\left(\delta(x+1) + \delta(x-1)\right) \qquad (2.152)$$

The PDF of the interference term will have equiprobable impulse functions at the following set of positions on the x axis.

$$x = \left\{\pm\alpha c_i' \pm \alpha c_i'' \pm \alpha c_j' \pm \alpha c_j''\right\} \qquad (2.153)$$

The probability of each of the 16 impulse functions at the given positions will be 1/16. Once the interference term has been superimposed onto the original signal constellation point, the $x = 0$ of the interference term will lie on the position of the original signal constellation point.

There will then be 8 interference points above, and 8 interference points below the original signal constellation point. Effectively the original complex signal point will be replaced by a set of 256 points (16 $\times$ 16) which is rotated by θ radians.

This analysis has shown that under bandlimiting situations, the orignal complex signal constellations are affected detramentally by the addition of an interference term which is a function of all four the data streams at some delay (τ_m) relative to the current symbol being received.

2.5 GENERALIZED QUADRATURE-QUADRATURE PHASE SHIFT KEYING

The original Q^{2}PSK makes use of two orthogonal carriers, and two orthogonal pulse shapes to form a four dimensional signalling space. The concept can be generalized to higher dimensions by the use of additional pulse shapes. If use is made of n mutually orthogonal pulse shapes, and two orthogonal carriers, then the dimension of the resultant signalling space is $2n$. The different forms of Q^{2}PSK are distinguished from one another by specifying the number of pulse shapes used. For example, the form wherein three pulse shapes are used to form a six-dimensional signalling space is denoted as Q^{2}PSK ($n = 3$) [10].

2.5.1 Formulation of Generalized Q^{2}PSK

The set of n pulse shaping functions is given by

$$\{p_i(t)\}_{i=1}^{n} \qquad -T_s/2 \leq t < T_s/2 \tag{2.154}$$

where

$$\int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} p_i(t)p_j(t)dt = \delta_{ij} \tag{2.155}$$

which, along with the two orthogonal carriers of frequency f_c, are used to generate the $2n$-dimensional signalling space. Use is made of two sets of n i.i.d. data streams to modulate these signals. The data streams are given by

$$\{a_i(t)\}_{i=1}^{n} \qquad -T_s/2 \leq t < T_s/2 \tag{2.156}$$

$$\{b_i(t)\}_{i=1}^{n} \qquad -T_s/2 \leq t < T_s/2 \tag{2.157}$$

and the modulated signal is given by

$$s(t) = \sum_{i=1}^{n} a_i(t)p_i(t)cos(2\pi f_c t) + \sum_{i=1}^{n} b_i(t)p_i(t)sin(2\pi f_c t) \tag{2.158}$$

This modulated signal thus makes use of the vertices of hypercube of dimension $2n$. The overall bit rate of the system is given by

$$R_b = \frac{2n}{T_s} \tag{2.159}$$

The special case of $n = 1$ is a two dimensional modulation scheme such as QPSK, and the case of $n = 2$ represents the orignal Q^{2}PSK modulation scheme.

2.6 CONCLUSION

An in-depth discussion of the dimensionality of a signal has been presented in this chapter. These concepts have been extended to show that two-dimensional modulation schemes do not necessarily utilize all the available dimensions in a given bandwidth, for a given symbol period. This fact is used to develop and describe a multidimensional digital modulation scheme which is referred to as quadrature-quadrature phase shift keying and is denoted by the abbreviation Q^2PSK. The bit error probability of multidimensional modulation schemes is also discussed, and the effect of multipath on the signal constellation has been derived.

CHAPTER 3

MOBILE WIRELESS CHANNEL CHARACTERISTICS and MODELLING

3.1 INTRODUCTION

The radio channel is commercially attractive as it can be harnessed to provide a tetherless communication link to a mobile user. In terms of the wide range of impairments imposed on a signal, it is one of the most hostile environments in which to operate. This is due to the multitude of propagation effects which have an effect on the transmitted electromagnetic wave before it is received. The effects on the electromagnetic wave are reflection, scattering, diffraction and attenuation, all of which can be attributed to the environment surrounding the antenna, and the user's motion through this environment. The various scattered and reflected components arriving simultaneously at the antenna interfere constructively and destructively with each other to build up a very irregular field distribution which gives rise to variable attenuation (fading) and distortion of the received signal.

In this section the effect of the VHF/UHF channel on the transmitted signal will be discussed in terms of the large-scale and small-scale (in terms of antenna position) effects. The large-scale effects will be discussed briefly for completeness, and the small-scale effects will be discussed in more detail. Attention will also be given to the accurate modelling and simulation of these small-scale effects.

3.1.1 Classification of Transmission Impairments

The large-scale channel effects represent the average signal power or path loss due to motion over large areas. This phenomenon is affected by prominent terrain features, such as hills, forests and clumps of buildings, between the transmitter and the receiver. These effects can be broadly categorized into two main groups, namely path loss and shadowing.

Small-scale channel effects are a result of movement of the receiver relative to the transmitter by amounts in the order of one wavelength. These changes in position are responsible for drastic changes in the amplitude and phase of the received signal. The small-scale effects can be categorized into two main causes, which are time spreading of the signal and the time variant nature of the channel.

The classification of these channel effects is summarised in Figure 3.1.

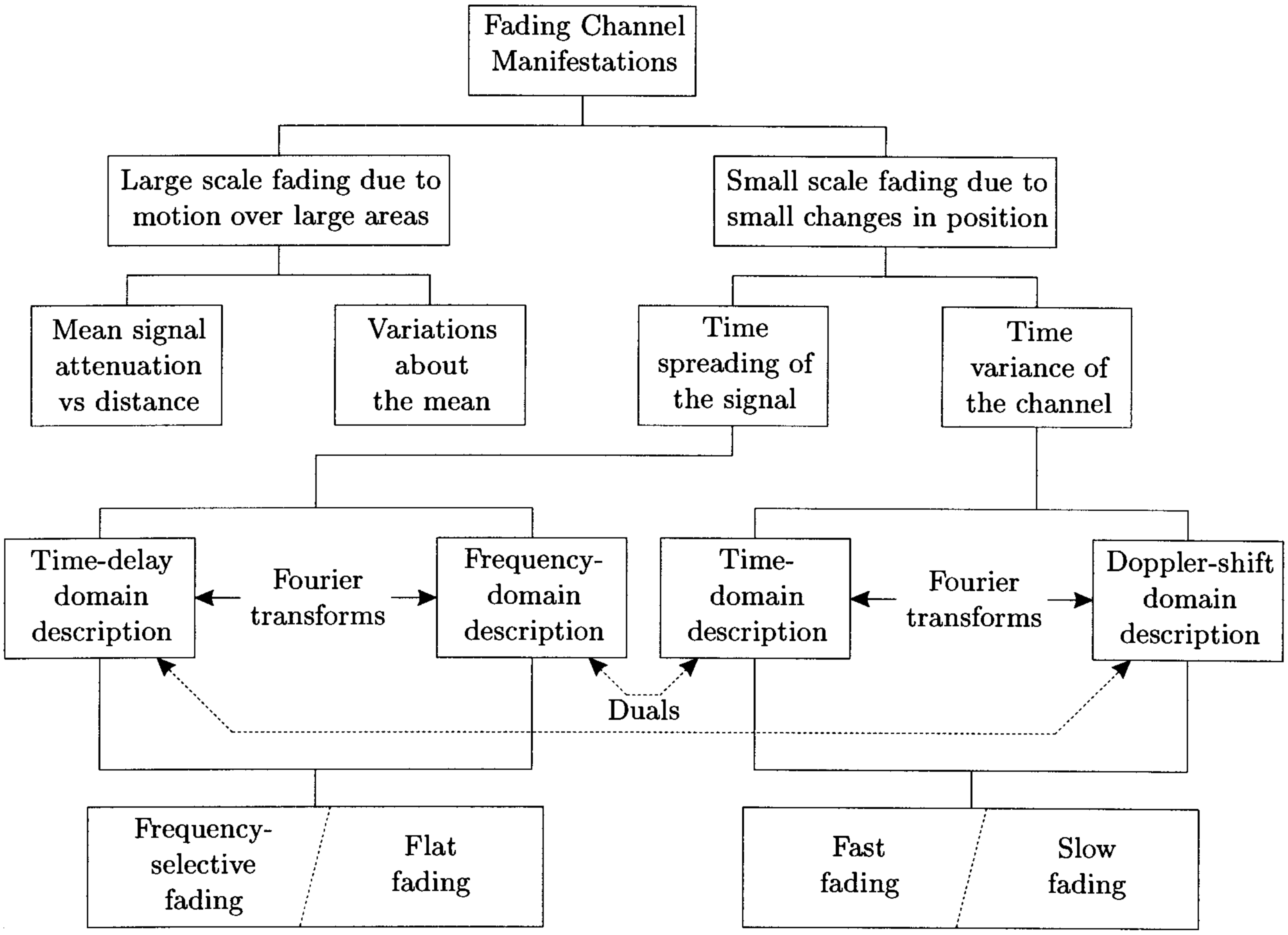
Fading Channel Manifestations
Large scale fading due to motion over large areas
Small scale fading due to small changes in position
Mean signal attenuation vs distance
Variations about the mean
Time spreading of the signal
Time variance of the channel
Time-delay domain description
Fourier transforms
Frequency-domain description
Time-domain description
Fourier transforms
Doppler-shift domain description
Duals
Frequency-selective fading
Flat fading
Fast fading
Slow fading

Figure 3.1: Fading Channel Manifestations

3.1.2 Signal Propagation Mechanisms

There are three major mechanisms which have an impact on the propagation of electromagnetic waves in a mobile communication system.

- **Reflection** occurs when a propagating electromagnetic wave impinges on a smooth surface which is much larger than the wavelength of the radio frequency signal. The reflection is an attenuated form of the incident wave and is reflected away from the surface.

- **Diffraction** occurs when the line of sight (LOS) between the Transmitter and Receiver is obstructed by a dense, sharply edged body with dimensions much larger than the wavelength of the signal. The secondary wave radiated from the object is also present behind the object. The receiver is thus in the radio shadow of the object, and this effect is sometimes referred to as "shadowing". This effect accounts for the fact that the receiver can still detect RF energy even though the direct path is obstructed.

- **Scattering** occurs when a radio wave impinges on either a large rough surface or a surface of dimension in the order of a wavelength or less. The incoming wave is reflected in all directions, thus the term "scattering".

All these effects have one common characteristic in that they are responsible for multiple versions of the original transmitted wave arriving at the receive antenna. Each one of these waves has its own time delay, phase shift and attenuation depending on the path it traveled to arrive at the antenna. As these waves interact with each other destructive or constructive interference can occur. Constructive interference may be beneficial to the communication link, but it is the destructive interference which ultimately limits the performance of the link.

3.1.3 Modelling Channel Effects

From the above discussion it can be seen that predicting the signal strength and phase at the receiver by means of purely electromagnetic models can easily become an intractable mathematical problem, even given a powerful computer. This approach is used mainly in the modelling of indoor communication channels and the most widely used methods are based on ray-tracing.

Instead of trying to solve Maxwell's equations with a very complicated set of boundary conditions, not even to mention the number of different topologies, we try to model and predict the occurrence of destructive interference in a statistical sense.

3.2 LARGE-SCALE FADING EFFECTS

3.2.1 Path Loss

Path loss is a quantitative measure of the loss in signal strength as a function of the distance to the receiver. Path loss increases with an increase in the distance between the transmitter and receiver. Path loss in indoor environments differs from that in outdoor environments due to the differences in surroundings.

Modelling Path loss

For the mobile radio channel Okumura [102] made comprehensive path-loss measurements, which were later transformed into parametric equations by Hata [103]. For this type of channel the mean path loss, $\overline{L_p}(d)$, as a function of distance, d, between the transmitter and receiver is proportional to the nth power of d, relative to some reference distance d_0 [34].

$$\overline{L_p}(d) \propto \left(\frac{d}{d_0}\right)^n \tag{3.1}$$

This relationship is often stated in decibels as

$$\overline{L_p}(d)(dB) = \overline{L_p}(d_0)(dB) + 10n\log\left(\frac{d}{d_0}\right) \tag{3.2}$$

The reference distance, d_0 is a close-in reference distance, which must be in the far field of the antenna, and is determined from measurements close to the transmitter. The bars over the variable in (3.1) and (3.2) denote ensemble averages over all possible path loss values for a given distance d. Some typical values for n are listed in Table 3.1 for various environments.

Environment	Path Loss Exponent (n)
Free Space	2
Urban area cellular radio	2.7 to 3.5
Shadowed urban cellular radio	3 to 5
In building line-of-sight	1.6 to 1.8
Obstructed in building	4 to 6
Obstructed in factories	2 to 3

Table 3.1: Path loss exponents for various environments

It is interesting to note that for line-of-sight (LOS) propagation in a building it is possible that n is less than 2, which is the exponent for free space propagation. This effect occurs for propagation in very strongly guided wave phenomena such as indoor LOS propagation, and urban street LOS propagation.

3.2.2 Shadowing

Equation (3.2) is an average over all path loss values for a given d and does not take into account that the surrounding environmental clutter can differ vastly from one location to the next. Measurements have shown that for any value of d the path loss is distributed normally (in decibels) around the average. This leads to an extension of (3.1) as follows

$$\overline{L_p}(d)(dB) = \overline{L_p}(d_0)(dB) + 10n\log\left(\frac{d}{d_0}\right) + X_\sigma \tag{3.3}$$

where X_σ is a site- and distance-dependent normally distributed random variable (in decibels) with zero mean and standard deviation σ also in decibels. This leads to the log-normal distribution if one is not working in decibels. This phenomenon is thus referred to as "log-normal shadowing". The log-normal distribution is given by

$$p_X(x) = \frac{1}{x\sqrt{2\pi\sigma^2}}\exp\left[-\frac{(\ln(x)-m)^2}{2\sigma^2}\right] \tag{3.4}$$

In the shadowing case $m = 0$ and standard deviation, σ is usually obtained by measurements, and takes on values in the 6-10 dB range or greater, depending on the surrounding environment.

3.3 SMALL-SCALE FADING EFFECTS

3.3.1 Multipath Effects

Multipath propagation is responsible for the small-scale fading effects in the radio channel. Reflections of the transmitted electromagnetic waves combine at the receive antenna with random time delays and phases relative to the original signal. This results in a received signal which can vary widely in amplitude and phase. The three most important effects of multipath are

- Rapid changes in signal strength over small travel distances or time intervals.
- Random frequency modulation due to varying Doppler shifts of individual multipath signals.
- Time dispersion (echoes) caused by the differing path lengths of each multipath.

These effects are interrelated, depending of the combination of channel characteristics and communication system parameters.

The approach in this section will be to first consider the effect of motion on the received signal, then the effect of close-in scattering, and then the effect of longer range reflections.

3.3.2 Receiver Motion in Free Space

Consider an antenna moving through free space at velocity v, while receiving a radio signal from a remote source S. If the receiver is moving in the $X-Y$ plane, and the z direction is used to represent height above the $X-Y$ plane, then the velocity of the receiver in the direction of the line between the transmitter and receiver is

$$v' = v\cos(\theta) \tag{3.5}$$

where v is the velocity of the receiver. This situation is depicted in Figure 3.2.

It can be shown [34] that the received signal has been shifted in frequency by

$$f_d = \frac{v}{\lambda}\cos(\theta) \tag{3.6}$$

where f_d is known as the Doppler shift, and λ is the wavelength of the RF signal. This equation can be rewritten in terms of the carrier frequency, f_c, as

$$f_d = \frac{vf_c}{c}\cos(\theta) \tag{3.7}$$

where c is the speed of light in free space. The actual frequency perceived by the receiver is thus given by

$$f_c' = f_c + f_d \tag{3.8}$$

Figure 3.2 is a three dimensional representation of the problem. From the definition of the angle θ in Figure 3.2 the Doppler shift is negative if the distance between the transmitter and receiver

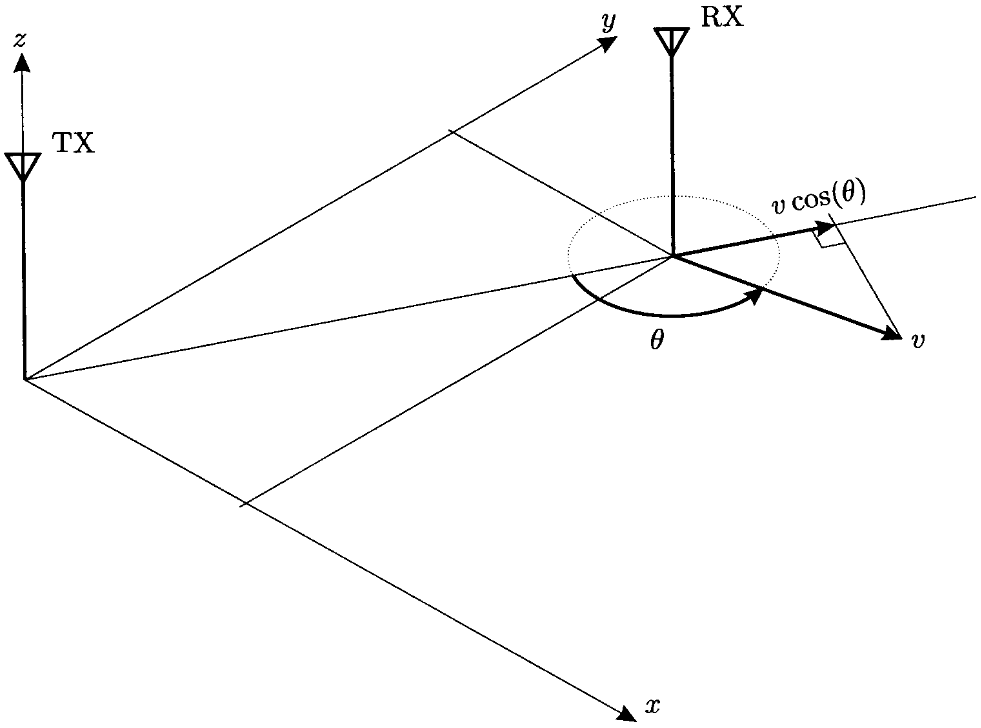

Figure 3.2: Receiver motion

is increasing, and positive if the distance is decreasing. The Doppler shift takes on a maximum value of $+vf_c/c$ for motion directly towards the transmitter, and a minimum value of $-vf_c/c$ for motion directly away from the transmitter.

Now consider the same scenario as in the previous section, only now the $X - Y$ plane is an actual surface which is capable of scattering the electromagnetic waves impinging onto it. We will now consider the effect of these scattered waves on the received signal. For the time being the line of sight propagation as discussed in the previous section will not be considered in this section.

Clarke [104] developed a statistical model in which the characteristics of the electromagnetic fields at the receive antenna are derived from the scattering. This model assumes a stationary transmit antenna with a vertically polarized antenna. The N electromagnetic waves which arrive at the receive antenna are assumed to be azimuthal plane waves with arbitrary angles of arrival, arbitrary carrier phases, and each wave having equal average amplitude. The equal average amplitude assumption implies that the scattered wave components arriving at the receive antenna will experience similar attenuation, and thus have similar small-scale propagation distances. This in turn implies that there will be very little difference in the time delay for each path, so relative to any data modulation on the carrier these scattered components are seen to be replicas of the transmitted signal, all having similar propagation delays, τ_k, but random phases, ϕ_k. It has been shown by Lam and Özlütürk [105] that τ_k and ϕ_k can be modelled as mutually independent random variables for large carrier frequencies, f_c. Figure 3.3 is a diagrammatic representation of the situation discussed above.

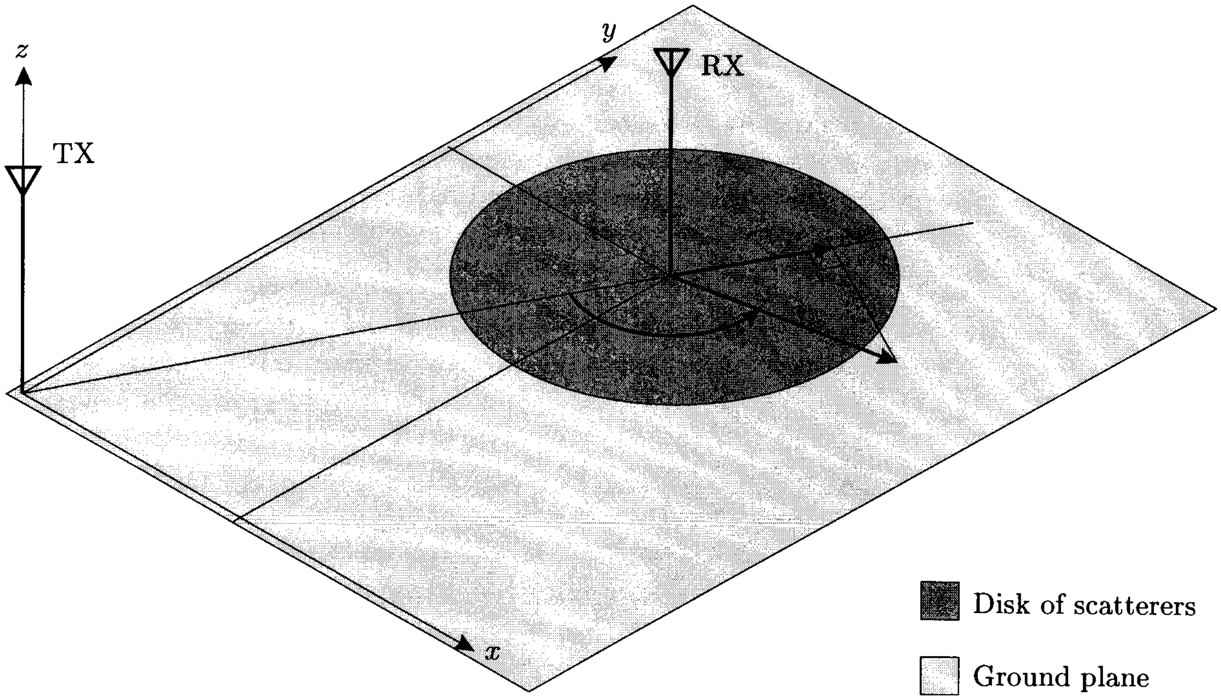

Figure 3.3: Receiver motion with ground plane and close-in disk of scatterers

Characteristics of Received Signal

Since the Doppler shift is small compared to the carrier frequency, $f_d \ll f_c$, the received fields may be modelled as narrowband random processes [106]. If N is sufficiently large, then the total received field may be modelled as Gaussian. The phase angles of each signal are assumed to be uniformly distributed on the interval $(0, 2\pi]$. Rice [107], in his classic paper has shown that the envelope, r, of the received signal has a Rayleigh envelope, which is given by

$$p(r) = \begin{cases} \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right) & 0 \leq r \leq \infty \\ 0, & r < 0 \end{cases} \tag{3.9}$$

This distribution is also known as the χ distribution with two degrees of freedom. Equation (3.9) is the PDF which results from the Random Variable (RV) transformation associated with

$$R = \sqrt{X^2 + Y^2} \tag{3.10}$$

where X and Y are independent zero mean Gaussian distributed RV's each with standard deviation σ.

If there is a line-of-sight propagation path, then it is equivalent to considering the RV transformation associated with equation (3.10), but where one of the input PDF's now has a non-zero mean, m. Both input RV's still have a standard deviation of σ. The output PDF then has a Rician distribution given by

$$p(r) = \begin{cases} \frac{r}{\sigma^2} \exp\left(-\frac{m^2+r^2}{2\sigma^2}\right) I_0\left(\frac{rm}{\sigma^2}\right) & 0 \leq r \leq \infty \\ 0, & r < 0 \end{cases} \tag{3.11}$$

where $I_0(\cdot)$ is the zeroth-order modified Bessel function of the first kind. This distribution is sometimes referred to as the Rice-Nakagami distribution in recognition of the work of Nakagami around the time of World War II. The Ricean distribution is often described in terms of the "Ricean factor", K, which is defined as the ratio of the deterministic signal power to the variance of the multipath. This factor is usually expressed in dB's by the following relationship

$$K = 10 \log \left(\frac{m^2}{2\sigma^2}\right) \qquad [\text{dB}] \tag{3.12}$$

The Ricean factor completely specifies the Ricean Distribution.

The cumulative effect of a large number of scattered wavefronts arriving at the receive antenna in the absence of a LOS component is equivalent to a complex narrow band noise process. In the presence of a LOS component it is equivalent to the random process of a sine wave in weaker narrowband noise. The discussion up to this point has been focused on the probability distribution function of the received signal envelope. The time dependence of this process is now characterized by means of the autocorrelation function, or equivalently the power spectral density of the received signal.

Gans, [108] developed a spectrum analysis for Clarke's model. He showed that for a $\lambda/4$ vertical receive antenna, and a uniformly distributed angle of arrival, the output spectrum is given by

$$S(f) = \frac{1.5}{\pi f_{d_{max}} \sqrt{1 - \left(\frac{f-f_c}{f_{d_{max}}}\right)^2}} \tag{3.13}$$

where $f_{d_{max}}$ is the maximum Doppler shift. This spectrum is plotted in Figure 3.4 below.

From this graph it can be seen that Doppler components arriving at exactly 0-degrees and 180-degrees have an infinite power spectral density. This is not a problem as the angle of arrival, α, is a continuous random variable, and thus the probability of a specific α is zero.

3.3.3 Impulse Response Model of a Multipath Channel

The various reflections of the original transmitted signal which impinge on the receive antenna can be seen as having individual delays, depending on the respective path lengths that were traveled by each reflection. This means that the phase of each multipath differs according to the path length and the RF carrier frequency. Each path has its own attenuation and delay, which leads to the modelling of the channel as a finite impulse response filter with time variant gains, where the time variation is due to receiver motion. The impulse response model is a wideband characterization of the channel as it contains all the information necessary to simulate or analyze any type of radio transmission through the channel.

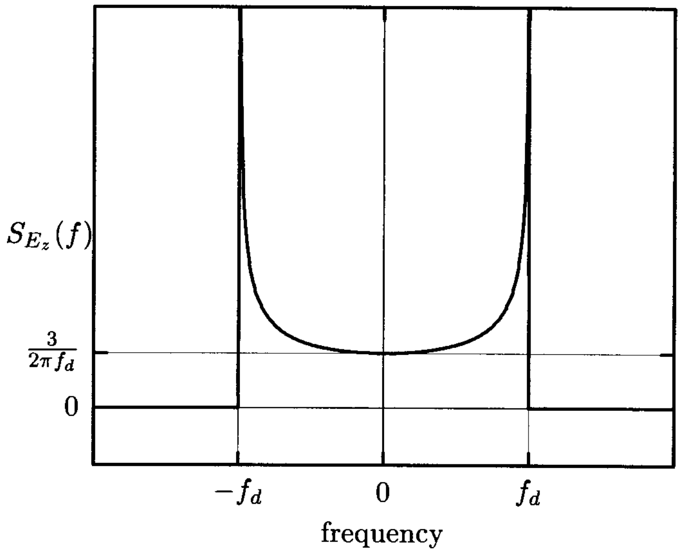

Figure 3.4: Theoretical Doppler spectrum

To show that the channel may be modelled as a linear filter with a time varying impulse response, consider the case where time variation is strictly due to receiver motion in space. It is assumed that the velocity of the receiver, v, is constant, and that the receiver motion is in the ground $(X - Y)$ plane. For a fixed position d, the channel can be modelled as a linear time-invariant filter in which all parameters are static, and it is assumed that the surrounding environment is stationary. However, the receiver motion causes it to move move through a set of spatial locations, each having a set of mutipaths with different delays and attenuations. From this argument it can be seen that the linear time invariant channel should be a function of the position of the receiver. That is the channel impulse response can be expressed as $h(d,t)$. Let $x(t)$ represent the transmitted signal, then the received signal, at a position d, is denoted by $y(d,t)$, which is given by the following convolution

$$\begin{aligned} y(d,t) &= x(t) \otimes h(d,t) && (3.14)\\ &= \int_{-\infty}^{\infty} x(\tau)h(d,t-\tau)d\tau && (3.15) \end{aligned}$$

Since the velocity of the receiver is constant, the position of the receiver is given by

$$d = vt \tag{3.16}$$

which by substitution into (3.15) leads to

$$y(vt,t) = \int_{-\infty}^{\infty} x(\tau)h(vt,t-\tau)d\tau \tag{3.17}$$

Since v is constant, (3.18) is only a function of t. Therefore (3.18) can be expressed as

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(vt,t-\tau)d\tau \tag{3.18}$$

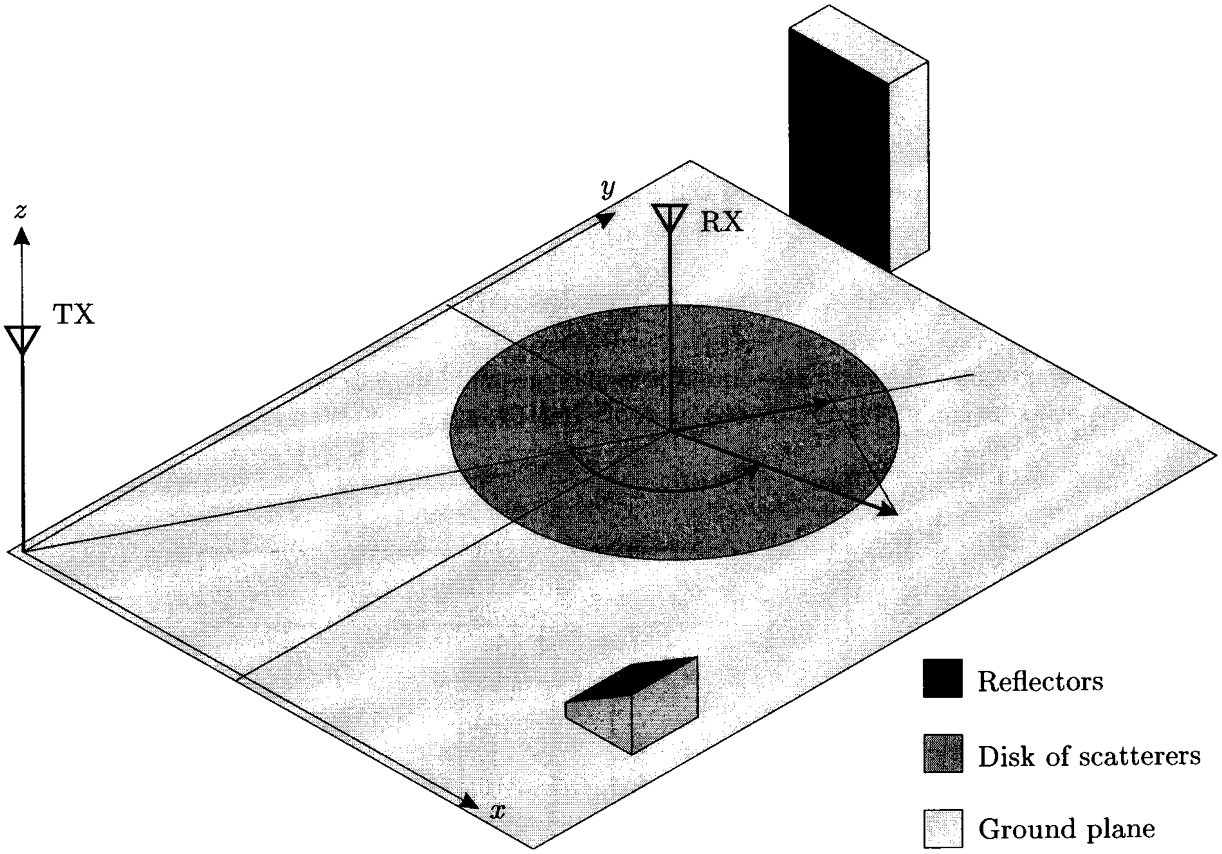

Figure 3.5: Receiver motion with ground plane and close-in disk of scatterers as well as reflectors which cause discrete multipaths

The impulse response of the channel is also independent of the velocity, and can be written as $h(t, \tau)$ which is the time varying impulse response of the multipath channel. The impulse response $h(t, \tau)$ completely characterizes the channel, and is a function of both t and τ. The variable t represents the time variations due to motion, and τ represents the channel multipath delay for a fixed value of t. The received signal is finally given by

$$\begin{aligned} y(t) &= x(t) \otimes h(t, \tau) && (3.19) \\ &= \int_{-\infty}^{\infty} x(\tau) h(t, \tau) d\tau && (3.20) \end{aligned}$$

which is the desired representation of the mobile wireless radio channel as a time variant linear system.

Figure 3.5 is a schematic representation of all the effects discussed thus far in this section. Reflectors which are relatively far from the receiving antenna are responsible for causing multipath echos, each having a different time delay, depending on the respective path lengths of each echo.

3.4 SIMULATION of the CLARKE CHANNEL MODEL WITH MULTIPATH

This section describes the software implementation of a fading multipath channel simulator for the models described in the preceding section. The approach in the design of this simulation is to model the theoretical fading mobile channel as closely as possible, but with minimal complexity so as not to be a processing burden in the total simulation environment.

3.4.1 Overview of Channel Simulation

The block diagram of the channel simulator is shown in Figure 3.6. This is the basic model as described by Opperman [48], which has its roots in [109, 110].

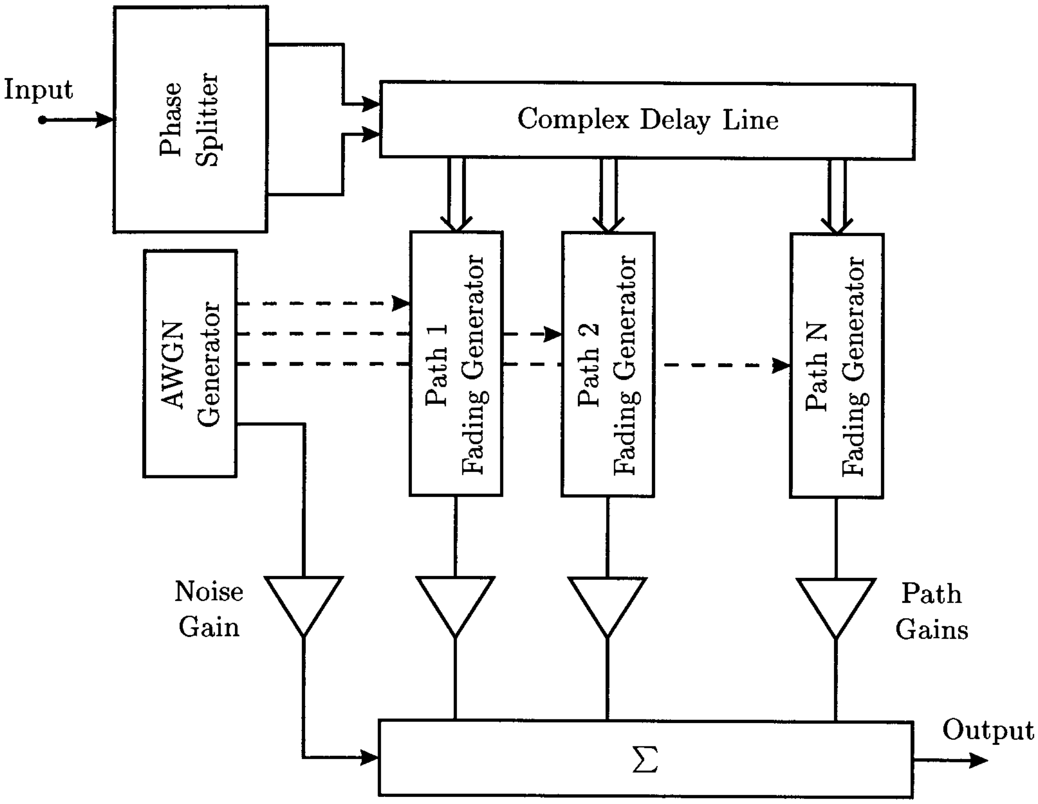

Figure 3.6: Schematic representation of channel simulator

The difference between this block diagram and others in the open literature is that use is made of only one noise generator. It is known that the noise generation method used, which is discussed further on, generates independent samples from a Gaussian (Normal) distribution. The samples are extracted from the generator as they are needed by the simulation components. This approach eliminates the need to start several random generators, each with a different seed, every time a new simulation is run. Thus only one seed is necessary to re-initialize the whole simulator.

3.4.2 Phase Splitter

A real signal, $x_R(t)$ can be represented as the real part of a complex signal, $\underline{x}(t)$, which is referred to as the "pre-envelope" or "analytic signal" of $x_R(t)$. The analytic signal can be expressed in terms of the real signal as

$$\underline{x}(t) = x_R(t) + j\hat{x}_R(t) \tag{3.21}$$

where $\hat{x}_R(t)$ is the Hilbert transform of $x_R(t)$. Use can thus be made of a Hilbert transformer to form an analytical signal from the real signal which enters the simulator.

Design of Hilbert Transformer

The Hilbert transformer was implemented as an finite impulse response (FIR) filter. This filter is responsible for the generation of the the quadrature (imaginary) component, and the center of its delay line was used to obtain the in-phase component. This setup is illustrated in Figure 3.8.

The design of the Hilbert filter had two goals

- Widest possible bandwidth and
- Lowest Possible ripple in the pass band.

This was achieved by making use of a 123 tap FIR Hilbert filter, which was windowed with a Kaiser window. A FIR filter was chosen as it has linear phase for a symmetric set of coefficients.

The impulse response of the ideal Hilbert FIR filter is given in [111] as

$$h[n] = \begin{cases} \dfrac{2}{\pi}\dfrac{\sin^2(\pi n/2)}{n}, & n \neq 0 \\ 0, & n = 0 \end{cases} \tag{3.22}$$

This impulse response extends from $-\infty$ to ∞, so it has to be truncated in practical applications, in this case to 123 taps. Equation (3.22) can also be written as

$$h[n] = \begin{cases} \dfrac{2}{n\pi} & n \text{ uneven} \\ 0, & \text{otherwise} \end{cases} \tag{3.23}$$

The impulse response corresponding to this equation is shown in Figure 3.7.

As the filter coefficients are symmetric in absolute value around the centre coefficient of the filter, the topology of Figure 3.8 can be used to reduce the number of multipliers necessary for the filter by half, and thus achieve a corresponding increase in simulation speed.

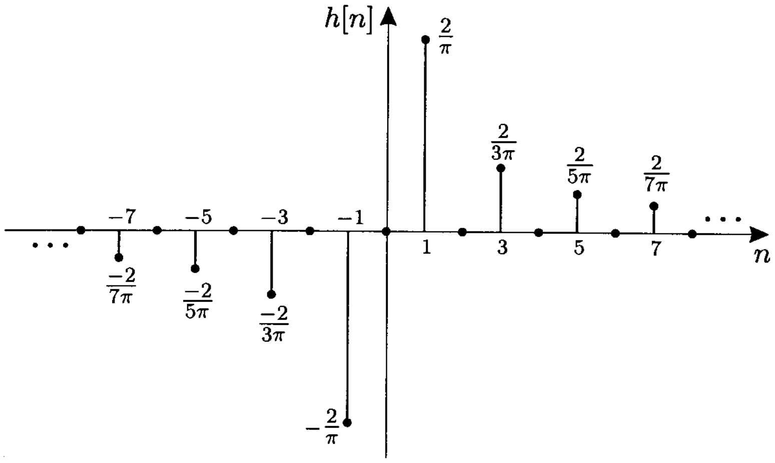

Figure 3.7: Impulse response of the theoretical Hilbert transformer

The Kaiser Window

The windowing function design problem for a linear phase FIR filter is a tradeoff between the mainlobe width, and sidelobe area of the frequency response of the windowing function. This problem can be solved by seeking the window function which is maximally concentrated around $\omega = 0$ in the frequency domain. This issue was considered in depth in a series of classic papers by Slepian et al. [91]. The solution presented by these authors involves the prolate spheroidal wave functions (PSWF), which are difficult to generate. Kaiser [112] found that a near optimal window can be formed by using the zeroth-order modified Bessel function of the first kind, which

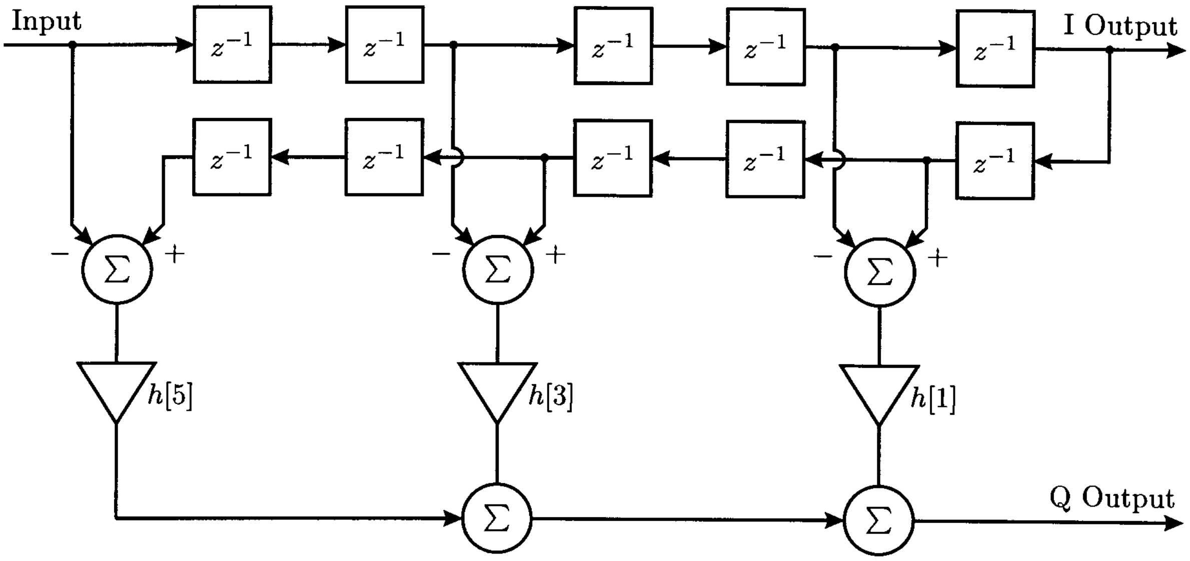

Figure 3.8: Schematic representation of Hilbert transformer. (Simplified "folded" version)

is easier to generate. The Kaiser window is defined as follows [111]

$$w[n] = \begin{cases} \dfrac{I_0\left[\beta\sqrt{1-(\frac{n-\alpha}{\alpha})^2}\right]}{I_0(\beta)} & 0 \leq n \leq M \\ 0, & \text{otherwise} \end{cases} \tag{3.24}$$

where $\alpha = M/2$, and $I_0(\cdot)$ is the zeroth-order modified Bessel function of the first kind. By varying the length, $M+1$, and the shape parameter β of the Kaiser window function sidelobe amplitude can be traded for mainlobe width. As applied to the Hilbert filter, the length of the window is fixed by letting $M = 122$. The shape parameter β is then the only parameter which has to be set.

The shape parameter was chosen as $\beta = 20$ for an intermediate trade-off between the bandwidth and passband ripple of the Hilbert filter. The frequency response of the Kaiser windowed Hilbert filter is shown in Figure 3.9, as well as a zoom view of the passband ripple which is labelled on the right hand y axis. These results were achieved with the use of 31 multipliers in the Hilbert filter.

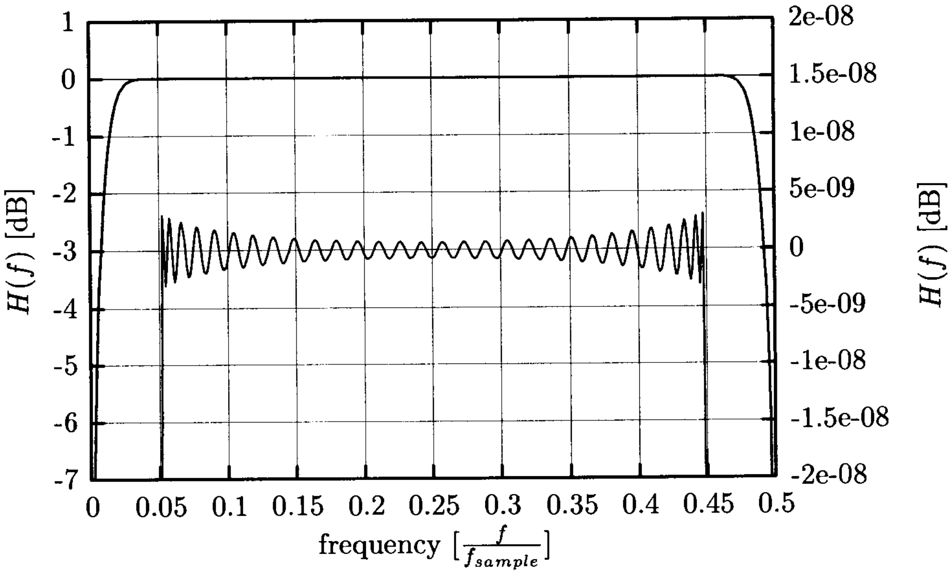

Figure 3.9: Hilbert transformer frequency response

According to [111] the most successful way of designing a phase splitter is to make use of two allpass systems, but for communication signals we will not consider this approach as it will introduce an unnecessary computational burden, for very little additional signal fidelity.

3.4.3 Doppler Offset

To obtain a Doppler offset for any path, use is made of a single sideband modulation technique. If the input signal to the Hilbert transformer is a bandpass function given by

$$x(t) = x_c(t)\cos(2\pi f_c t) - x_s(t)\sin(2\pi f_c t) \tag{3.25}$$

where $x_c(t)$ is the inphase modulation component, and $x_c(t)$ is the quadrature modulation component. The Hilbert transform of $x(t)$ is given by

$$\hat{x}(t) = x_c(t)\sin(2\pi f_c t) + x_s(t)\cos(2\pi f_c t) \tag{3.26}$$

A Doppler offset of f_d Hz can be achieved by the following modulation

$$\begin{aligned} x'(t) &= x(t)\cos(2\pi f_d t) + \hat{x}(t)\sin(2\pi f_d t) && (3.27)\\ &= x_c(t)\cos(2\pi(f_c + fd)t) - x_s(t)\sin(2\pi(f_c + fd)t) && (3.28)\end{aligned}$$

or alternatively in complex notation

$$\begin{aligned} x'(t) &= \mathrm{Re}\left\{(x_c(t) + jx_s(t))e^{-j2\pi f_c t}e^{-j2\pi f_d t}\right\} && (3.29)\\ &= \mathrm{Re}\left\{(x_c(t) + jx_s(t))e^{-j2\pi(f_c+f_d)t}\right\} && (3.30)\\ &= x_c(t)\cos(2\pi(f_c + f_d)t) - x_s(t)\sin(2\pi(f_c + f_d)t) && (3.31)\\ & && (3.32)\end{aligned}$$

From this equation it can be seen that the carrier frequency has been shifted from f_c to $f_c + f_d$, which is the desired result. Figure 3.10 shows the implementation of the Doppler offset generation circuit.

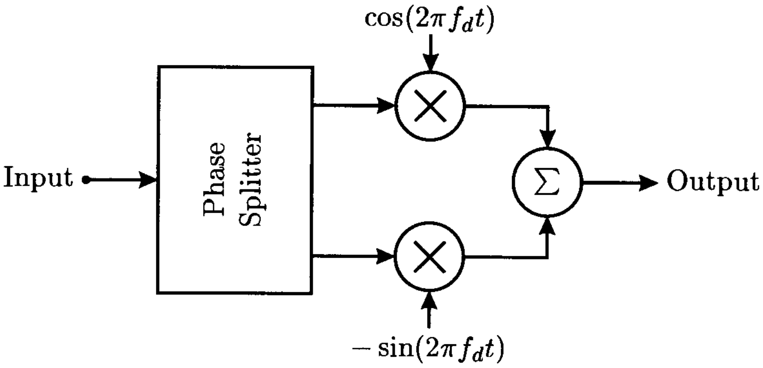

Figure 3.10: Doppler offset generation

3.4.4 Gaussian Noise Generation

To generate white Gaussian noise, use was made of a uniform random number generator, whose output samples are then mapped by means of a RV transformation to a Normal distribution. The algorithm used to generate uniformly distributed random numbers in the range $[0, 1]$ was published by Wichmann and Hill in [113]. This generator makes use of three linear congruential generators which are combined to break up any correlation which exists between consecutive samples. The period of this generator is 10^{12}. The Box-Muller algorithm was then applied to transform the uniformly distributed RV to one having a Normal distribution [114].

3.4.5 Doppler Spread Generation

Baseband infinite impulse response filters (IIR) are used to simulate the Doppler power spectrum. Figure 3.11 shows a block diagram of the combination of the Doppler power spectrum generation process and the Doppler offset generation process.

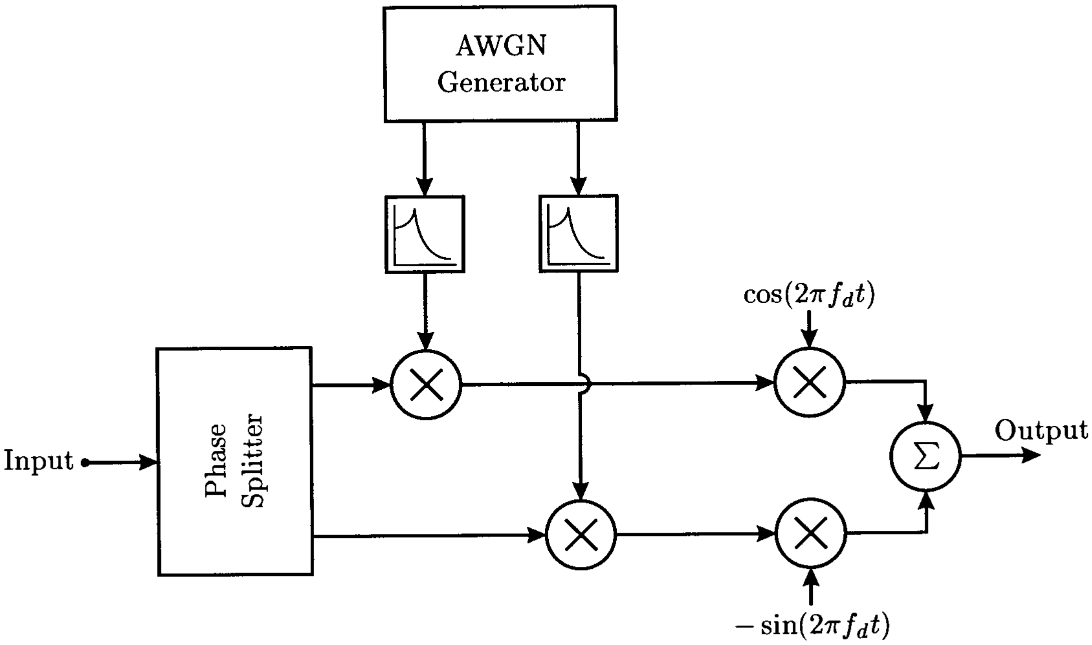

Figure 3.11: Doppler spread generation

Independent samples of Gaussian distributed white noise are used as the inputs to the two Doppler filters. The filters are designed to approximate the square root of the Doppler power spectrum. This can easily be seen by examining the relationship between the input and output power spectra of a linear time-invariant filter which is used to filter a random process :

$$\Phi_{yy}(e^{j\omega}) = \left|H(e^{j\omega})\right|^2 \Phi_{xx}(e^{j\omega}) \tag{3.33}$$

where $\Phi_{yy}(e^{j\omega})$ and $\Phi_{xx}(e^{j\omega})$ are the Fourier transforms of the output autocorrelation sequence, $\phi_{yy}[m]$, and the input autocorrelation sequence, $\phi_{xx}[m]$ respectively. $H(e^{j\omega})$ is the frequency response of the filter under consideration. White Gaussian noise has a flat power density spectrum, so

$$\Phi_{xx}(e^{j\omega}) = \frac{N_0}{2} = \sigma_x^2 \tag{3.34}$$

where σ_x is the standard deviation of the noise. Equation (3.33) reduces to

$$\Phi_{yy}(e^{j\omega}) = \left|H(e^{j\omega})\right|^2 \sigma_x^2 \tag{3.35}$$

The output spectrum must be equal to equation (3.13), so the frequency response of the Doppler shaping filters must be given by

$$H(f) = \sqrt{\frac{1.5}{\pi f_{d_{max}} \sqrt{1 - \left(\frac{f-f_c}{f_{d_{max}}}\right)^2}}} \tag{3.36}$$

Due to the abrupt transitions in this frequency response it is obvious that the filter will have to be an approximation to (3.36). The simulated Doppler frequency at the output of the Doppler generation process is given by

$$\hat{f}_d = \frac{1}{2\pi}\sqrt{\frac{2\hat{b}_2}{\hat{b}_0}} \tag{3.37}$$

where $\hat{f}_d$ is the simulated Doppler frequency, and $\hat{b}_n$ the nth moment of the spectrum of the filter. The moments are defined by

$$b_n = (2\pi)^n \int_0^{\infty} S(f) f^n df \tag{3.38}$$

Several approaches have been made to this problem by other authors. Smith [115] proposed a method which is based on implementing equation (3.36) in the frequency domain, and then making use of an inverse FFT to generate a fading signal. This method suffers from discontinuities between successive IFFT blocks, as well as problems at very low Doppler spreads. F. Swarts [116] and J. Swarts [49] made use of a 3rd order IIR digital filter given by

$$\begin{aligned} a_0 y[n] &= b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + b_3 x[n-3] \\ &\quad -a_1 y[n-1] - a_2 y[n-2] - a_3 y[n-3] \end{aligned} \tag{3.39}$$

where

$$\begin{aligned} b_0 &= b_3 = \omega_c^3 T^3 \\ b_1 &= b_2 = \omega_c^3 T^3 \\ a_0 &= 8 + 4A\omega_c T + 2B\omega_c^2 T^2 + C\omega_c^3 T^3 \\ a_1 &= -24 + 4A\omega_c T + 2B\omega_c^2 T^2 + 3C\omega_c^3 T^3 \\ a_2 &= 24 + 4A\omega_c T - 2B\omega_c^2 T^2 + 3C\omega_c^3 T^3 \\ a_0 &= -8 + 4A\omega_c T - 2B\omega_c^2 T^2 + C\omega_c^3 T^3 \\ A &= 1.55 \\ B &= 1.090625 \\ C &= 0.9953125 \end{aligned}$$

T is the sampling period, and ω_c is the cutoff frequency of the filter which is set equal to the maximum Doppler shift. This is a good approximation for larger Doppler spread values, but the design equations give unstable poles for low Doppler spread channels in the tens of m/s and lower ranges. Opperman [48] made use of 8 cascaded 2nd order Butterworth filter sections, but these filters also become unstable in the lower Doppler spread ranges. The approximation is not better than that of Swarts, even though a much higher order filter is used. Haeb [117] implemented a very accurate 8th order IIR filter. The zeros and poles of this filter are given in Table 3.2. This filter is designed for a very high Doppler spread, and linear interpolation is then used to scale the Doppler frequency to a lower value.

The interpolation can only take on integer values, so only a specific set of receiver speeds can be simulated. The linear interpolation also changes the frequency response of the filter very slightly.

The frequency responses of the various Doppler filters are plotted in Figure 3.12 for comparative purposes. The doppler frequency was set to $f_d/f_s = 0.009049551$, which is the theoretical Doppler frequency for the 8th order filter by Haeb, for all the filter designs. The theoretical filter response has also been plotted up to a point just before the Doppler frequency.

Poles	Zeros
$p1 = 0.99015456438065 \pm j0.04500919952989$	$z1 = 0.99835836887360 \pm j0.05727641656995$
$p2 = 0.98048448562622 \pm j0.01875760592520$	$z2 = 0.99744373559952 \pm j0.07145611196756$
$p3 = 0.99652880430222 \pm j0.05493839457631$	$z3 = 0.99440407752991 \pm j0.10564350336790$
$p4 = 0.99827980995178 \pm j0.05666938796639$	$z4 = 0.96530824899673 \pm j0.26111298799515$

Table 3.2: Poles and zeros for 8th order Doppler filter by Haeb.

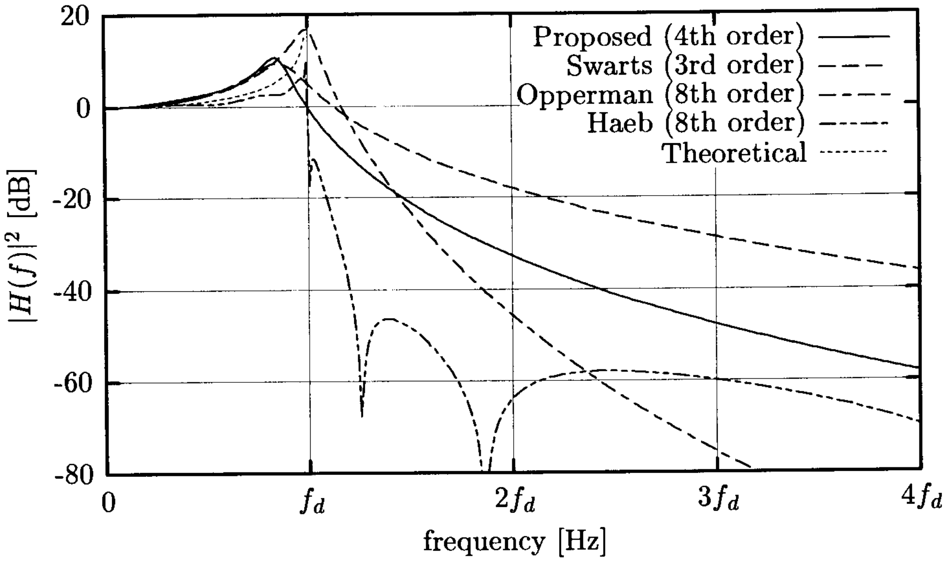

Figure 3.12: Frequency response of various Doppler filters.

Doppler Filter Design

After much experimentation it was determined that a 4th order IIR filter can approximate the Doppler spread PSD relatively accurately. The block diagram of this filter is shown in Figure 3.13, where the filter has been implemented as two cascaded second-order sections for improved numerical stability.

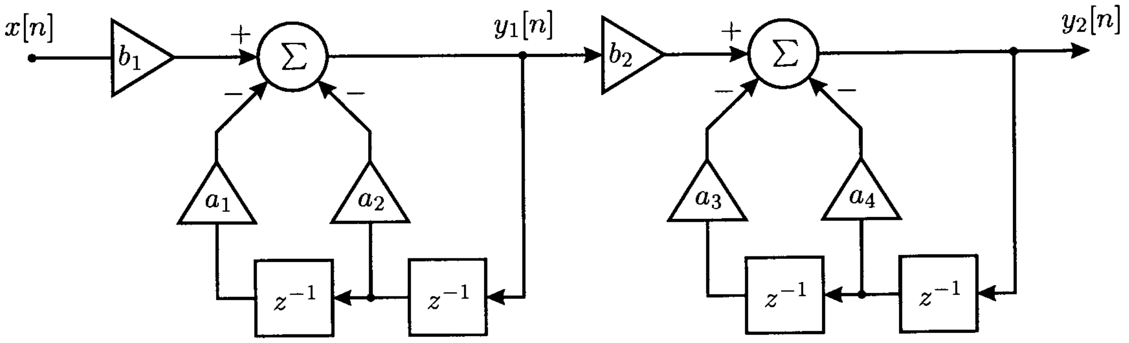

Figure 3.13: Doppler filter schematic

This filter is described by the following iterative discrete time equations

$$y_1[n] = b_1 x[n] - a_1 y_1[n-1] - a_2 y_1[n-2] \tag{3.40}$$

$$y_2[n] = b_2 y_1[n] - a_3 y_2[n-1] - a_4 y_2[n-2] \tag{3.41}$$

The b_2 coefficient can be set to one and its value incorporated in b_1 if it is necessary to save one multiplier. Two b coefficients were used to limit the dynamic ranges within each filter section.

The design equations for the filters are not included at this point due to patent considerations.

Filter Power Calibration

For the fading simulator it is necessary that the variance of the output of the Doppler filters be unity. It can easily be shown that relationship between the input power and output power of a digital filter is given in terms of its impulse response as

$$\sigma_{out}^2 = \sum_{n=0}^{\infty} h^2[n]\sigma_{in}^2 \tag{3.42}$$

The Gaussian noise generator has a unity output variance, so $\sigma_{in}^2 = 1$. It is now necessary to determine the sum of the square of the impulse response. Most authors use an approximate method to achieve this sum. The filter is driven with an impulse, followed by zeros and the output of the filter is squared and accumulated [48, 118]. For this method to be accurate the sum should span 100 to 1000 times the inverse of the bandwidth of the filter, in time, which wastes valuable computing resources. This problem can be solved analytically by making use of Parseval's relation, and complex analysis methods. Parseval's relation states that

$$\sum_{n=-\infty}^{\infty} x_1[n]x_2^*[n] = \frac{1}{2\pi j}\oint_C X_1(v)X_2^*(1/v^*)v^{-1}dv \tag{3.43}$$

for two complex sequences $x_1[n]$ and $x_2[n]$. A special case of this equation arises when $x_1[n] = x_2[n] = x[n]$, where $x[n]$ is a real sequence :

$$\sum_{n=-\infty}^{\infty} x^2[n] = \frac{1}{2\pi j}\oint_C X(v)X(v^{-1})v^{-1}dv \tag{3.44}$$

where C is a closed contour in the Region of convergence (ROC) of $X(z)$. Equation (3.44) can now be evaluated straightforwardly using the Cauchy residue theorem. For an IIR filter of order N, but with all its zeros at the origin of the z-plane, and one input gain b, we now have

$$\sum_{n=-\infty}^{\infty} h^2[n] = \frac{1}{2\pi j}\oint_c H(z)H(z^{-1})z^{-1}dz \tag{3.45}$$

where

$$H(z) = \frac{bz^N}{\prod\limits_{i=1}^{N}(z-p_i)} \tag{3.46}$$

and

$$H(z^{-1}) = \frac{b}{\prod\limits_{i=1}^{N}(1-p_i z)} \tag{3.47}$$

which can be simplified as

$$\begin{aligned}
\sum_{n=-\infty}^{\infty} h^2[n] &= \frac{1}{2\pi j}\oint_c \frac{b^2 z^{N-1}}{\prod\limits_{i=1}^{N}(z-p_i)\prod\limits_{i=1}^{N}(1-p_i z)}dz \\
&= \sum_{i=1}^{N}\lim_{z\to p_i}\frac{b^2 z^{N-1}}{\prod\limits_{\substack{j=1\\ j\neq i}}^{N}(z-p_j)\prod\limits_{j=1}^{N}(1-p_j z)} \\
&= \sum_{i=1}^{N}\frac{b^2 p_i^{N-1}}{\prod\limits_{\substack{j=1\\ j\neq i}}^{N}(p_i-p_j)\prod\limits_{j=1}^{N}(1-p_j p_i)} \qquad (3.48)
\end{aligned}$$

This is the desired closed form expression for the sum of the squares of the impulse response of an IIR filter. In the derivation it has been assumed that the filter is stable implying that all the poles of the filter lie within the unit circle in the z plane. From this assumption C was chosen to lie on the unit circle, except for excursions to account for the singularities. Equation (3.48) can easily be implemented algorithmically.

Instantaneous Start-up of the Doppler Filters

Due to the extremely narrow bandwidth of some of the filters implemented, it was necessary to use the method presented by Steven Kay in [89] to eliminate the startup transients of the Doppler shaping filters.

The filter structure discussed above is used to generate noise with a given autocorrelation function, $Z[n]$, by filtering white noise, $u[k]$. The general discrete-time recursive equation for an IIR filter is given by

$$Z[n] = -\sum_{l=1}^{p} a_l Z[n-l] + \sum_{l=0}^{q} b_l u[n-l], \quad n \geq 0 \qquad (3.49)$$

where $q \leq p$. The above equation can be expressed in terms of the following two equations

$$\begin{aligned}
S[n] &= -\sum_{l=1}^{p} a_l S[n-l] + u[n] \\
Z[n] &= -\sum_{l=0}^{q} b_l S[n-l], \quad n \geq 0 \qquad (3.50)
\end{aligned}$$

$S[n]$ is thus the output of an all pole filter. It is necessary to generate values for

$$\mathbf{X}[-1] = [S[-1], S[-2], \cdots, S[-p]]^T \qquad (3.51)$$

It has been shown by Anderson and Moore [119] that if $\mathbf{X}[-1]$ is a zero mean vector, and if

$$E\{\mathbf{X}[-1]\mathbf{X}[-1]^T\} = \mathbf{R_S} \qquad (3.52)$$

where $[R_S]_{ij} = R_S(i-j)$ is the autocorrelation sequence of $S[n]$, then $S[n]$ will be a wide sense stationary process. We now seek a linear transformation which will yield a vector $\mathbf{V}$

of independent zero mean unit variance random variables to have the given autocorrelation sequence, R_S. The linear transformation is given by

$$\mathbf{X}[-1] = \mathbf{AV} \tag{3.53}$$

where $\mathbf{A}$ is chosen so that

$$E\{\mathbf{X}[-1]\mathbf{X}[-1]^T\} = \mathbf{R_S} \tag{3.54}$$

which can be expanded as follows

$$\begin{aligned} \mathbf{R_S} &= E\{\mathbf{AV}(\mathbf{AV})^T\} \\ &= \mathbf{AIA}^T \\ &= \mathbf{AA}^T \end{aligned} \tag{3.55}$$

Thus it is only necessary to perform a square rooting of the Toeplitz covariance matrix $\mathbf{R}_S$ which can be achieved by means of the Levinson-Durban algorithm. It can be shown that [120]

$$\mathbf{B}^T\mathbf{R}_S\mathbf{B} = \mathbf{P} \tag{3.56}$$

where

$$\mathbf{B} = \begin{bmatrix} 1 & a_1^{(1)} & a_2^{(2)} & \cdots & a_{p-1}^{(p-1)} \\ 0 & 1 & a_1^{(-2)} & \cdots & a_{p-2}^{(p-1)} \\ & & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{3.57}$$

$$\mathbf{P} = \mathrm{diag}(P_0, P_1, \cdots, P_{p-1}) \tag{3.58}$$

and $a_i^{(j)}$ is the ith coefficient of the jth prediction error filter. The prediction error power of the jth order prediction error filter is denoted by P_j. Note that the original feedback filter coefficients, a_i are equal to $a_i^{(p)}$, which would be an extra column to the right of the matrix in (3.57). By solving (3.56) for $\mathbf{R}_S$ we now have

$$\mathbf{R_S} = \mathbf{B}^{T^{-1}}\mathbf{PB}^{-1} \tag{3.59}$$

Defining

$$\sqrt{\mathbf{P}} = \mathrm{diag}(\sqrt{P_0}, \sqrt{P_1}, \cdots, \sqrt{P_{p-1}}) \tag{3.60}$$

$\mathbf{A}$ can now be expressed as

$$\mathbf{A} = \mathbf{B}^{T^{-1}}\sqrt{\mathbf{P}} \tag{3.61}$$

The matrix $\mathbf{B}^T$ is always non-singular as it can be shown that

$$|\mathbf{B}^T| = 1 \tag{3.62}$$

By solving for $\mathbf{X}[-1]$

$$\mathbf{X}[-1] = \mathbf{B}^{T^{-1}}\sqrt{\mathbf{P}}\,V \tag{3.63}$$

or alternatively

$$\left(\sqrt{\mathbf{P}}\right)^{-1}\mathbf{B}^T\mathbf{X}[-1] = \mathbf{V} \tag{3.64}$$

Once again it has been assumed that the filter is stable, which implies that all the poles of the filter lie within the unit circle in the z plane. This in turn implies that

$$\mathbf{P}_i \neq 0 \qquad i = 0, 1, \cdots, p-1 \tag{3.65}$$

To find $\mathbf{X}[-1]$ it is necessary to solve the above set of linear equations. However, since $\left(\sqrt{\mathbf{P}}\right)^{-1}\mathbf{B}^T$ is a lower triangular matrix, $\mathbf{X}[-1]$ can be found efficiently. Firstly by evaluation of $\left(\sqrt{\mathbf{P}}\right)^{-1}\mathbf{B}^T$ we have

$$\left(\sqrt{\mathbf{P}}\right)^{-1}\mathbf{B}^T = \begin{bmatrix} \frac{1}{\sqrt{P_0}} & 0 & \cdots & 0 \\ \frac{a_1^{(1)}}{\sqrt{P_1}} & \frac{1}{\sqrt{P_1}} & \cdots & 0 \\ & \vdots & & \\ \frac{a_{p-1}^{(p-1)}}{\sqrt{P_{p-1}}} & \frac{a_{p-2}^{(p-1)}}{\sqrt{P_{p-1}}} & \cdots & \frac{1}{\sqrt{P_{p-1}}} \end{bmatrix} \tag{3.66}$$

Now solving for the elements of $\mathbf{X}[-1]$ recursively we have

$$S[-1] = \sqrt{P_0}\, V[1] \tag{3.67}$$

$$S[-k] = \sqrt{P_{k-1}}\, V[k] - \sum_{l=1}^{k-1} a_{k-l}^{(k-l)} S[-l], \quad k = 2, 3, \cdots, p \tag{3.68}$$

To evaluate the above recursive equation it is necessary to have evaluated $\mathbf{B}$ and $\mathbf{P}$. This can be achieved by making use of the ”step down” procedure [121, 122]. Firstly to find $\mathbf{B}$

$$a_i^{(j-1)} = \frac{a_i^{(j)} - k_j a_{j-i}^{(j)}}{1 - k_j^2}, \quad i = 1, 2, \cdots, j-1 \tag{3.69}$$

with

$$k_j = a_j^{(j)} \tag{3.70}$$

Secondly to find $\mathbf{P}$ we have

$$P_0 = \frac{\sigma_u^2}{\prod_{i=1}^{p}(1 - k_i^2)} \tag{3.71}$$

and for $j = 1, 2, \cdots, p-1$

$$P_j = P_{j-1}(1 - k_j^2) \tag{3.72}$$

Opperman [48] claims that the method described above can be used to eliminate the transients in IIR filters which are implemented as cascades of second order sections, but never goes as far as to give an exact method.

To apply this method to a cascaded filter topology such as the one under consideration for use as the Doppler shaping filter, it is first necessary to find the direct form I coefficients in terms of the cascaded filter coefficients. The discrete time difference equations for the filter are

$$\begin{aligned} y_1[n] &= b_1 x[n] - a_1 y_1[n-1] - a_2 y_1[n-2] && (3.73) \\ y_2[n] &= b_2 y_1[n] - a_3 y_2[n-1] - a_4 y_2[n-2] && (3.74) \end{aligned}$$

Now by solving (3.74) for $y_1[n]$

$$y_1[n] = \frac{1}{b_2}\left\{y_2[n] + a_3 y_2[n-1] + a_4 y_2[n-2]\right\} \tag{3.75}$$

and substituting this result into (3.73) we have

$$\begin{aligned}&\frac{1}{b_2}\left\{y_2[n]+a_3y_2[n-1]+a_4y_2[n-2]\right\}=\\&\quad b_1x[n]-\frac{a_1}{b_2}\left\{y_2[n-1]+a_3y_2[n-2]+a_4y_2[n-3]\right\}\\&\quad -\frac{a_2}{b_2}\left\{y_2[n-2]+a_3y_2[n-3]+a_4y_2[n-4]\right\}\end{aligned}\tag{3.76}$$

which can be simplified to give

$$\begin{aligned}y_2[n] \;=\;& b_1b_2x[n]\\&-(a_1+a_3)\,y_2[n-1]\\&-(a_2+a_1a_3+a_4)\,y_2[n-2]\\&-(a_1a_4+a_2a_3)\,y_2[n-3]\\&-(a_2a_4)\,y_2[n-4]\end{aligned}\tag{3.77}$$

where $y_2[n] = S[n]$. The coefficients in this expression are actually the convolution of the feedback coefficients of the two second order sections. This is the desired direct form I filter expression. From this equation it can be seen that

$$a_1^{(p)} = a_1+a_3 \tag{3.78}$$
$$a_2^{(p)} = a_2+a_1a_3+a_4 \tag{3.79}$$
$$a_3^{(p)} = a_1a_4+a_2a_3 \tag{3.80}$$
$$a_4^{(p)} = a_2a_4 \tag{3.81}$$

which are the initial values required in the determination of $\mathbf{B}$. Once the values for $\mathbf{X}[-1]$ have been determined, these values have to be mapped from the initial values of the direct form I memory elements to the initial values of the cascaded form. For the second section this is straight forward as

$$y_2[-1] = S[-1] \tag{3.82}$$
$$y_2[-2] = S[-2] \tag{3.83}$$

For the first section, to find the initial values for y_1 we make use of (3.75) to give

$$y_1[-1] = \frac{1}{b_2}\left\{S[-1]+a_3S[-2]+a_4S[-3]\right\} \tag{3.84}$$

and

$$y_1[-2] = \frac{1}{b_2}\left\{S[-2]+a_3S[-3]+a_4S[-4]\right\} \tag{3.85}$$

which concludes the fast startup algorithm for the Doppler spread filter.

It was found that this fast startup algorithm becomes unstable for Doppler filters with bandwidths under $10^{-4}f_{sample}$.

3.4.6 Multipath

Multipath effects, which are responsible for frequency selective fading, are simulated by shifting each complex output of the phase splitter into a complex delay line. At the time delay positions in the delay line where multipaths whish to be simulated, the whole doppler spread and Doppler shift section of Figure 3.11 is connected to the delay line. The output of this section is then weighted with the relative strength of the multipath. This setup is depicted schematically in Figure 3.6. To ensure that the input power remains equal to the output power, the relative strengths of the multipaths are normalized to ensure that the sum of their squares is unity.

3.4.7 Power Calibration

Due to the fact that the signal power was brought into consideration, and described mathematically throughout the design of the channel simulator, no special gains are necessary to ensure that the input signal power is equal to the output signal power. For modulated signals which have not been bandlimited at the transmitter there is a very small discrepancy between the input and output power of the channel simulator. This is due to the fact that use was only made of one filter in the phase splitter section, instead one on each branch, and the fact that the Hilbert filter is not an allpass filter.

3.5 CHANNEL SIMULATOR RESULTS

This section gives some examples of results obtained from the channel simulator. Results obtained are bench marked against theoretical values to check the accuracy of the simulation.

3.5.1 Fading Distribution Results

In this section the measured probability distribution function (PDF) of the output of the channel simulator is plotted against the theoretical Rayleigh and Rician PDF's for several Doppler frequencies. The simulation for each Doppler frequency was run for $N = 20000 f_d^{-1}$ samples, and the bin widths for the histogram were set to 0.1 Volt.

Rayleigh Distributed Envelope Results

The set of graphs presented in this section show the results for the Rayleigh fading case for Doppler frequencies from 10^{-2}, stepped lower in decades to 10^{-6}. Note that the percentage error is annotated on the right hand y-axis of each graph.

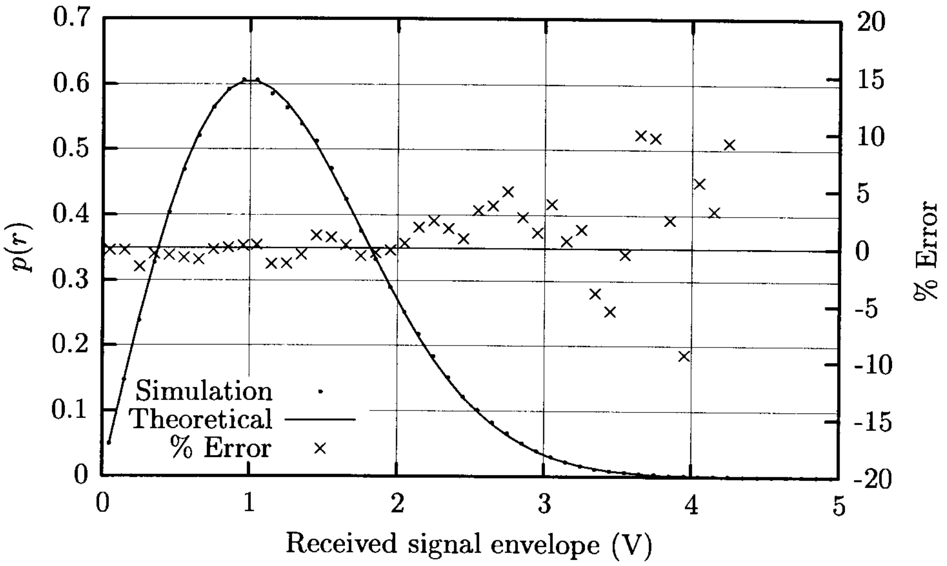

Figure 3.14: Rayleigh PDF for $f_d' = 10^{-2}$

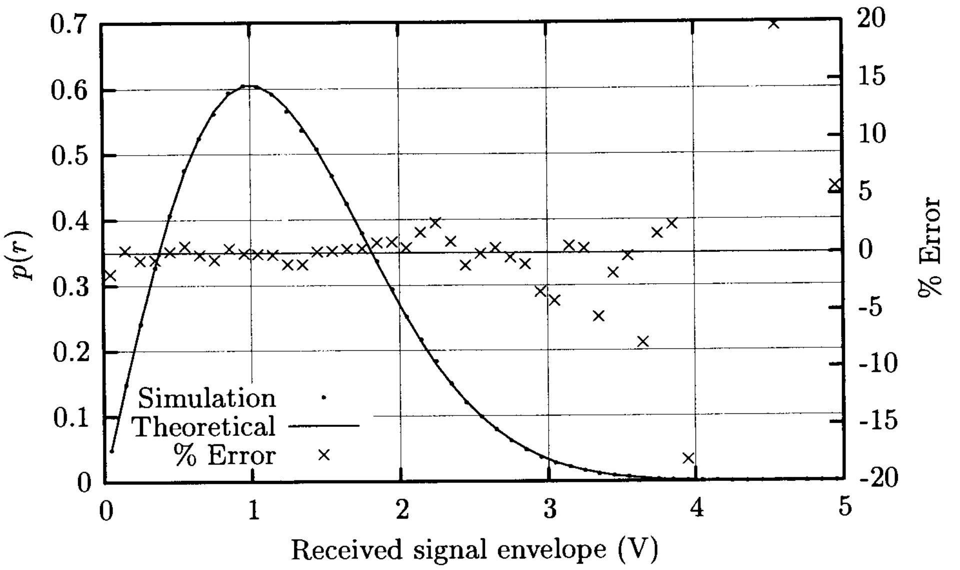

Figure 3.15: Rayleigh PDF for $f_d' = 10^{-3}$

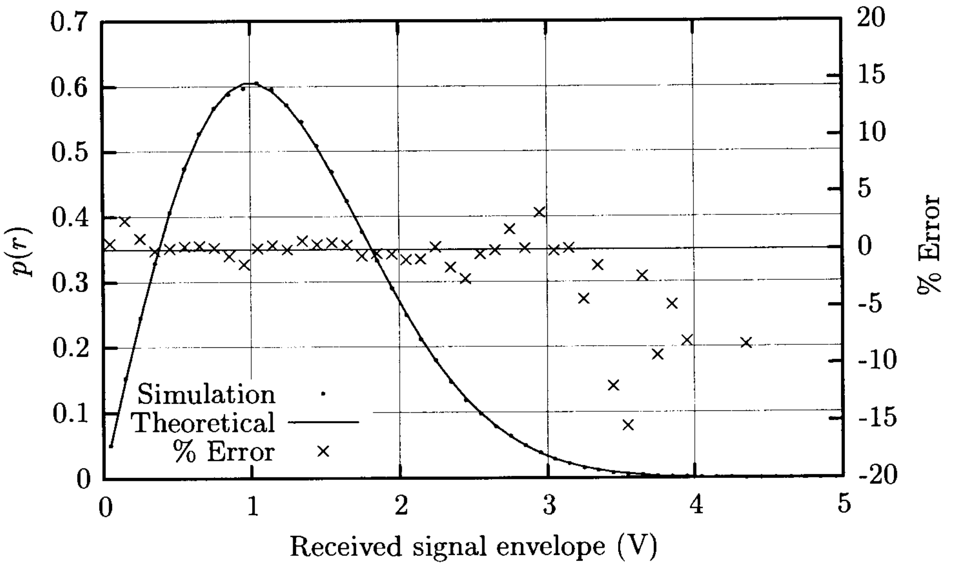

Figure 3.16: Rayleigh PDF for $f_d' = 10^{-4}$

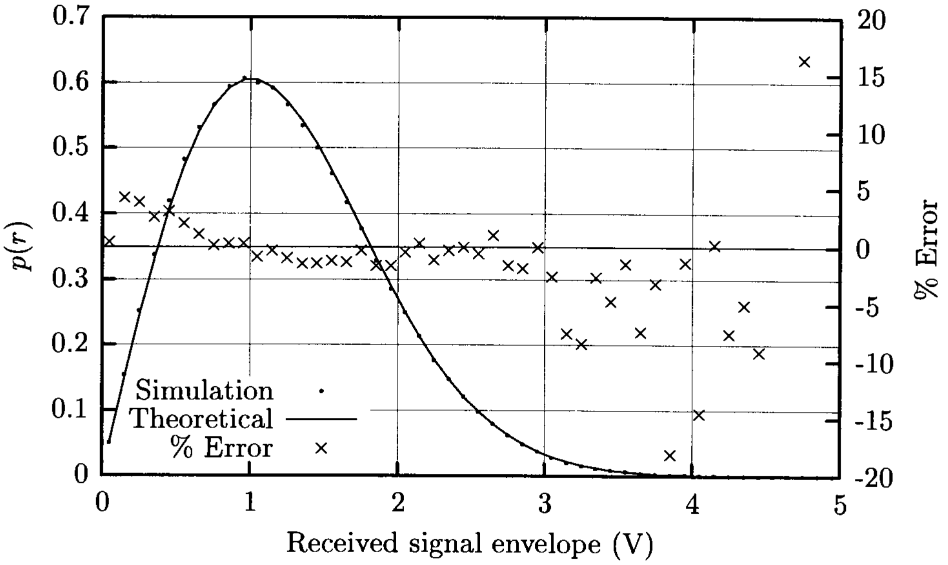

Figure 3.17: Rayleigh PDF for $f_d' = 10^{-5}$

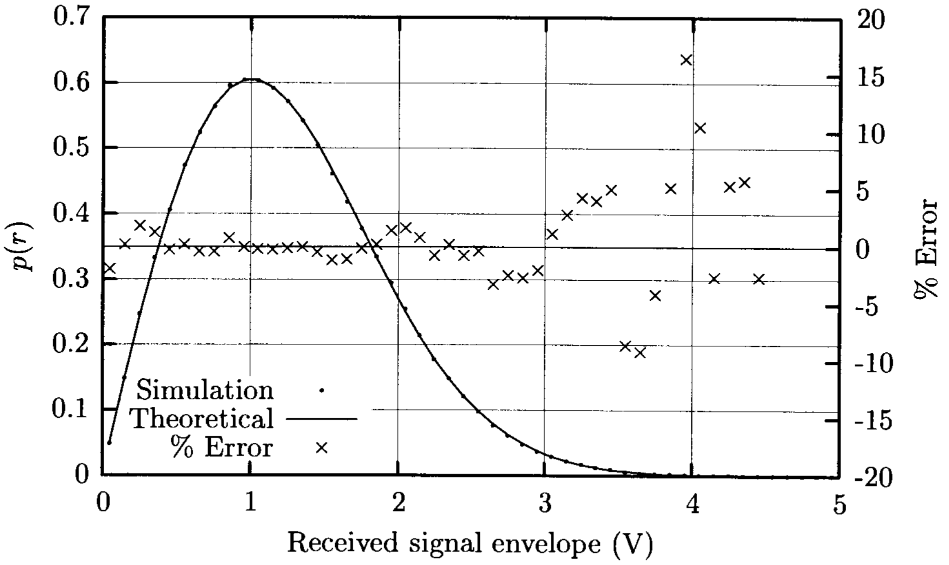

Figure 3.18: Rayleigh PDF for $f_d' = 10^{-6}$

From these results it can be seen that the Rayleigh PDF is approximated accurately over the whole range of frequencies tested. The loss in accuracy in the tail of the distribution is a measurement error caused by a lack of samples, due to insufficient simulation length, at these values in the test simulation.

Rice Distributed Envelope Results

The set of graphs presented in this section show the results for the Ricean fading case for Doppler frequencies from 10^{-2}, stepped lower in decades to 10^{-6}. Note that the percentage error is referenced to the right hand y-axis of each graph.

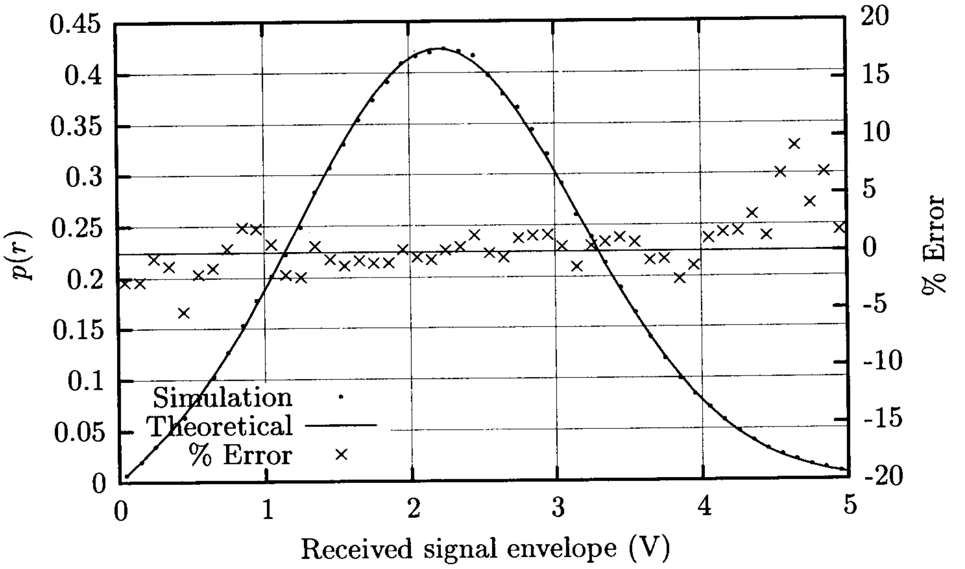

Figure 3.19: Ricean PDF for $f_d' = 10^{-2}$

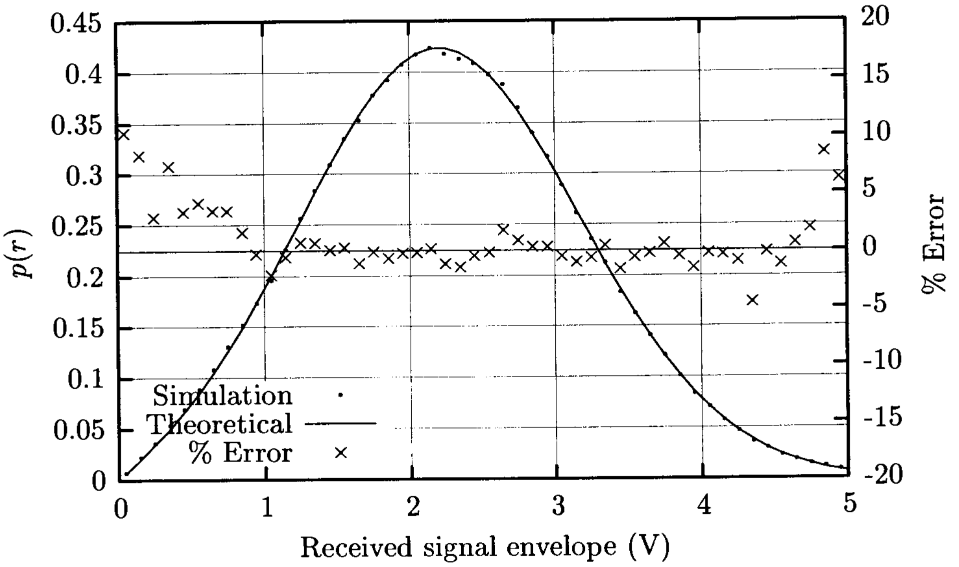

Figure 3.20: Ricean PDF for $f_d' = 10^{-3}$

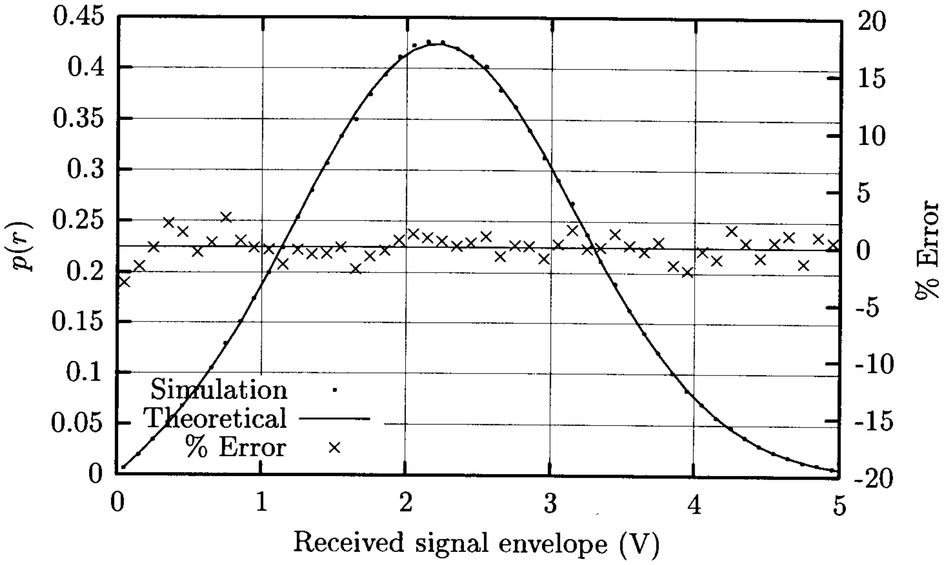

Figure 3.21: Ricean PDF for $f_d' = 10^{-4}$

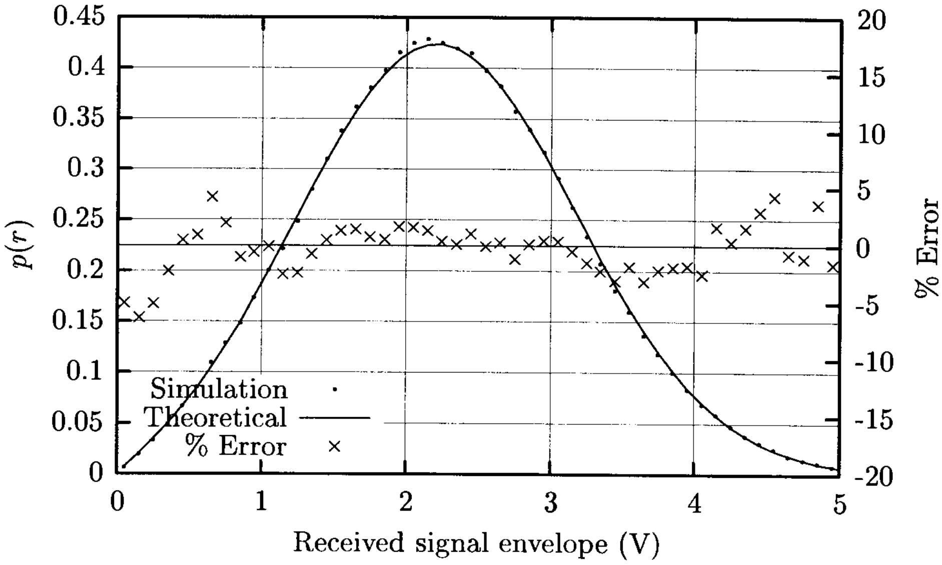

Figure 3.22: Ricean PDF for $f_d' = 10^{-5}$

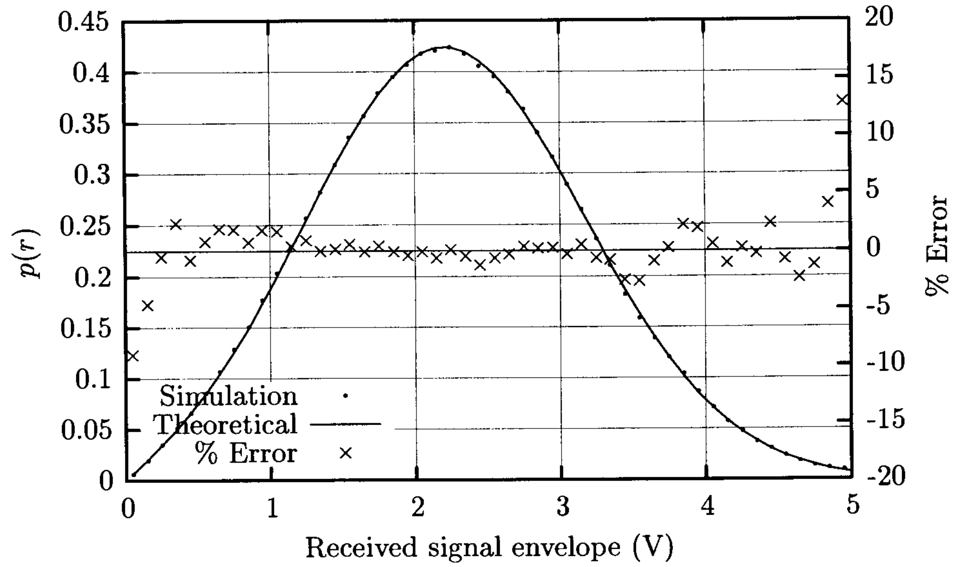

Figure 3.23: Ricean PDF for $f'_d = 10^{-6}$

From these results it can be seen that the Ricean PDF is approximated accurately over the whole range of frequencies tested. There is a slight loss in accuracy to either side of the main lobe of the distribution. This is a measurement error caused by a lack of samples, due to insufficient simulation length, at these values in the test simulation.

3.5.2 Level Crossing Rate Results

In this section the measured level crossing rate (LCR) of the envelope of the output of the channel simulator is plotted against the theoretical value of the LCR for several Doppler frequencies. The simulation run lengths are the same as for the previous section. Both the measured and the theoretical LCR values are normalized by the Doppler frequency.

The level crossing rate, which is defined as the expected rate at which the envelope, R, crosses a specified level, R_s, in the positive direction, is given by Gans [108] for the case of a vertical monopole antenna as follows :

$$N_{R_s} = \sqrt{2\pi}\, f_{d_{max}} \left(\frac{R_s}{\sqrt{2}\sigma}\right) \exp\left(-\frac{R_s^2}{2\sigma^2}\right) \tag{3.86}$$

The normalized LCR is given by N_{R_s}/f'_d, where $f'_d = f_{d_{max}}/f_{sample}$ is the maximum theoretical Doppler offset. The measured normalized LCR curves for the Rayleigh faded case, making use of the proposed 4th order Doppler filter for $f'_d = \{10^{-2}, 10^{-3}, 10^{-4}, \ldots, 10^{-7}\}$, are presented in the following set of figures. The theoretical normalized LCR curve and the percentage error between the theoretical and measured values are also plotted on the same system of axis. The error is plotted as a percentage of the theoretical value, and is labelled on the right hand y axis on each plot.

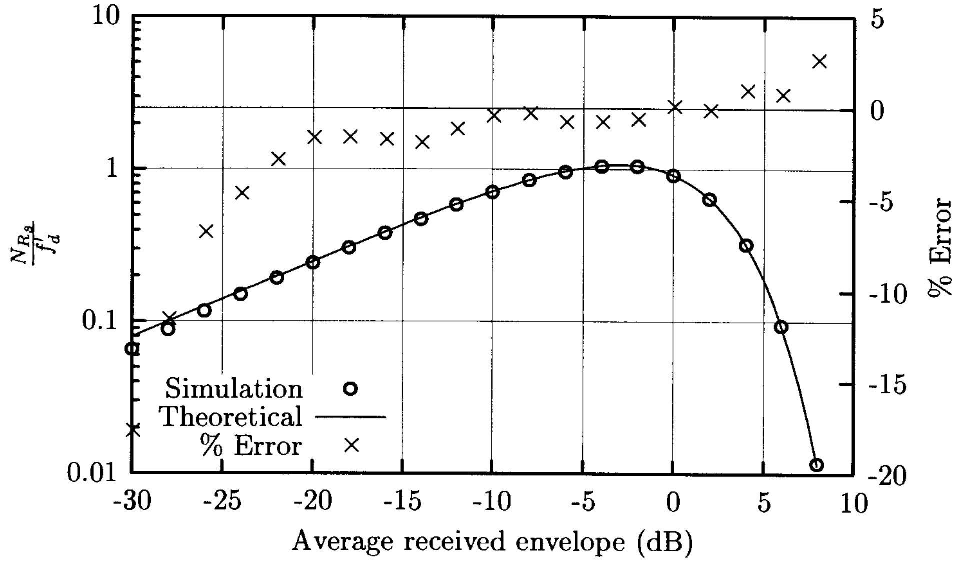

Figure 3.24: Normalized level crossing rate for $f'_d = 10^{-2}$

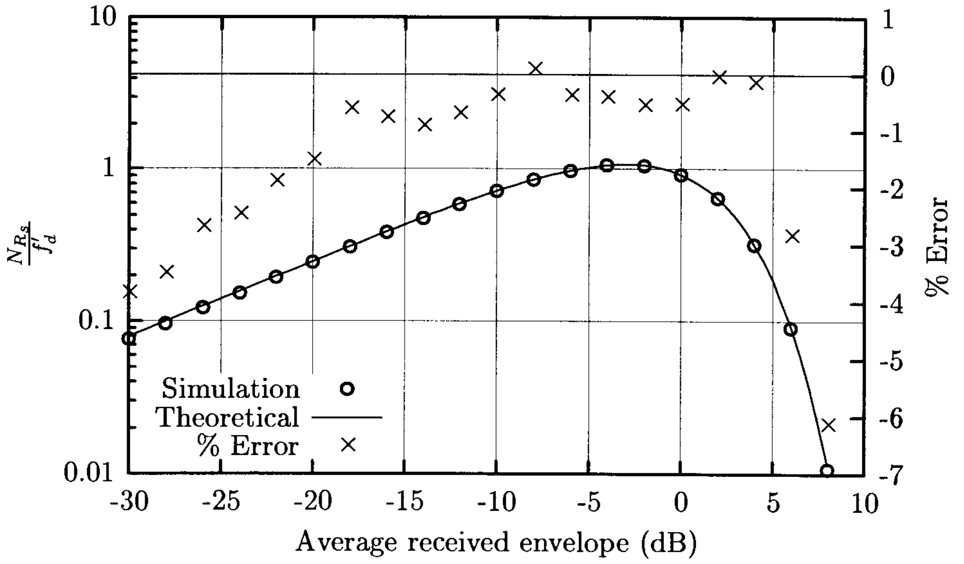

Figure 3.25: Normalized level crossing rate for $f'_d = 10^{-3}$

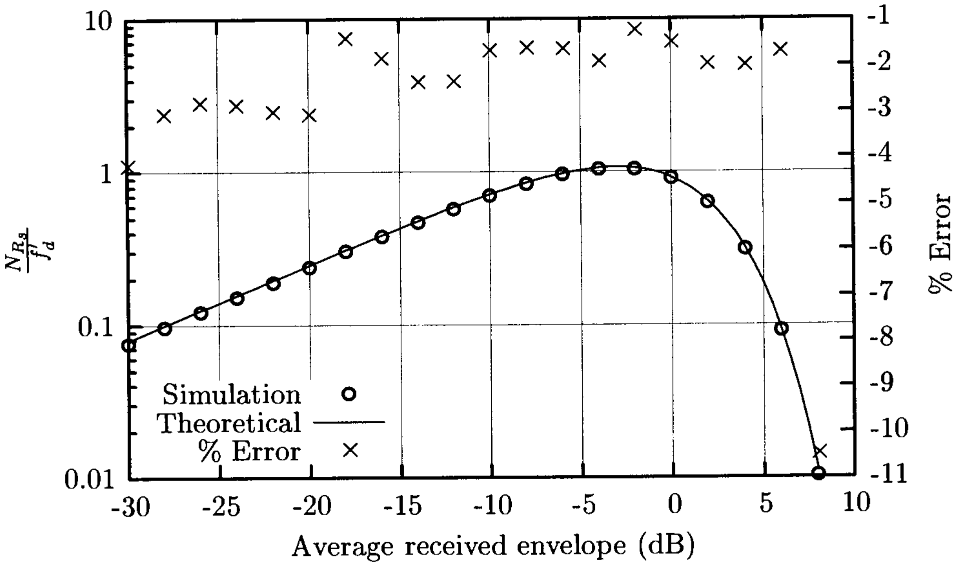

Figure 3.26: Normalized level crossing rate for $f'_d = 10^{-4}$

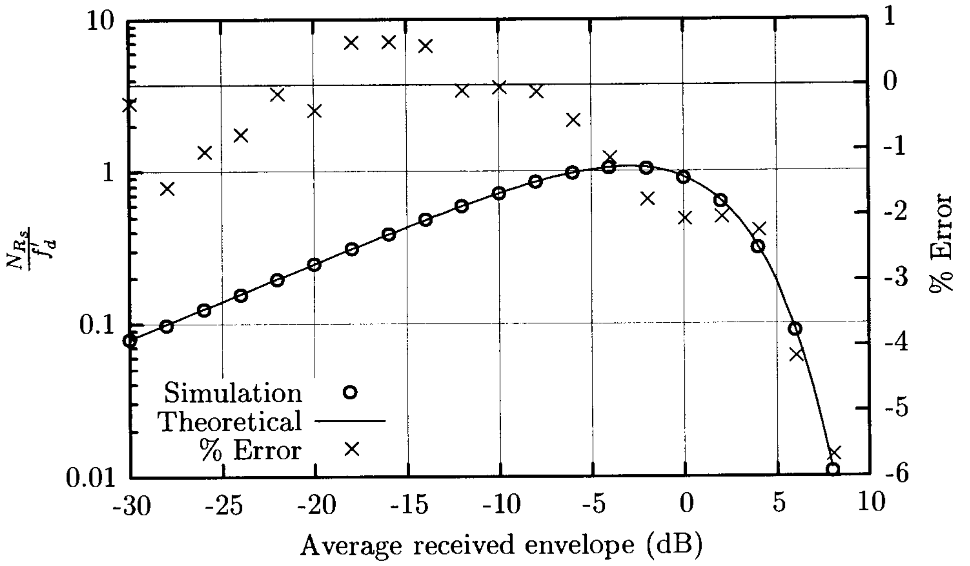

Figure 3.27: Normalized level crossing rate for $f'_d = 10^{-5}$

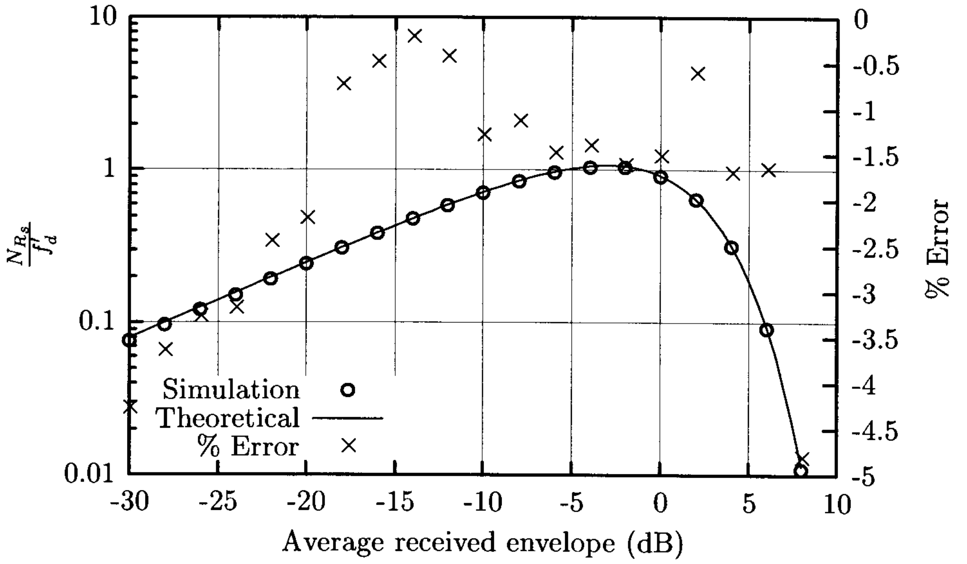

Figure 3.28: Normalized level crossing rate for $f'_d = 10^{-6}$

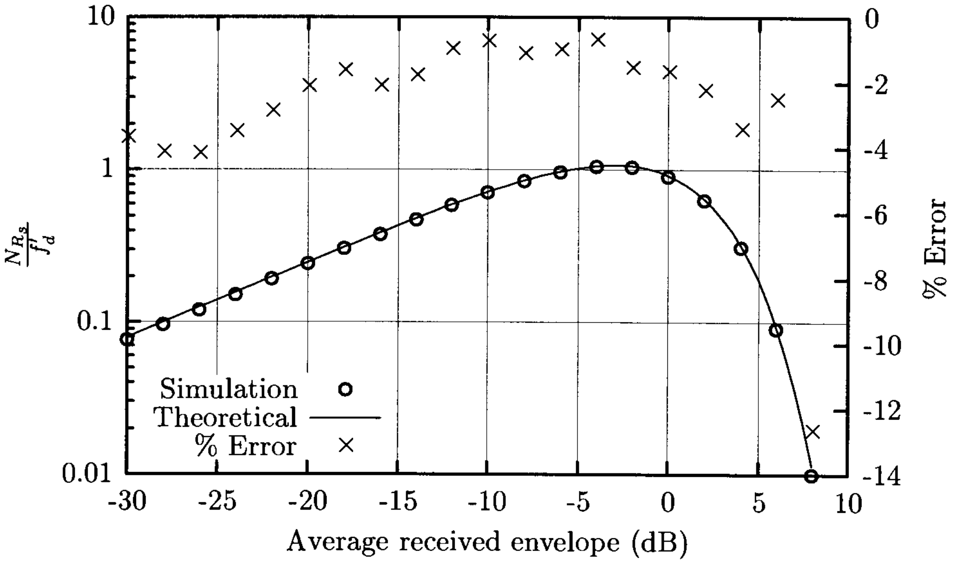

Figure 3.29: Normalized level crossing rate for $f'_d = 10^{-7}$

From these results it can be seen that the greatest approximation error occurs at the lower LCR rates on the far left and far right of each graph. This is a measurement error in that these points have much fewer level crossings in one simulation, which implies a loss in accuracy due to fewer occurrences.

The absolute approximation error is less than 5 percent for the greatest part of the level range, for each Doppler frequency. This shows how well the Doppler filter approximates the theoretical fading statistics over a frequency span of 7 decades. Lower Doppler frequencies could not be tested due to a lack of computing power. The test for $f'_d = 10^{-7}$ ran for about 36 days on an HP B2352B workstation with 256 MByte of RAM.

3.5.3 Fading Examples

In this section some examples of the fading envelope of the received signal are presented. Only the results for $f'_d = 10^{-2}$ are shown, as the lower fading rates require many more samples to achieve the same visual effect.

The Rayleigh faded envelope for this Doppler frequency is shown in Figure 3.30 below. This is equivalent to the following system parameters :

- $f_{sample} = 10\ KHz$
- $v = 120\ km/h$
- $f_c = 900\ MHz$

The Ricean faded envelope for this Doppler frequency is shown in Figure 3.31 below, for a Ricean factor of $K = 3.01dB$, and for the same system parameters. Both of these envelope plots have been ploted relative to the average received envelope, which translates to $0dB$. These plots only allow a qualitive, and not a quantitative representation of the fading process. The PDF and LCR plots are used for the quantitative evaluation of the accuracy of the channel simulator. From the fading plots presented in this section, it can be seen that the excursions away from the mean received level are far more destructive for the Rayleigh case, than for the Ricean case.

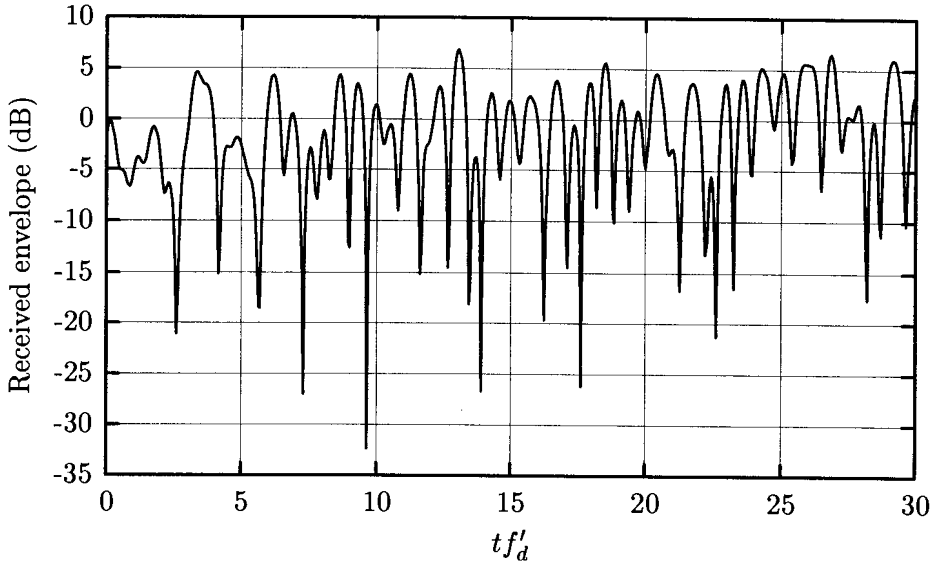

Figure 3.30: Example of the Rayleigh fading envelope from the channel simulator for $f'_d = 10^{-2}$

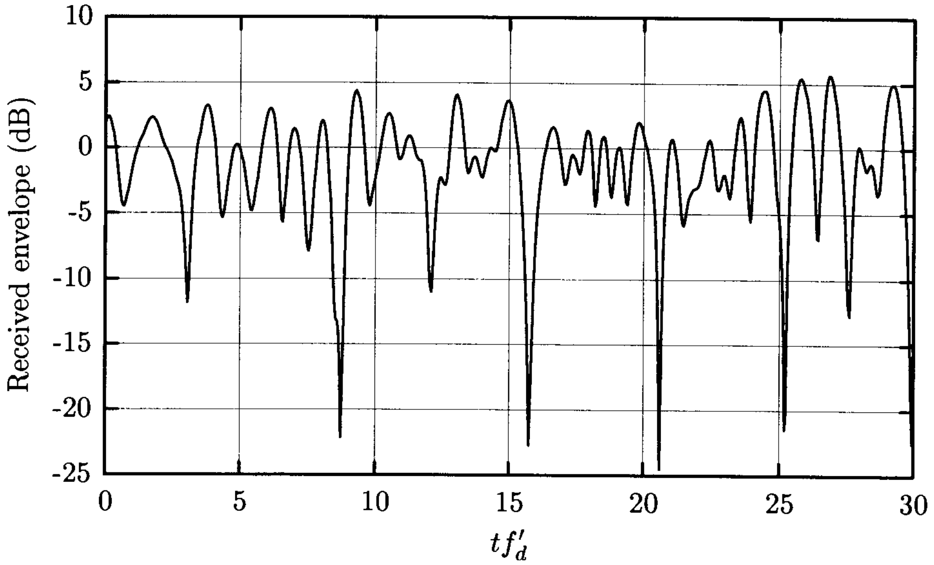

Figure 3.31: Example of the Ricean fading envelope from the channel simulator for $f'_d = 10^{-2}$

3.6 STATIC TWO-RAY MULTIPATH MODEL

This section describes the two-ray static multipath model which will be used for preliminary evaluation of the final modem and equaliser structure. The channel consists of two discrete multipaths of adjustable amplitude, and an adjustable relative time delay. This model was developed by Rummler for the characterisation of multipath in microwave links, but can be generalized, and applied to any communication link.

3.6.1 Rummler's Two-Ray Model

In 1979 Rummler [123] proposed a statistical delay network model for the microwave channel, which has since become one of the most widely cited studies of its kind. Drawing upon 25 000 swept frequency scans, Rummler fitted measured propagation data to a two path transfer function

$$H(f) = a\left[1 - b\exp\left(j2\pi(f - f_0)\tau\right)\right] \tag{3.87}$$

where a is the scale factor, b is the shape factor, f_0 is the frequency where the fade minimum occurs (notch frequency) and τ is the relative time delay between the two paths. This model can be implemented as a two coefficient FIR filter which can be designed to synthesize a frequency notch at any frequency in the band of interest. The parameter a controls the average attenuation over the whole frequency range, and the b parameter controls the depth of the notch.

Use will be made of a simpler form of this model for simulation purposes, which has a discrete-time impulse response given by

$$h[n] = a\left(\delta[n] + b\delta[n - \tau_d]\right) \tag{3.88}$$

where τ_d is an integer denoting the relative delay of the second path in samples. To ensure that the input power is equal to the output power for this channel model, the scale factor must be set as

$$a = \frac{1}{\sqrt{1 + b^2}} \tag{3.89}$$

The frequency response of this filter is given by

$$H(e^{j\omega}) = a^2\left[1 + b^2 + b\cos(\omega\tau_d)\right] \tag{3.90}$$

which has the following parameters in the frequency domain:

- Nominal gain : $20\log_{10}(a)$
- maximum gain (relative to nominal) : $20\log_{10}(1 + b)$
- Null depth (relative to nominal) : $20\log_{10}(1 - b)$
- Gain variation : $20\log_{10}\left(\frac{1+b}{1-b}\right)$
- Null separation : $\frac{1}{\tau_d} = \frac{1}{nT_{sample}}$

3.7 CONCLUSION

This chapter has discussed the wireless mobile channel in terms of a statistical model of the mechanisms which are responsible for fading. Broadly speaking it can be said that most short term fading effects are caused by multipath propagation. Multipath reflections with very little difference in propagation delays sum vectorially at the receive antenna, and are thus responsible for flat fading. Multipaths having more distributed propagation delays cause frequency selective fading.

Both these effects have been incorporated into a software channel simulator to enable the testing of the digital communication platform under realistic, repeatable channel conditions. The performance of this simulator has been demonstrated by verification against theoretical bench marks, in terms of level crossing rates and fading distributions.

CHAPTER 4

GENERIC MULTIDIMENSIONAL DIGITAL COMMUNICATION SIMULATION PLATFORM

This chapter contains a detailed description of the implementation of the multidimensional digital communication simulation platform, which was used as the back bone to test the adaptive equaliser. The design philosophy of the simulation platform was to create a generic multidimensional modem, even though the focus of this study was four dimensional signalling, which is not as computationally intensive as higher dimensional modulation schemes. This is especially true in the case of the adaptive lattice equaliser.

Some of the most prominent design goals were

- Low complexity.
- Bit error probability equal to theoretical value.
- Allowance for various carrier and symbol synchronization techniques.
- Inherent extensibility to higher dimensional modulation schemes.
- Allowance for fractionally spaced equalisation.

The simulation system parameters chosen are summarized in the following table

These parameters are only a guideline, and can be scaled to match the parameters of a practical system.

Parameter	Value
sampling frequency	100 kHz
samples per symbol	10
symbol rate	10 $k\ symbols/s$
bit rate	40 $k\ bits/s$

Table 4.1: System parameters for simulation.

4.1 TRANSMITTER

Several methods of generating a Q^{2}PSK signal exist in the literature. The first approach is that of the direct implementation of the basis functions, which are given by

$$\phi_1(t) = \frac{2}{\sqrt{T_s}}\cos(2\pi f_d t)\cos(2\pi f_c t) \qquad -\frac{T_s}{2} \leq t \leq \frac{T_s}{2} \tag{4.1}$$

$$\phi_2(t) = \frac{2}{\sqrt{T_s}}\sin(2\pi f_d t)\cos(2\pi f_c t) \qquad -\frac{T_s}{2} \leq t \leq \frac{T_s}{2} \tag{4.2}$$

$$\phi_3(t) = \frac{2}{\sqrt{T_s}}\cos(2\pi f_d t)\sin(2\pi f_c t) \qquad -\frac{T_s}{2} \leq t \leq \frac{T_s}{2} \tag{4.3}$$

$$\phi_4(t) = \frac{2}{\sqrt{T_s}}\sin(2\pi f_d t)\sin(2\pi f_c t) \qquad -\frac{T_s}{2} \leq t \leq \frac{T_s}{2} \tag{4.4}$$

$$\phi_i(t) = 0 \qquad |t| > \frac{T_s}{2} \tag{4.5}$$

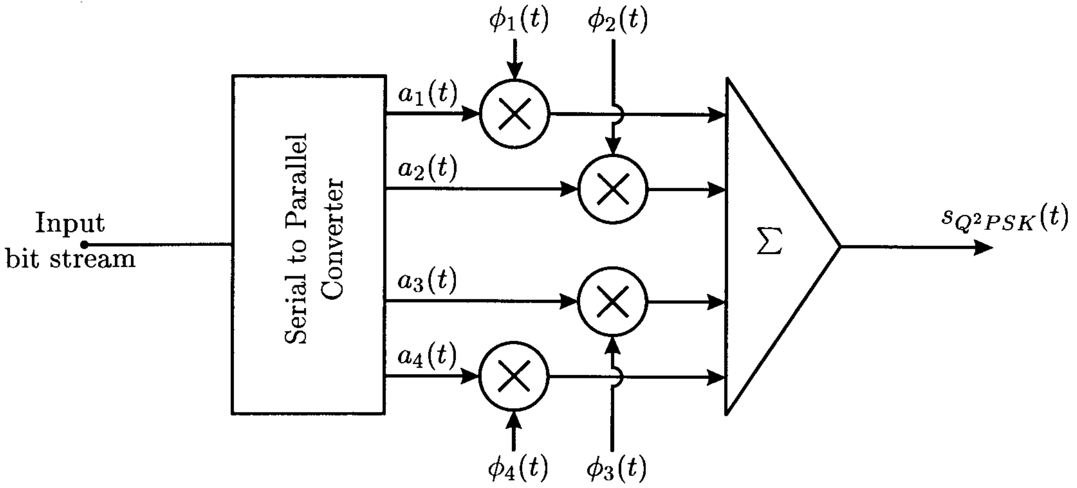

Figure 4.1: Direct implementation of Q^{2}PSK transmitter.

where $T_s = 1/R_s$ is the symbol period, $f_d = 1/2T_s$ is the deviation frequency and f_c is the carrier frequency. These functions form a set of equal-energy orthonormal signals under the restriction that

$$f_c = \frac{n}{2T_s} \qquad n = \text{integer} \geq 2 \tag{4.6}$$

which ensures orthogonality of the carrier components. The direct implementation of these equations is given in [16, 11], and is depicted in Figure 4.1.

This approach is relatively complex if implemented directly, but can be simplified by the use of quadrature modulator structures [27, 15] as shown in Figure 4.2. The inset in the figure shows the internal detail of each I/Q modulator. This method can lead to a very elegant solution if both the symbol shaping waveform and the IF waveform are generated digitally. The modulator based method is limited in that it only allows sinusoidal waveforms, and does not allow for the shaping functions to span more than one symbol period.

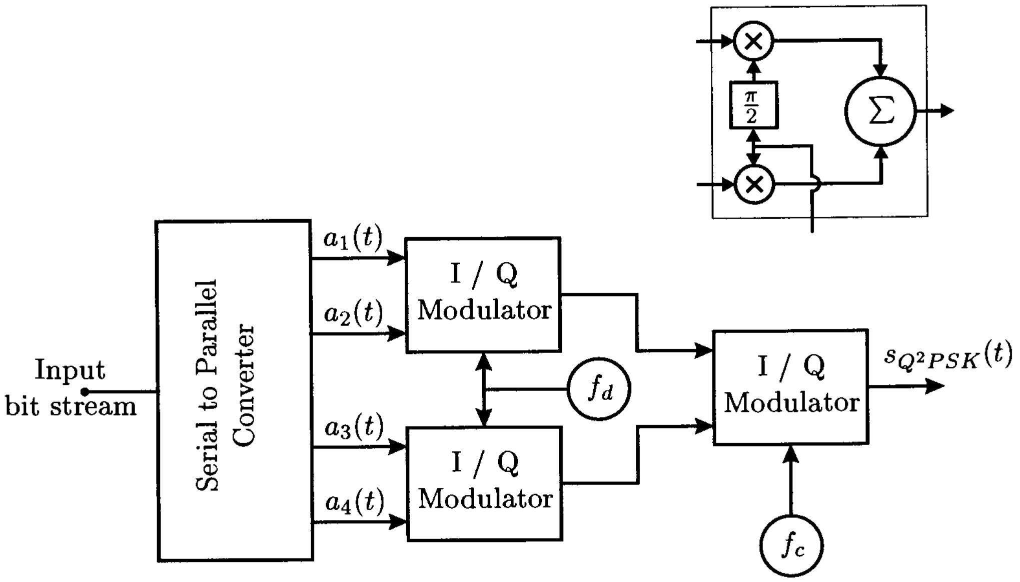

Figure 4.2: Modulator based implementation of Q^{2}PSK transmitter.

The best way to allow for shaping functions which span more than one symbol period is to make use of a finite impulse response (FIR) filter. A generalized mathematical representation of the Q^{2}PSK signalling format is given by

$$\begin{aligned} s_{Q^2PSK}(t) &= \sum_{n=-\infty}^{\infty} \left[a_{1n}p_1(n-nT_s) + a_{2n}p_2(n-nT_s)\right]\cos(2\pi f_c t) \\ &+ \sum_{n=-\infty}^{\infty} \left[a_{3n}p_1(n-nT_s) + a_{4n}p_2(n-nT_s)\right]\sin(2\pi f_c t) \end{aligned} \tag{4.7}$$

where $p_1(t)$ and $p_2(t)$ are real lowpass signals of arbitrary duration. This representation led to the implementation structure which was finally decided on.

4.1.1 Structure

The transmitter structure decided on makes use of FIR filters for the pulse shaping function, followed by a normal quadrature modulator to frequency translate the baseband signals to an intermediate frequency (IF). This structure is depicted in Figure 4.3 below.

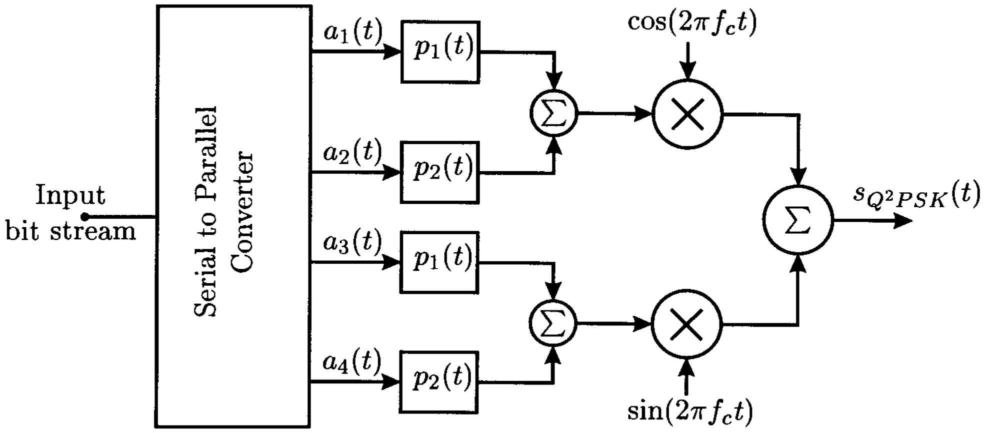

Figure 4.3: Filter based implementation of Q^2PSK transmitter.

As each pulse shaping waveform is used in two of the pulse shaping filters, each waveform only has to be stored once, although both delay lines have to be realized. A further saving in computational complexity can be achieved by noticing that the data in the delay lines is a zero interpolated set with elements ±1. This implies that the filter can be implemented as a lookup table with an offset stored in a register, and the delay line only has to be clocked every symbol period. If the pulse shapes exhibit even or odd symmetry, as is the case with most of the pulse shapes in the literature, the complexity of the filters can be further reduced by a factor of 50 %.

The transmitter can be followed by a passband filter centred at f_c which reduces the sidelobes of the modulated spectra to ensure that the transmission has a minimal effect on adjacent channels. This filter is designed according to the adjacent channel interference (ACI) specification for the specific RF band in which the carrier frequency falls.

4.1.2 Block Transmission Strategy

Use is made of a block transmission strategy to allow the modem to operate in a frequency hopping mode, or in a TDMA environment. The random data transmitted by the transmitter is interspersed with predetermined sequences at regular intervals. Use is made of these sequences to achieve frame synchronisation, as well as cyclic training of the equaliser. The block transmission strategy consists of a header, followed by the information bearing block of symbols, as shown in Figure 4.4 below. The header and associated data block are collectively referred to as a frame.

In practice, in a frequency hopping system, the frame is also preceded, and followed by guard symbols which ensure that no data is lost while the synthesisers are changing the transmission carrier frequency.

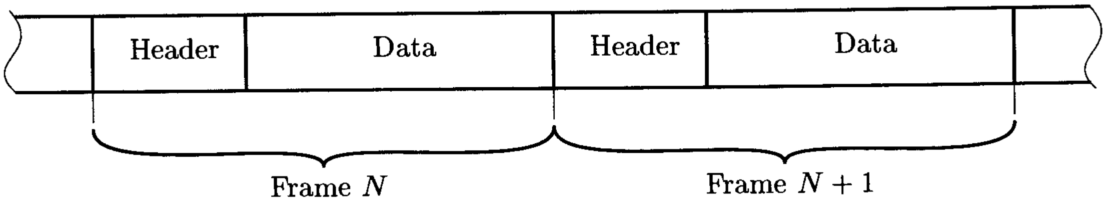

Figure 4.4: Block transmission strategy.

Design of Block Transmission Strategy

In a traditional modem receiver, the total frame length is chosen in such a way as to insure that the channel conditions can be approximated as constant over one block. For a vehicle speed of $120km/h$ the equivalent maximum Doppler frequency is $f_d = 100Hz$. Allowing for a maximum phase rotation, θ_{frame} (in degrees), of 20 degrees over 1 frame allows the frame length to be found as

$$\begin{aligned} T_{frame} &= \frac{\theta_{frame}}{360} f_{d_{max}}^{-1} \\ &= \frac{\theta_{frame}}{360} \frac{c}{v f_c} \qquad (4.8) \\ &= 555.6\mu s \qquad (4.9) \end{aligned}$$

where the carrier frequency is $f_c = 900MHz$. It is interesting to note at this point that the frame length in the Group Special Mobile (GSM) cellular standard is $576.9\mu s$. At a symbol rate of $270.833ksymbol/s$ the corresponding number of symbols in one frame is 150. Due to the fact that the equaliser has to be trained over the header of each frame which is computationally intensive, the evaluation of the equaliser will be restricted to channels having a fading rate of less than $5Hz$. This implies that the frame length can safely be extended to 2000 symbols.

The header sequence used is generated by inserting the data bits given in Table 4.1.2 into the transmitter structure.

It is also necessary to place the last few symbols of the sync sequence before the main sequence, and the first few symbols of the sync sequence after the main sequence, to ensure continuity of the data for the cyclic training of the equaliser in multipath conditions. This can be visualised by placing three sync sequences consecutively, and then cutting out the middle sequence, as well as some buffer symbols of each of the first and third sequences. The number of symbols chosen is dependent on the maximum delay spread of the channel. For evaluation of the equaliser the largest delay spread will be restricted to three symbol periods to limit the complexity of the equaliser necessary to mitigate the multipath effects.

The final header length,L_H, is thus 41 symbols, and 2000 symbols, denoted by L_D, are available for the user data payload per frame. The detailed final frame structure is depicted in Figure 4.5. In this figure h_n is used to denote the nth header symbol, and d_n is used to denote the nth data symbol in the frame structure.

Data stream	Sequence
a_1	+1 +1 +1 +1 +1 -1 -1 +1 +1 -1 +1 -1 -1 +1 -1 -1 -1 -1 +1 -1 +1 -1 +1 +1 +1 -1 +1 +1 -1 -1 -1
a_2	+1 +1 +1 -1 -1 +1 -1 -1 -1 +1 -1 +1 -1 +1 +1 +1 +1 -1 +1 +1 -1 +1 -1 -1 +1 +1 -1 -1 -1 -1 -1
a_3	+1 -1 -1 -1 -1 -1 +1 +1 -1 -1 +1 -1 +1 +1 -1 +1 +1 +1 +1 -1 +1 -1 +1 -1 -1 -1 +1 -1 -1 +1 +1
a_4	+1 -1 -1 -1 +1 +1 -1 +1 +1 +1 -1 +1 -1 +1 -1 -1 -1 -1 +1 -1 -1 +1 -1 +1 +1 -1 -1 +1 +1 +1 +1

Table 4.2: Synchronisation Sequences.

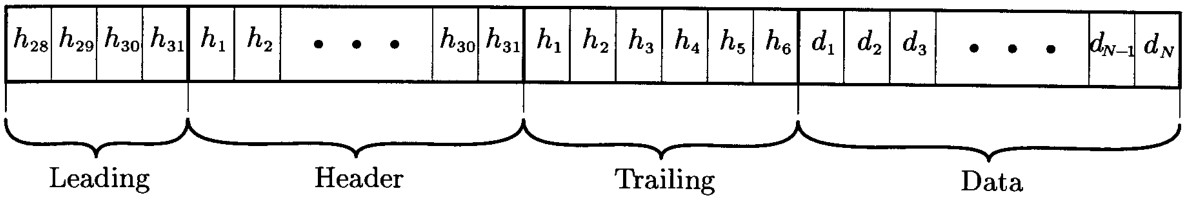

Figure 4.5: Detail frame structure.

4.1.3 Spectra and Waveforms

In this section some of the key waveforms and spectra pertaining to the transmitter will be plotted.

Figure 4.6 shows a 12 symbol example of the transmitter output waveform (upper plot) as well as the envelope (lower plot) of the same transmitter output waveform. It can be seen that for certain combinations of input bits the envelope of the output signal is constant. It should also be noted that where the envelope is zero, the transmitted signal is also zero.

Figure 4.7 shows the computed spectrum of the output signal of the transmitter. The spectrum was computed using an averaging periodogram method, based on a 512 point Fast Fourier Transform (FFT). The data record length was 1.362 million samples.

The simulation results presented in this section compare well with the theoretically expected waveforms and spectra, which were discussed in Chapter 2.

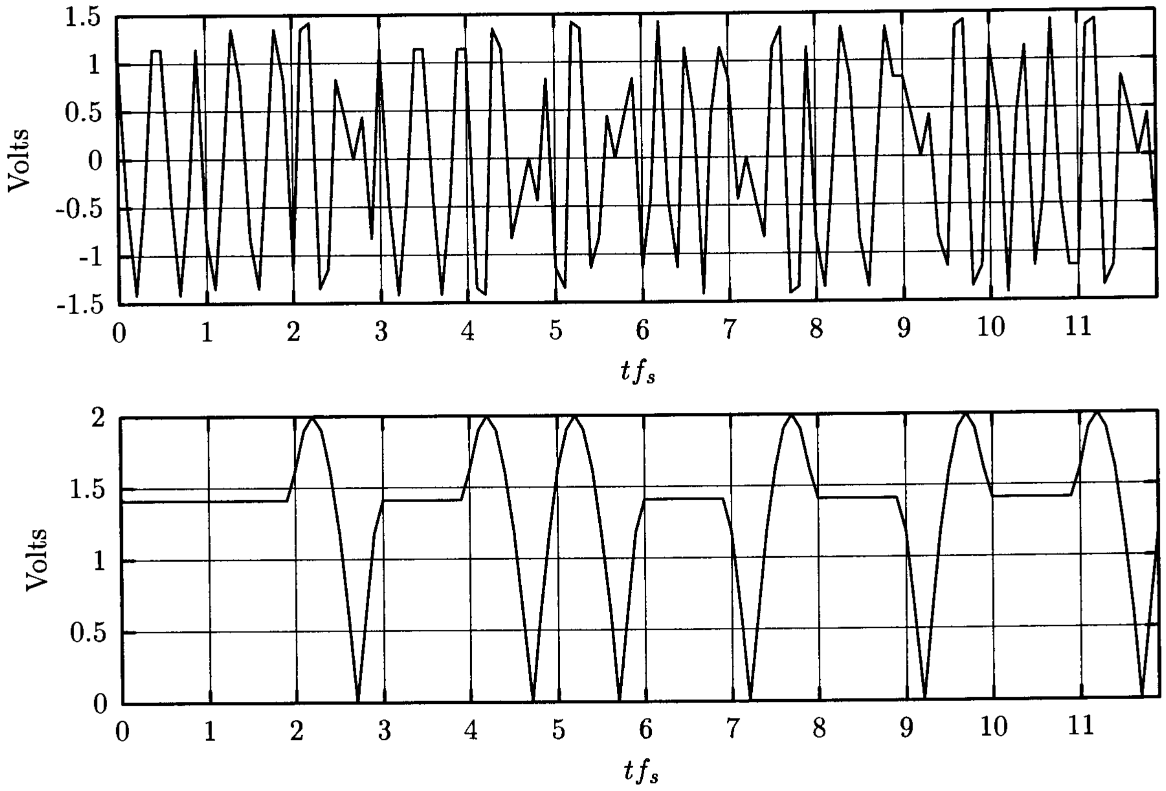

Figure 4.6: Example of transmitter output and envelope for 12 symbols.

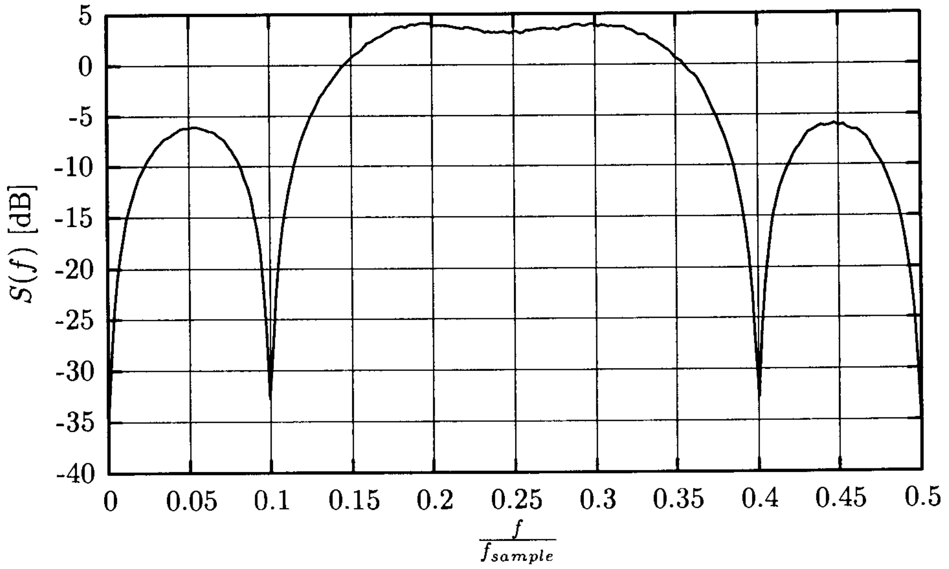

Figure 4.7: Spectrum of transmitter output

4.2 RECEIVER

The receiver section of any digital communication system is by far the most complex subsystem of the modem. This section contains a detailed description of the receiver structure used in the simulation, as well as relevant discussions on certain choices which were made, and options which were considered. Attention is also given to the synchronisation sub-systems necessary in the receiver.

4.2.1 Structure

In this section the various demodulator strucures which were tested will be presented, along the the structure which was finally decided on.

The major design goals for the multidimensional receiver were

- A receiver capable of generating fractionally spaced samples to allow the testing of a fractional equaliser.
- A perfectly open eye pattern at the symbol sampling times to accurately evaluate the effect of multipath, and equalisation on the signal. If this constraint is met then the theoretical BEP will also be achieved.
- Logical unlimited expansion to higher dimensional modulation techniques.
- Allowance for pulse shapes spanning more than one symbol period.
- Low computational complexity.

The following three receiver structures will be considered

- Integrate and dump,
- Single Sideband demodulator based structure and
- Matched filter receiver.

Integrate and Dump Structure

The first publications pertaining to Q^{2}PSK, for example [11], advocated the use of a receiver structure based on direct multiplication by the basis functions, followed by an integrate and dump mechanism as shown in Figure 4.8. Due to the nonlinearity of the integrate and dump mechanism, this structure does not allow the receiver to generate fractionally spaced samples, and was thus discarded.

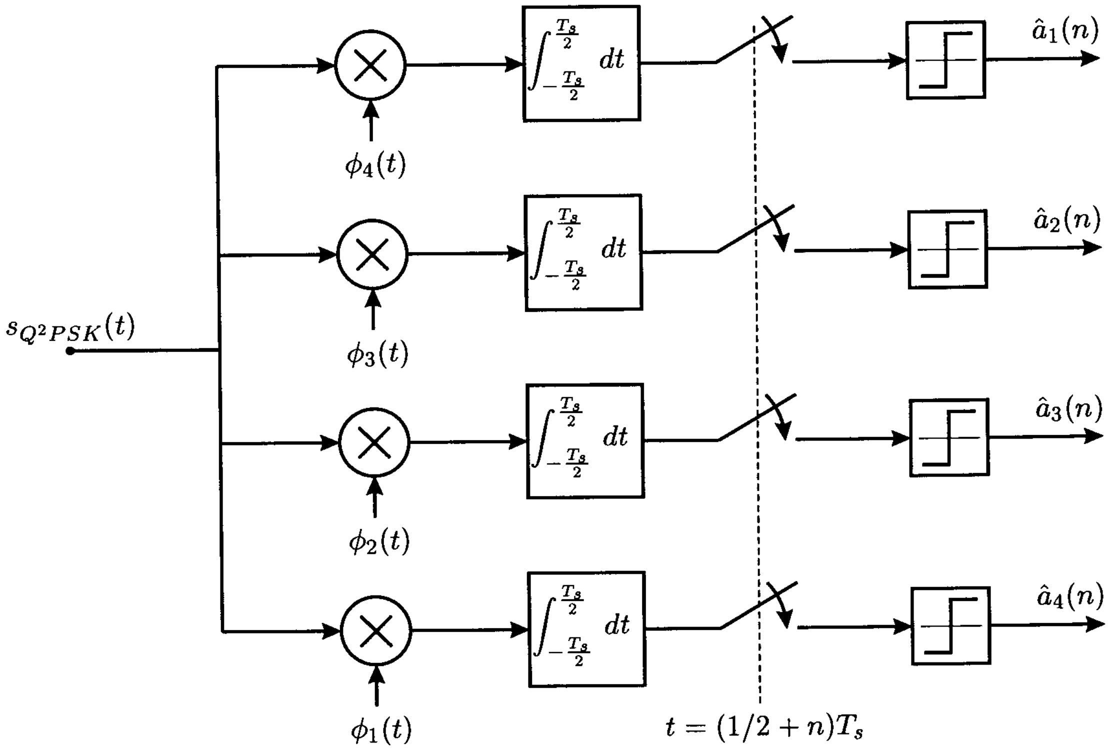

Figure 4.8: Integrate and dump receiver structure for Q^2PSK.

Single Sideband Demodulator Structure

Acha and Carrasco, [16] were the first to present a receiver structure for Q^2PSK which allowed for fractional sampling. The block diagram of this structure is shown in Figure 4.9. From this figure it can be seen that the receiver structure is based on a single sideband demodulator, which is firstly applied to remove the carrier component from the received signal, thus generating an in-phase, and a quadrature phase output. A single sideband demodulator is then applied to each of these outputs to remove the pulse shaping, and produce two signals which can be sampled at the correct timing epoch to extract the data bits.

Each one of the single sideband demodulator blocks contains two bandpass filters, which are orthogonal to each other, and centered around the frequency which has to be eliminated from the signal, i.e. either f_c initially, or f_d once the carrier has been removed. These filters are designed in the baseband, and then translated to the center frequency of interest. The translation as well as the necessary orthogonality is achieved by multiplication of the baseband impulse response by the cosine (for the in-phase branch filter, denoted by Re) and the sine (for the quadrature-phase branch filter, denoted by Im) of the frequency of interest. Due to the low frequency of the data shaping waveforms relative to the symbol length, the output of the first demodulator stage can be considered to be a baseband signal. The filter design becomes a relatively difficult excercise, and is the limiting factor for this receiver structure.

The best eye openings achieved with this structure were only approximately 85 % open, which is not nearly open enough for evaluation of the equaliser. This approach was discarded due to the lack of quality of the output eye patterns.

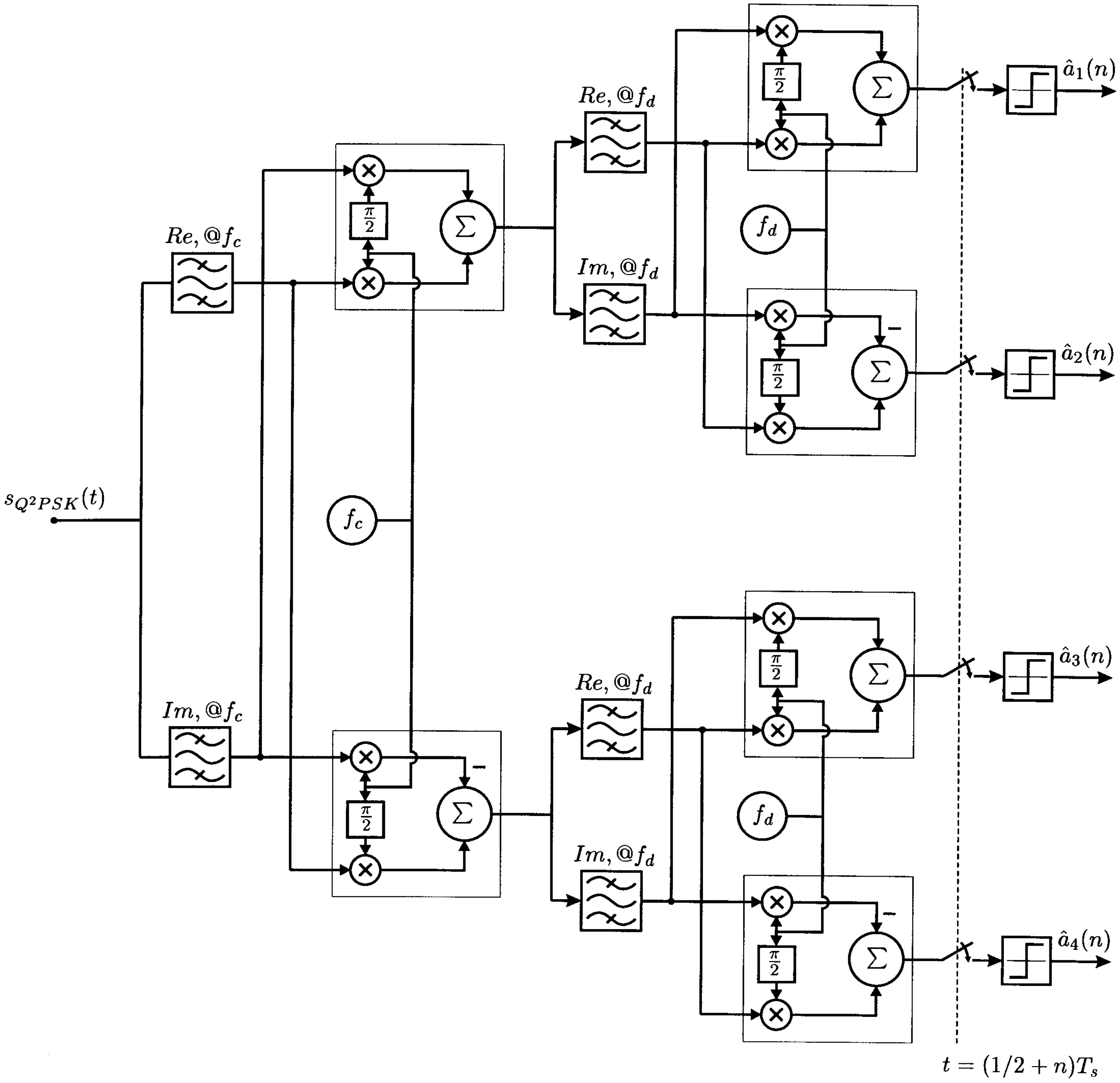

Figure 4.9: Single sideband receiver structure for Q^2PSK.

Matched Filter Receiver

It was finally decided that a matched filter structure would fulfill all the requirements stated in the design goals for the receiver. This form of receiver is mathematically equivalent to the integrate and dump structure discussed above.

The block diagram of this receiver structure is shown in Figure 4.10. If fractionally spaced samples are desired, the samplers can be clocked at twice the symbol rate.

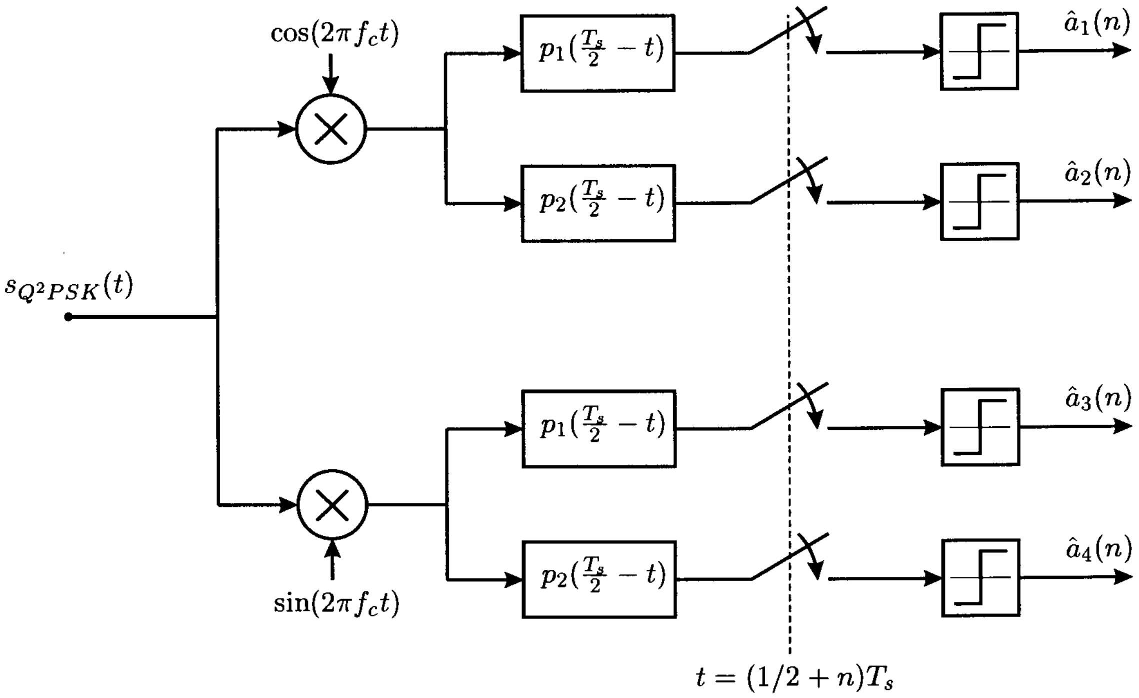

Figure 4.10: Matched filter receiver structure for Q^2PSK.

4.2.2 Synchronization

Synchronisation sub-systems are vitally important to the operation of a receiver demodulator. As the receiver starts receiving a new data frame, on a new frequency, it has no a priori information as to the channel state, the carrier phase and frequency offset or the relative time delay of the receiver waveform. These parameters must thus be estimated first before any attempt can be made to estimate the bit stream which was transmitted. The strategy for the estimation of these unknown parameters is discussed in this section, and leads to the eventual estimation of the data stream.

Frame and Symbol Synchronization

As the receiver starts receiving a new frame the assumption is made that the received signal will be relatively close in frequency to the local oscillator of the receiver. Use is now made of two complex finger correlators which are matched to the synchronisation sequence of Table 4.1.2 to search for the synchronisation sequence in the demodulated received waveforms, as shown in Figure 4.11. In this figure the double lines represent complex signals, and the single lines represent real signals. It can be shown that this structure is equivalent to correlating with the convolution of the pulse shapes and the synchronisation sequence, but it is computationally less intensive. This technique not only finds the frame zero time reference, but also the symbol zero time reference.

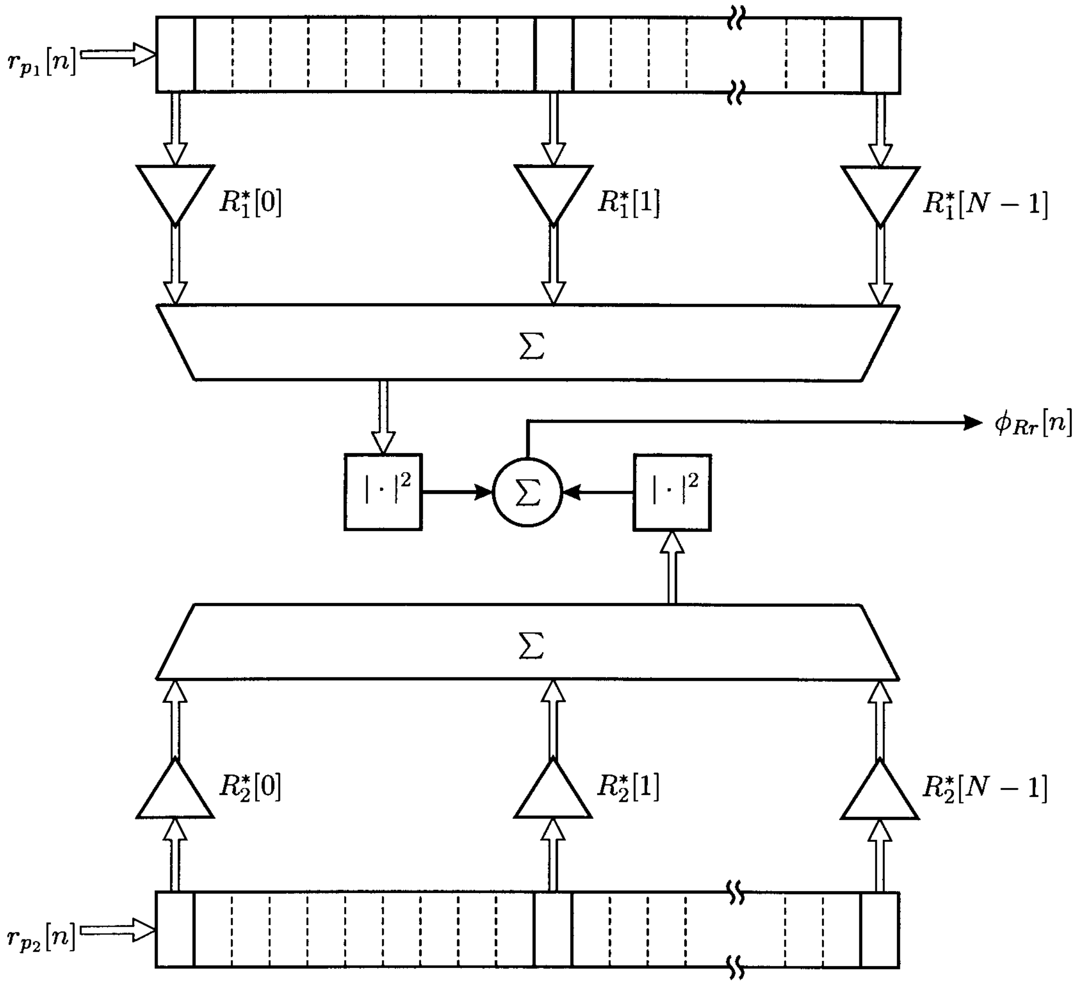

Figure 4.11: Correlator structure

The outputs of the filters matched to the same pulse shape are combined to form a complex number, which is then used as an input sample to a complex finger correlator. Let the output of the filter matched to $p_1(t)$ and $p_2(t)$ be denoted by $r_{p_1}[n]$ and $r_{p_2}[n]$ respectively. Let the reference synchronization sequence related to the $p_1(t)$ and $p_2(t)$ pulse shape be denoted by $R_1[n]$ and $R_2[n]$ respectively. The analytical description of the correlation related to the $p_1(t)$ pulse shape is then

$$\phi_{Rr1}[n] = \sum_{m=0}^{M-1} R_1^*[m] r_{p_1}[n - mI] \tag{4.10}$$

where M is the length of the synchronisation sequence, and I is the interpolation factor for one symbol alternatively referred to as the number of "samples per symbol". The expression for the correlation related to the $p_2(t)$ pulse shape is given by

$$\phi_{Rr2}[n] = \sum_{m=0}^{M-1} R_2^*[m] r_{p_2}[n - mI] \tag{4.11}$$

To obtain a single complex valued output from the two correlators, their outputs are added together.

$$\phi_{Rr}[n] = \phi_{Rr1}[n] + \phi_{Rr2}[n] \tag{4.12}$$

To obtain frame and symbol synchronisation the value of m has to be found for which the absolute value of the output of the combined correlator is a maximum. This is achieved by means of a peak detection algorithm. This algorithm ignores the output of the correlator if it is below a certain threshold value, which is typically set at 5 to 10 percent of the possible peak expected from the correlator. Once the correlator output exceeds this level, the output of the correlator becomes the new reference level, and the time index of this peak is stored. At every time instant which this happens it is assumed that the final peak has been found. The correlator now searches for a number of symbols after this peak to make sure that no higher peaks exist. The distance that the correlator searches after a peak location event is referred to as the "look ahead length". During this procedure the output samples of the matched filter are buffered. If the look ahead length is reached without the detection of a new peak, then the last peak found has a high probability of being the correct frame synchronisation peak. The stored time index of this peak is then used as the zero time reference, as well as the start of the symbol sampling clock at the receiver. The data buffer is also sampled once per symbol, relative to the zero time reference to extract the cyclic training sequence.

The argument (phase angle) of the correlation peak is an estimation of the phase offset of the local oscillators relative to transmitted signal. This phase angle can be used to correct phase offsets of the received data block if the channel varies very little over a single frame.

Carrier Recovery

If the received signal has a phase and frequency offset relative to the local oscillator then a more elaborate carrier recovery algorithm is necessary. Use was made of an adaptive dual-loop digital phase locked loop (DPLL) for decision directed carrier recovery. The DPLL consists of an adaptive estimator which estimates the phase and frequency offsets of the received Q^{2}PSK passband signal. The parameters of the estimator are updated by means of a recursive least squares (RLS) algorithm.

Some synchronization algorithms for Q^{2}PSK have been presented in [11, 27, 16, 28] for stationary channels. Most of these algorithms are based on second or fourth power loops, traditional phase locked loops and maximum-likelihood parameter estimation. In a mobile communication scenario larger carrier frequency offsets (doppler) are possible, especially in airborne applications. Due to the inherent frequency hopping capability of the simulation platform rapid acquisition and synchronization is necessary which implies large loop bandwidths and low jitter reduction. Some method is thus necessary to reduce the loop bandwidth after acquisition to improve the jitter reduction. The Kalman DPLL is an optimal method for achieving continuous loop bandwidth reduction. It also has the ability to change the loop bandwidth dynamically as a function of the received signal to noise ratio, which can vary as a function of time due the fading mobile channel.

The dual-loop DPLL consists of two, first order, interlinked recursive digital filters and a digital differentiator. One of the recursive filters is used to track the frequency offset, and the other the phase offset of the received signal. The input to the loop is the instantaneous angular offset of the received data signal, which serves as one of the inputs to the phase tracking filter. The derivative of the input angle, which is obtained by means of the first order digital differentiator, is the instantaneous frequency offset. The block diagram of the dual-loop DPLL is shown in Figure 4.12 below.

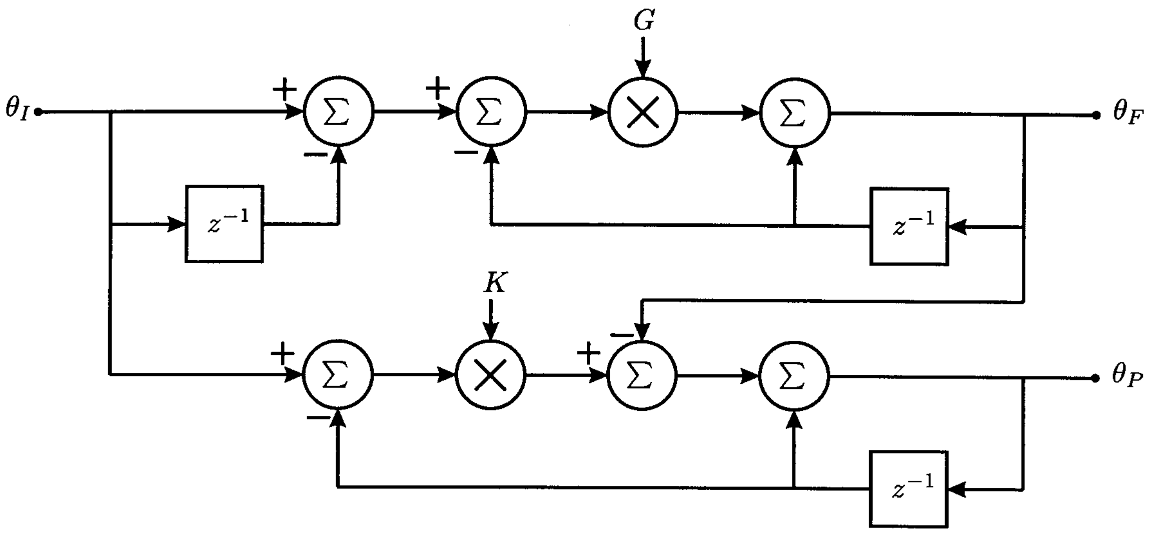

Figure 4.12: Dual-loop digital phase-locked loop block diagram.

As shown in this figure, there are two control parameters, G for the frequency-locked loop, and K for the phase-locked loop. These parameters are time variant, and are thus expressed as $K(n+1)$ and $G(n+1)$. If the sampling period is denoted by T_s, then $\theta_P(n+1)$ and $\theta_F(n+1)$ are the outputs of the phase-locked and frequency-locked loops respectively at time $t = (n+1)T_s$. The time domain update equations for the dual-loop DPLL are described by the following equations:

$$\theta_P(n+1) = \theta_P(n) + \theta_F(n+1) + K(n+1)\,[\theta_I(n) - \theta_P(n)] \tag{4.13}$$
$$\theta_F(n+1) = \theta_F(n) + G(n+1)\cdot[\theta_I(n) - \theta_I(n-1) - \theta_F(n)] \tag{4.14}$$

Equation (4.14) is a correction to [124]. If the input frequency is slowly changing with time over the interval of interest, $\theta_P(i)$ and $\theta_F(i)$ at time $(n+1)T_s$ can be expressed as:

$$
\begin{aligned}
\theta_{Pn+1}(i) &= \theta_{Pn+1}(n+1) - (n+1-i)\theta_{Fn+1} && (4.15)\\
\theta_{Fn+1}(i) &\approx \theta_{Fn+1} = \theta_{Fn+1}(n+1) && (4.16)
\end{aligned}
$$

These equations are a linearisation of the system model to aid the derivation of the update algorithm. The update algorithm is based on the principle of minimization of the exponentially weighted least squared error at time $(n+1)T_s$, which is given by:

$$
\varepsilon_{n+1} = \sum_{i=1}^{n+1} \lambda^{n+1-i} \left|\theta_I(i) - \theta_{Pn+1}(i)\right|^2 \qquad (4.17)
$$

where λ is the exponential weighting factor. The case $\lambda = 1$ corresponds to infinite memory. Positive values of λ less than 1 control the time span or memory length of the algorithm.

If the partial derivatives of Equation (4.17) to $K(n+1)$ and $G(n+1)$ are taken, and set to zero, a set of linear equations for computing the gains can be derived:

$$
\mathbf{v}_{n+1} = \mathbf{\Phi}_{n+1}\mathbf{w}_{n+1} \qquad (4.18)
$$

where the component matrices of this equation are given by:

$$
\mathbf{v}_{n+1} = \begin{bmatrix} \sum_{i=1}^{n+1} \lambda^{n+1-i} X_n(i) \\ \sum_{i=1}^{n+1} \lambda^{n+1-i} X_n(i) Z_n(i) \end{bmatrix} \qquad (4.19)
$$

$$
\mathbf{\Phi}_{n+1} = \begin{bmatrix} \sum_{i=1}^{n+1} \lambda^{n+1-i} Y_n(i) & \sum_{i=1}^{n+1} \lambda^{n+1-i} - Z_n(i) \\ \sum_{i=1}^{n+1} \lambda^{n+1-i} Y_n(i) Z_n(i) & \sum_{i=1}^{n+1} \lambda^{n+1-i} - Z_n^2(i) \end{bmatrix} \qquad (4.20)
$$

and

$$
\mathbf{w}_{n+1} = \begin{bmatrix} K_{n+1} \\ G_{n+1} \end{bmatrix} \qquad (4.21)
$$

Again where

$$
\begin{aligned}
X_n(i) &= \theta_I(i) - \theta_{Pn}(n) - (n-i)\theta_{Fn} && (4.22)\\
Y_n(i) &= \theta_I(n) - \theta_{Pn}(n) && (4.23)
\end{aligned}
$$

and

$$Z_n(i) = (n-i)\left[\theta_I(n) - \theta_I(n-1) - \theta_{Fn}\right] \tag{4.24}$$

The updated parameters K_{n+1} and G_{n+1} can be obtained from:

$$\mathbf{w}_{n+1} = \mathbf{\Phi}_{n+1}^{-1}\mathbf{v}_{n+1} \tag{4.25}$$

However the computational cost of the summations in the above equations can be eliminated by means of the following recursive equations [125]:

$$\mathbf{v}_{n+1} = \mathbf{G}_n v_{n+1} \tag{4.26}$$

$$\mathbf{\Phi}_{n+1} = \mathbf{A}_n \phi_{n+1} \tag{4.27}$$

$$v_{n+1} = \mathbf{\Lambda}_n v_n + \mathbf{P}_n \tag{4.28}$$

$$\phi_{n+1} = \lambda \phi_n + \mathbf{R}_n \tag{4.29}$$

where $\mathbf{v}_n$, v_n, $\mathbf{G}_n$ and $\mathbf{P}_n$ are of dimension 2×1, and $\mathbf{\Phi}_n$, ϕ_n, $\mathbf{A}_n$, $\mathbf{\Lambda}_n$ and $\mathbf{R}_n$ are of dimension 2×2. The symbol "•" is usd to denote element-by-element multiplaication.

The component elements of these matrices are given by:

$$\mathbf{G}_n = \begin{bmatrix} 1 \\ F_{nd} \end{bmatrix} \tag{4.30}$$

$$\mathbf{P}_n = \begin{bmatrix} -\lambda P_n S_{0,n} + \lambda Q_n S_{1,n} + R_n \\ -\lambda P_n S_{1,n} + \lambda Q_n S_{2,n} - R_n \end{bmatrix} \tag{4.31}$$

$$\mathbf{A}_n = \begin{bmatrix} P_{nd} & F_{nd} \\ P_{nd}F_{nd} & F_{nd}^2 \end{bmatrix} \tag{4.32}$$

$$\mathbf{\Lambda}_n = \begin{bmatrix} \lambda & 0 \\ \lambda & \lambda \end{bmatrix} \tag{4.33}$$

$$\mathbf{R}_n = \begin{bmatrix} 1 & -\lambda S_{0,n} + 1 \\ \lambda S_{0,n} - 1 & \lambda S_{0,n} - 2\lambda S_{1,n} - 1 \end{bmatrix} \tag{4.34}$$

where

$$F_{nd} = \Theta_I(n) - \Theta_I(n-1) - \Theta_{Fn} \tag{4.35}$$

$$P_{nd} = \Theta_I(n) - \Theta_{Pn}(n) \tag{4.36}$$

$$P_n = \Theta_{Pn}(n) - \Theta_{Pn-1}(n-1) - \Theta_{Fn-1} \tag{4.37}$$

$$Q_n = \Theta_{Fn} - \Theta_{Fn-1} \tag{4.38}$$

$$R_n = \Theta_I(n+1) - \Theta_{Pn}(n) - \Theta_{Fn} \tag{4.39}$$

and

$$S_{2,n+1} = \sum_{i=1}^{n+1} \lambda^{n+1-i}(n+1-i)^2 \tag{4.40}$$

$$S_{1,n+1} = \sum_{i=1}^{n+1} \lambda^{n+1-i}(n+1-i) \tag{4.41}$$

$$S_{0,n+1} = \sum_{i=1}^{n+1} \lambda^{n+1-i} \tag{4.42}$$

The adaptive loop filter was first tested in isolation to determine the stability, convergence time and tracking ability of the RLS algorithm. It was found that the algorithm can become momentarily unstable for one sample, so the gains have to be limited as follows

$$K, G \leq 1 \tag{4.43}$$

This has the effect of limiting the filter frequency response to lowpass filter forms only and allows convergence of the algorithm within 15 samples.

The DPLL algorithm was trained over the header and then switched into a data directed mode for reception of the data block. In this mode the received complex constellation are first derotated by the phase angle estimated by the DPLL. Decisions are then made on the derotated symbols. The phase difference between the decision and the derotated symbol is used as the input to the DPLL. The DPLL gains are then updated, and then the DPLL predicts the phase offset of the next symbol.

As the modem operates in a frequency hopping mode, channel information gathered during the reception of one frame is invalid at the start of the next frame. The DPLL thus has to be retrained on every received header block.

Data Recovery

Once the frame and symbol synchronisation has been achieved, and the carrier tracking loop has stabilised, the data bits can be retrieved by making bipolar decisions on the derotated outputs of the matched filter. These decision can then be converted back to a serial data stream by means of a parallel to serial converter. This data stream can be compared with the original data stream which was transmitted to estimate the probability of bit errors.

4.2.3 Waveforms and Signal Constellations

In this section some of the key waveforms in the receiver will be shown. The eye patterns at the outputs of two of the matched filters will be presented, followed by full signal constellations to demonstrate the effect of multipath.

The eye patterns at the outputs of the filters matched to p_1 (upper plot) and p_2 (lower plot) are shown in Figure 4.13 below. These plots span two symbol periods, so two eyes can be seen in each plot, and no noise has been added to the received signal. It can be seen that the eye patterns are open at the optimal sampling times, and that the amplitudes are exactly unity at the sampling point. The eye opening associated with p_2 is only 50% the width of the eye opening associated with p_1 in the horizontal direction. This is due to the sharp discontinuities present at symbol transitions in the p_2 pulse train.

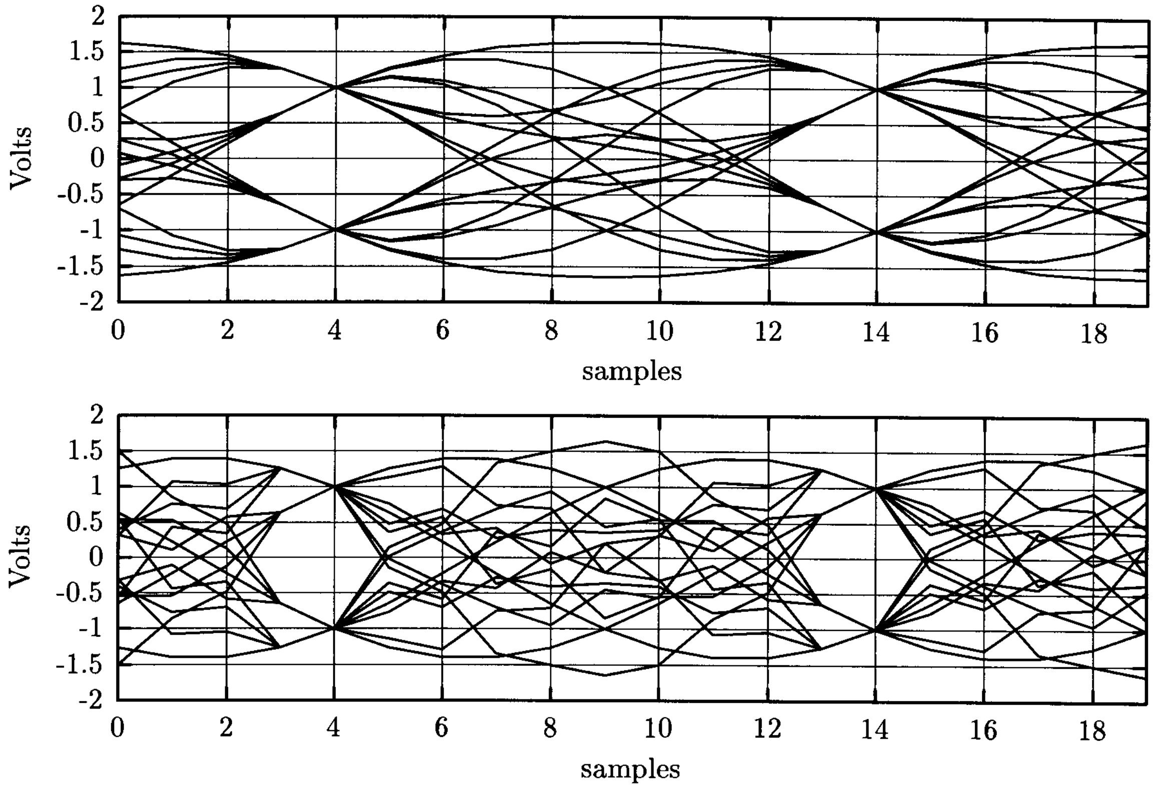

Figure 4.13: Examples of the eye patterns at the outputs of the matched filters.

If the outputs of the matched filters are sampled at the optimum sampling points, these samples can be plotted as two signal constellation plots, as described in Paragraph 2.4.3. Figure 4.14 shows the signal constellations for AWGN at $E_b/N_0 = 30$ dB. Figure 4.15 shows the signal constellations in the presence of multipath. For this case there is a single multipath at 10dB below the line of sight path, and at a time delay of $0.7T_s$. The bit energy to noise is $E_b/N_0 = 40$ dB. For both of these plots the x-axis represents the in-phase (I) component, and the y-axis represents the quadrature-phase (Q) component of the carrier.

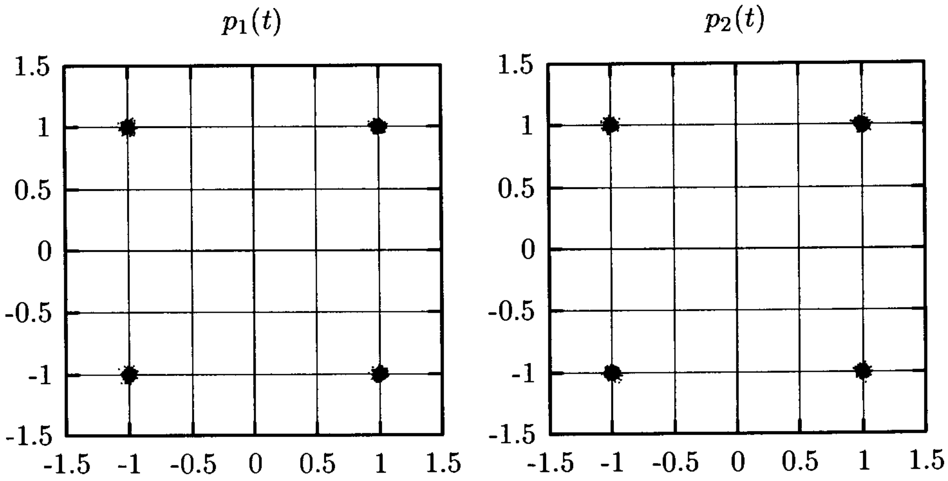

Figure 4.14: Examples of the signal constellation at the outputs of the matched filters in the presence of AWGN.

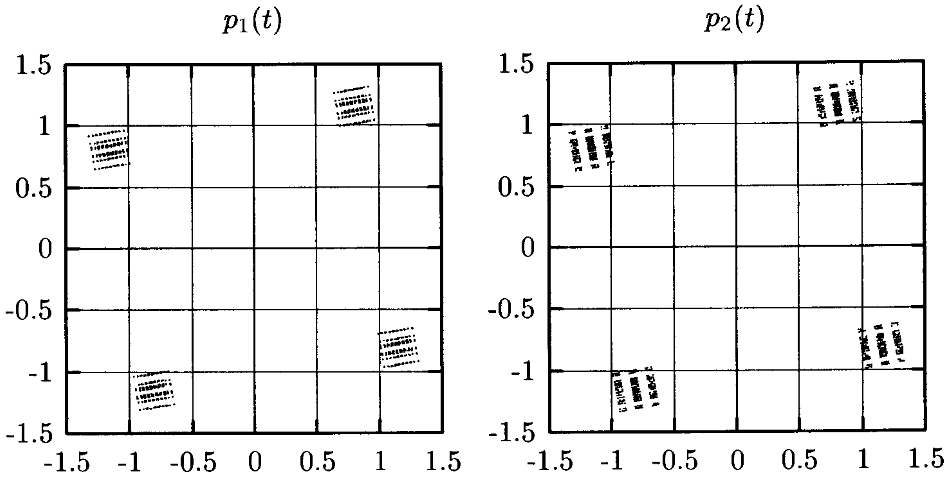

Figure 4.15: Examples of the signal constellation at the outputs of the matched filters in the presence of multipath propagation.

4.3 CONCLUSION

A detailed description of the generic multi-dimensional modem simulation platform has been given in this section. The data framing strategy for frequency hopping protocols has been designed. The structure of the transmitter sub-system, which is responsible for the generation and transmission of the multi-dimensional waveforms has been discussed. The integration of the frame protocol with this transmitter structure was also covered. The structure and operation of the receiver sub-system was also discussed. This system is responsible for the estimation of unknown channel related parameters which are necessary for accurate retrieval of the data bits from the received signal. The synchronisation sub-systems of the receiver, which are responsible for frame synchronisation, carrier synchronisation, symbol synchronisation and data stream recovery were also described in detail. Examples of some of the signals and spectra at certain points throughout the transmitter and receiver structure were also given.

This chapter has described the implementation of a simulation platform for transmission of the data stream as a multidimensional signal and the extraction of this data stream from the received signal but has not addressed the problem of compensating for the effect of multipath on the received signal. This topic will be addressed by first introducing the criteria for distortionless transmission in the following chapter.

CHAPTER 5

CHANNEL EQUALISATION for Q^{2}PSK

This chapter presents the proposed algorithm for channel equalisation of a multidimensional Q^{2}PSK modulation format signal. Firstly the concept of and criteria for the distortionless transmission of a signal over a channel are discussed. From this the methods of compensation for and correction of distortion by means of linear filtering are presented. For time varying channels, such as the mobile communication channel, this linear filtering process has to be a function of time, which implies some form of an adaptive filter. In leading up to a specialised form of an adaptive filter, known as the lattice filter, the concepts and algorithms associated with linear prediction are also presented. Finally the adaptive multichannel lattice filter algorithm is presented as a solution to the multidimensional signal equalisation problem at hand. The integration of this filter with the existing receiver structure is also discussed.

5.1 THEORETICAL DISCUSSION

This section presents a detailed discussion of distortionless transmission, and methods to mitigate the effects of distortion.

5.1.1 Distortionless Transmission

Distortionless transmission can be defined as the situation in which the received signal in a communication system is an exact replica of the transmitted signal, except for a possible change in amplitude, and a constant time delay. As these signals are the signals at the input and output ports of the channel, distortion is an effect of the channel. An input signal to the channel, $x(t)$, is related to the output signal, $y(t)$ of the channel by

$$y(t) = Kx(t - t_0) \tag{5.1}$$

if the channel is distortionless. The constant K and t_0 account for the change in amplitude and transmission delay respectively. By taking the Fourier transform of this relationship the transfer function of the distortionless channel can be found as

$$\begin{aligned} H(f) &= \frac{Y(f)}{X(f)} \\ &= K\exp(-j2\pi f t_0) \end{aligned} \tag{5.2}$$

The generalized expression for distortionless transmission is given by

$$H(f) = K\exp\left[j(-2\pi f t_0 \pm n\pi)\right] \qquad n \in I \tag{5.3}$$

From this transfer function the conditions for distortionless transmission are:

1. The amplitude respone, $|H(f)|$ is constant for all frequencies of interest:

$$|H(f)| = K \tag{5.4}$$

2. The phase $\phi(f)$ is a linear function of frequency, passing through zero or a multiple of π at $f = 0$:

$$\phi(f) = -2\pi f t_0 \pm n\pi \tag{5.5}$$

If, as is usually the case with a communication system, the spectrum of the transmitted signal is limited to a specific band of frequencies, then these criteria need only be satisfied within that band of of frequencies. These conditions are illustrated in Figures 5.1 and 5.2 for low-pass, and band-pass channels respectively.

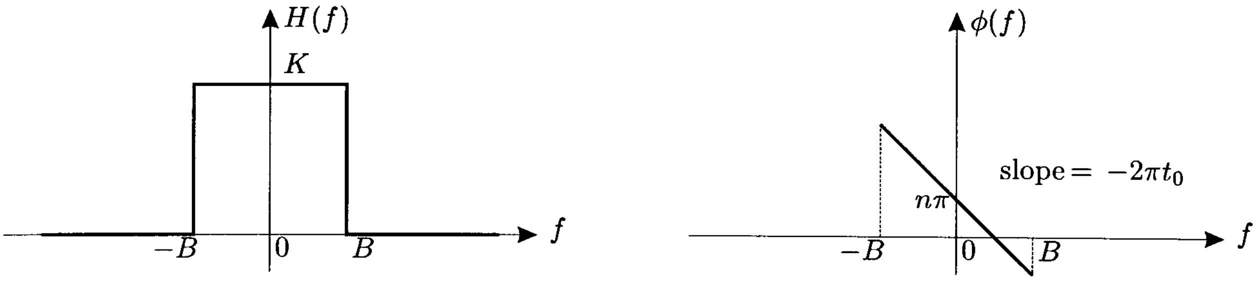

Figure 5.1: Frequency response of an ideal distortionless low-pass channel

In a practical system, the criteria for distortionless transmission can only be approximated, so there is always a certain amount of distortion in the output signal of the channel. Such a channel is termed 'dispersive', and two forms of distortion can be distinguished for such a channel :

1. Amplitude distortion occurs when the amplitude response, $|H(f)|$ is not constant within the frequency band of interest.

2. Phase or delay distortion occurs when the phase response of the channel is not linear with frequency over the frequency band of interest.

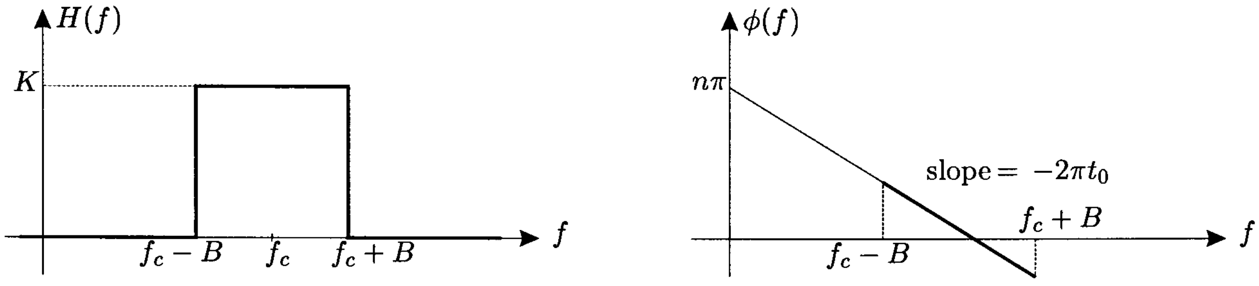

Figure 5.2: Frequency response of an ideal distortionless band-pass channel

Obviously these two forms of distortion rarely occur separately, but rather in conjunction with one another, as the phase of a linear system tends to vary with the rate of variation of the magnitude function.

Most digital modulation techniques make use of essentially time limited pulses to transmit data. Passing this type of signal through a dispersive channel can have a very detrimental effect on the correct reception of the pulses. This is due to the fact that the dispersive channel smears the pulse over time to such an extent that it overlaps the symbol transmitted in the next signalling interval. This effect is known as intersymbol interference and is often abbreviated to ISI.

Distortionless transmission and the criteria therefor have been defined. The effect of non-adherence to these criteria, which leads to dispersive channels, has also been discussed, as well as its effect on a digital communication system.

5.1.2 Inverse Modelling and Deconvolution

It is possible to compensate for the effects of distortion discussed in the previous section by adding a linear time invariant system in cascade to the channel. If the channel transfer function is known, it is immaterial wether the filter is added before or after the channel. In a digital communication system the channel is unknown, and a function of time, so only the case where the compensating filter is added after the channel will be considered as depicted in Figure 5.3, and as discrete time processing will be used at the receiver, the compensating filter will be considered in this domain. The object of the addition of this filter is to compensate for the distortion caused by the channel.

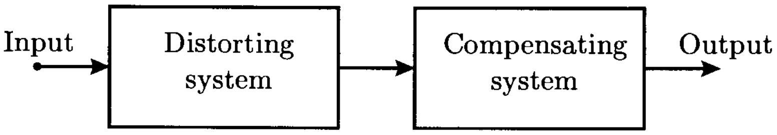

Figure 5.3: Post distortion compensation.

For a given linear time-invariant system with a transfer function denoted by $H(z)$, the corresponding inverse system is defined to be the system having the transfer function $H_i(z)$, such that if it is cascaded with $H(z)$ the overall transfer function of the system, $G(z)$ is given by

$$G(z) = H(z)H_i(z) = 1 \tag{5.6}$$

or alternatively in the time domain

$$g[n] = h[n] \otimes h_i[n] = \delta[n] \tag{5.7}$$

which is the definition of deconvolution. Deconvolution is thus a linear operation that removes the effect of some previous convolution, in this case caused by the impulse response of the channel, $h[n]$. The overall system thus conforms to the criteria for distortionless transmission. From (5.6) the transfer function of the inverse system is given by

$$H_i(z) = \frac{1}{H(z)} \tag{5.8}$$

This implies that the log magnitude, phase, and group delay of the inverse system are the negatives of their corresponding functions for the original system.

Existence of the Inverse System

The inverse function of a system does not always exist. This can be seen be regarding a general transfer function given by

$$H(z) = \frac{b_0}{a_0}\frac{\prod\limits_{k=1}^{M}(1 - c_k z^{-1})}{\prod\limits_{k=1}^{N}(1 - d_k z^{-1})} \tag{5.9}$$

which has zeros located at $z = c_k$, and poles $z = d_k$. The transfer function of the inverse function is now given by

$$\begin{aligned} H_i(z) &= \frac{1}{H(z)} \\ &= \frac{a_0}{b_0}\frac{\prod\limits_{k=1}^{N}(1 - d_k z^{-1})}{\prod\limits_{k=1}^{M}(1 - c_k z^{-1})} \end{aligned} \tag{5.10}$$

The poles of $H_i(z)$ are at the positions of the zeros of $H(z)$, and the zeros of $H_i(z)$ are at the positions of the poles of $H(z)$. The criteria for the existence of the inverse system in the discrete-time domain are thus as follows:

1. The inverse system has to be a causal system, which implies that the region of convergence must be given by

$$|z| > \underset{k}{max}|c_k| \tag{5.11}$$

2. The inverse system also has to be stable, which implies that the region of convergence of $H_i(z)$ must include the unit circle. Therefore it must be true that

$$\underset{k}{max}|c_k| < 1 \tag{5.12}$$

This means that all the zeros of $H(z)$ must be inside the unit circle.

Thus a linear time-invariant system is stable and causal and also has a stable and causal inverse if and only if both the poles and zeros of $H(z)$ are inside the unit circle:

$$|c_k| < 1, \qquad |d_k| < 1 \tag{5.13}$$

Such systems are referred to as minimum phase systems.

The process described above whereby correction of channel induced distortion is achieved by means of inverse modelling is known as equalisation, and the compensation filter is referred to as an equaliser. If the channel is known, and adheres to the criteria for the existence of an inverse model, then the inverse model can be implemented, and cascaded with the channel, as a fixed equaliser. If the channel is time-variant, the equaliser has to be made adaptive to cope with the changing channel model.

5.1.3 Time Variant Solutions

From the above discussion pertaining to the correction for distortion by means of inverse modelling, and the discussion on the characteristics of mobile channel, it can be seen that an adaptive equaliser is a prerequisite for successful digital communication over a fading mobile channel. This section gives a brief description of the main classes of adaptive equalisers and their modes of operation.

Linear Equalisation

The linear filter most often used for equalisation is the finite impulse response (FIR) filter, which is also referred to as a transversal filter. This group can be divided into two main categories. Structures in which the sampling rate of the filter is equal to the symbol rate, R_s, are the first class. Equalisers in the second class are referred to as fractional equalisers, and have a sampling rate which is higher than the symbol rate. The most common sampling rate is twice the symbol rate, which implies two samples per symbol, but non-integer rates between R_s and $2R_s$ are also possible. The coefficients of the filter are varied by means of some algorithm which makes use of a performance measure to optimize the response of the filter. In general this type of equaliser has to have an infinite number of coefficients to completely eliminate the distortion caused by the channel. These equalisers can thus only mitigate relatively mild distortion to acceptable levels. If the equaliser is limited in length, then a global optimal setting for the coefficients exists in most cases so these algorithms tend to converge.

Decision Feedback Equalisation

This type of equaliser has a feedback structure which is very similar to an infinite impulse response filter, except that decision made on the output of the equaliser are fed back, and not the actual filter output. If the error rate is low enough to ensure that most of the decisions which are fed back are correct, the effect of these symbols may be subtracted from the next received symbol. The feedback portion of the equaliser can only be symbol spaced, but the feedforward section of the equaliser may be fractionally spaced. The feedforward section is identical to the linear equaliser described in the previous section. It should be observed that if the estimated symbol were fed back then this would be a linear equaliser, but due to the fact that the decision is fed back this is a nonlinear equaliser. This type of equaliser can synthesize an infinite length impulse response due to its IIR structure. Thus such equalisers yield a significant performance improvement over linear equalisers in terms of bit error probability, as well as the severity of the distortion which can be mitigated. Unfortunately the decision-feedback equaliser (DFE) is not without it own set of associated problems. The two most prevalent of these are:

- Loss of performance due to incorrect errors being fed back.
- The IIR structure introduces the possibility that the equaliser can become unstable.

5.2 THE MULTICHANNEL ADAPTIVE LATTICE FILTER

In the previous section different equaliser topologies were discussed, but no attention was given to the algorithms necessary for the adjustment of the equaliser coefficients. In this section the recursive least squares (RLS) algorithm will be developed from a linear prediction theory approach. The lattice filter topology will be introduced as a special case of the RLS algorithm, and will be extended to the multidimensional case for non-equal length data records for application as a decision-feedback equaliser for multidimensional digital modulation techniques.

5.2.1 Linear Optimal Filtering

The problem addressed in this section, is the optimal design of a FIR filter impulse response with the goal of obtaining the optimal estimate of some desired signal from a set of input samples to the FIR filter. This problem will be discussed for the case of complex data and coefficients, as the application is to a digital communication system which uses complex baseband signals for information transfer.

Figure 5.4 shows a block diagram of the problem at hand. The input to the FIR filter is the sequence $x[n]$, and its coefficients are denoted by $w[0], w[1], \ldots, w[N-1]$. At some discrete time n the filter produces an output $y[n]$, which provides an estimate of the desired response, $d[n]$. The filter input, and the desired response are single realisations of respective stochastic processes. The error signal is defined as the difference between the desired signal and the output of the filter

$$e[n] = d[n] - y[n], \tag{5.14}$$

and has statistical characteristics of its own. The requirement is to make this estimation error as small as possible in some statistical sense. This statistical optimization is normally carried out by means of the minimization of a cost function, or performance index. Use is made of the mean-square value of the estimation error, which is termed the mean-square error (MSE). Other choices exist, but the MSE leads to tractable mathematics, and a second order dependence for the cost function on the unknown filter coefficients. The MSE thus has a distinct global minimum, which uniquely defines the optimal statistical design of the filter. The MSE cost function is defined as

$$\begin{aligned} J &= E\{e[n]e^*[n]\} && (5.15)\\ &= E\{|e[n]|^2\}. && (5.16) \end{aligned}$$

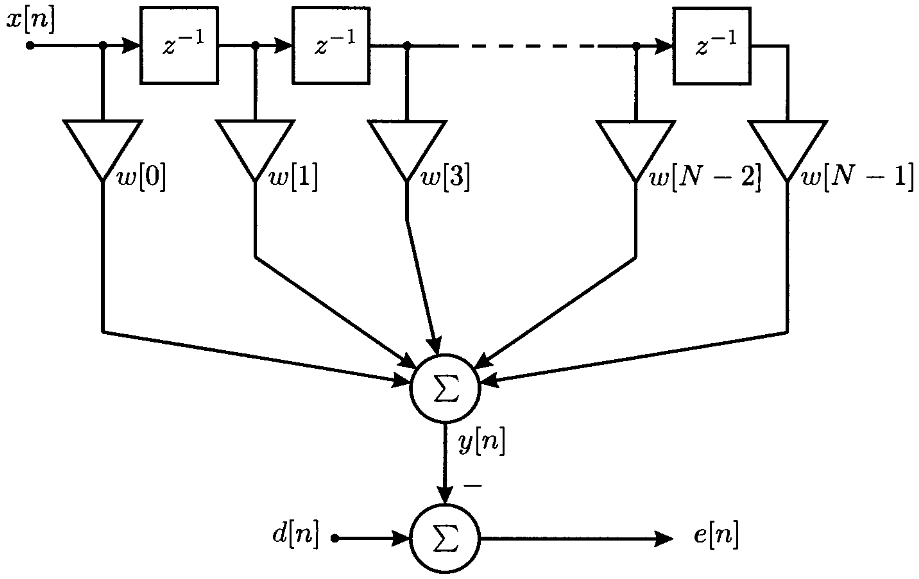

Figure 5.4: Linear predictor.

To expand this equation it is necessary to express the output of the filter in terms of its input and coefficients as

$$\begin{aligned} y[n] &= \sum_{k=0}^{N-1} w^*[k]x[n-k] && (5.17) \\ &= \mathbf{w}^H\mathbf{x} && (5.18) \end{aligned}$$

where the superscript H denotes Hermitian transpose. The cost function J can now be expressed as a function of the filter coefficients as follows

$$\begin{aligned} J(\mathbf{w}) &= E\{e[n]e^*[n]\} \\ &= E\{(d[n]-\mathbf{w}^H\mathbf{x})\,(d[n]-\mathbf{w}^H\mathbf{x})^*\} \end{aligned}$$

which, after some brief algebraic manipulation, gives

$$J(\mathbf{w}) = \sigma_d^2 - \mathbf{w}^H\mathbf{p} - \mathbf{p}^H\mathbf{w} + \mathbf{w}^H\mathbf{R}\mathbf{w} \tag{5.19}$$

where $\mathbf{R}$ is the $N \times N$ input correlation matrix defined by

$$\mathbf{R} = E\{\mathbf{x}[n]\,\mathbf{x}^H[n]\}, \tag{5.20}$$

$\mathbf{p}$ is the $N \times 1$ cross-correlation vector defined by

$$\mathbf{p} = E\{\mathbf{x}[n]\,d^*[n]\} \tag{5.21}$$

and $\sigma_d^2 = E\{d[n]\;d^*[n]\}$ is the power in the desired signal.

The optimal solution for the filter coefficients can now be found by taking the derivative of (5.19) to the tap weight vector, $\mathbf{w}$, and setting it to zero. Each complex coefficient, $w[k]$ can be written in terms of its real and imaginary parts as

$$w[k] = a[k] + jb[k] \qquad k = 0, 1, 2, \ldots, N \tag{5.22}$$

The gradient operator ∇ of this coefficient is defined in terms of first order partial derivatives with respect to a and b as

$$\nabla_k = \frac{\partial}{\partial a[k]} + j\frac{\partial}{\partial b[k]} \qquad k = 0, 1, 2, \ldots, N \tag{5.23}$$

Given this definition, it is now possible to apply the operator ∇ to the cost function J, and obtain a multidimensional gradient vector ∇J, of which the kth element is given by

$$\nabla_k J = \frac{\partial J}{\partial a[k]} + j\frac{\partial J}{\partial b[k]} \qquad k = 0, 1, 2, \ldots, N \tag{5.24}$$

It should be noted that for this definition of the complex gradient to be valid, it is essential that J be real. For the cost function to obtain its minimum value, all the elements of the gradient vector ∇J must be simultaneously equal to zero, i.e.

$$\nabla_k J = 0 \qquad k = 0, 1, 2, \ldots, N \tag{5.25}$$

which is a set of N simultaneous equations. The application of the ∇ operator to the cost function gives the following result

$$\nabla_k J = 2\mathbf{R}\mathbf{w}_{opt} - 2\mathbf{p} = 0 \tag{5.26}$$

where $\mathbf{w}_{opt}$ is the optimum set of coefficients for the filter, and can be solved for to give

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} \tag{5.27}$$

which is known as the Wiener-Hopf equation. If the filter coefficients are set to $\mathbf{w}_{opt}$, then the filter is said to be optimum in the mean-squared-error sense.

5.2.2 Forward and Backward Linear Prediction

Forward and backward linear predictors are special cases of the optimal filtering problem discussed in the previous section. The forward prediction filter attempts to predict the current observation, $x[n]$, from its subspace of past observations spanned by $x[n-1], x[n-2], \ldots, x[n-N]$. This can be achieved in an optimal manner by application of the theory of optimal filtering in the mean-squared-error sense. For this case the desired signal $d[n]$ in the previous section is replaced by $x[n]$, and the error $e[n]$ now becomes the forward prediction error, $f_N[n]$, where the subscript denotes the order of the forward predictor. Due to the fact that the desired signal is now a sample from the same sequence as the input to the filter, the form of the cross-correlation vector has to change. The cross-correlation vector (previously denoted by $\mathbf{p}$) is now denoted by $\mathbf{r}$, and is defined as

$$\mathbf{r} = E\{\mathbf{x}[n-1]\, x^*[n]\} \tag{5.28}$$

The definition of the input correlation matrix also has to be changed slightly, but its numerical value stays the same, so it is still denoted by **R**, and is defined as follows

$$\mathbf{R} = E\{\mathbf{x}[n-1]\,\mathbf{x}^H[n-1]\}, \tag{5.29}$$

The coefficients for the forward linear predictor are now denoted by $\mathbf{w}_f$, and can be solved optimally in a mean-squared-error sense as

$$\mathbf{w}_f = \mathbf{R}^{-1}\mathbf{r} \tag{5.30}$$

For the backward linear prediction case the set of N input samples $x[n], x[n-1], \ldots, x[n-N+1]$ are used to predict one sample backwards in time, $x[n-N]$. Once again some subtle changes in the definitions of the variables are necessary. The cross-crrelation vector is still denoted by **r**, and is defined as

$$\mathbf{r}^{B*} = E\{\mathbf{x}(n)\,x^*[n-N]\} \tag{5.31}$$

where the superscript B denotes backward arrangement of the vector. Once again the definition of **R** changes as it is now time indexed to n, but its numerical value stays the same. The backward prediction error is denoted by $b_N[n]$. The coefficients for the backward linear predictor are now denoted by $\mathbf{w}_b$, and can be solved optimally in a mean-squared-error sense as

$$\mathbf{w}_b = \mathbf{R}^{-1}\mathbf{r}^{B*} \tag{5.32}$$

As can be seen from the definitions of this section, the problems of forward, and backward linear prediction are in many ways the duals of one another.

Forward and Backward Error Filters

Many algorithms have been derived to accomplish linear prediction for a broad range of applications. The derivation of these linear prediction algorithms generally require the solution of both the forward and backward linear prediction subproblems. It is thus often necessary to solve for the prediction errors directly, instead of first generating the prediction, and then calculating the prediction error. This direct calculation of the prediction errors can be achieved by means of a FIR filter, by expanding the prediction errors in terms of the prediction filters.

The forward prediction error can be written as

$$\begin{aligned} f_N[n] &= x[n] - \sum_{k=1}^{N} w_f^*[k]x[n-k] && (5.33)\\ &= x[n] - \mathbf{w}_f^*\mathbf{x}[n-1] && (5.34)\end{aligned}$$

from which it can be seen that it is possible to form a new $N+1$ length FIR filter to directly generate the prediction error. The new filter coefficients are defined as

$$\mathbf{a_N} = \begin{bmatrix} 1 \\ -\mathbf{w}_f \end{bmatrix} \tag{5.35}$$

and the forward prediction filter equation is now given as

$$\begin{aligned} f_N[n] &= \sum_{k=0}^{N} a_N^*[k]x[n-k] && (5.36)\\ &= \mathbf{a}_N^*\mathbf{x}[n] && (5.37)\end{aligned}$$

Before proceeding to the direct generation of the backward prediction error it is first necessary to derive a relationship between the backward and forward predictors coefficients. Starting off with the linear equations for forward and backward prediction,

$$\begin{aligned} \mathbf{R}\,\mathbf{w}_f &= \mathbf{r} & (5.38)\\ \mathbf{R}\,\mathbf{w}_b &= \mathbf{r}^{B*}, & (5.39) \end{aligned}$$

taking into account that $\mathbf{R}^H = \mathbf{R}$ and making use of the backward arrangement operator as well as complex conjugate, (5.39) can be reformed as

$$\mathbf{R}\,\mathbf{w}_b^{B*} = \mathbf{r} \tag{5.40}$$

If this result is compared with (5.38), it can be seen that

$$\mathbf{w}_b^{B*} = \mathbf{w}_f \tag{5.41}$$

which represents the fundamental relationship between the coefficient vectors of the froward and backward predictors.

Now, in a similar fashion to the forward prediction error filter, and by making use of the relationship between the forward and backward predictor coefficients, the backward prediction error filter can be expressed as

$$\begin{aligned} b_N[n] &= \sum_{k=0}^{N} a_N[N-k]x[n-k] & (5.42)\\ &= \mathbf{a}_N^B \mathbf{x}[n] & (5.43) \end{aligned}$$

Comparison of these representations of the backward and forward prediction error filters reveals that these two forms of filters are uniquely related to each other for stationary inputs. In particular, we may modify a forward prediction error filter into the corresponding backward prediction error filter by reversing the order of the coefficients, and taking their complex conjugates.

Orthogonality of the Backward Prediction Errors

The backward prediction errors, $b_0[n], b_1[n], \ldots, b_N[n]$, which are the elements of the vector $\mathbf{b}[n]$ have a very important property: They are all orthogonal to each other.

$$E\{b_m[n]\, b_i^*[n]\} = \begin{cases} P_m, & i = m \\ 0\ \ , & i \neq m \end{cases} \tag{5.44}$$

where P_m is the backward prediction error power defined by

$$P_m = E\{|b_m[n]|^2\} \tag{5.45}$$

This orthogonality can be proved by making use of the principle of orthogonality, which states that backward prediction errors are orthogonal to the input samples $x[n]$. It can also be shown that $\mathbf{b}[n]$ and $\mathbf{x}[n]$ are equivalent in the sense that they contain the same amount of information. The Gram-Schmidt algorithm can be used to transform the correlated samples of $\mathbf{x}[n]$ into the vector of uncorrelated backward prediction errors. This algorithm requires a backward prediction error filter for every backward prediction error to be generated. A far more efficient method for the generation of backward prediction errors is by means of a recursive structure known as a lattice predictor.

5.2.3 The Lattice Algorithm

A lattice predictor is an order recursive device which is a far more computationally efficient method for the generation of the backward prediction errors. This device combines several forward and backward prediction error filtering operations into a single structure. It consists of a cascade connection of elementary units (stages) which have a structure similar to that of a lattice, hence the name. To generate M forward and backward prediction errors, M lattice stages are necessary.

The derivation of the lattice predictor is based on the Levinson-Durbin algorithm, and is dependent on the specific form of this algorithm utilized in the derivation. The derivation proceeds by expressing the forward and backward prediction error filters of order m in terms of the same filters of order $m-1$. These update equation are thus referred to as order update equations, and are given by

$$\mathbf{a}_m = \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} + k_m \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} \tag{5.46}$$

for the forward prediction error filter and by

$$\mathbf{a}_m^{B*} = \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} + k_m^* \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} \tag{5.47}$$

for the backward prediction error filter, where the reflection coefficient is given by

$$k_m = \frac{E\left\{b_{m-1}[n-1]\, f_{m-1}^*[n]\right\}}{E\left\{|f_{m-1}[n]|^2\right\}} \tag{5.48}$$

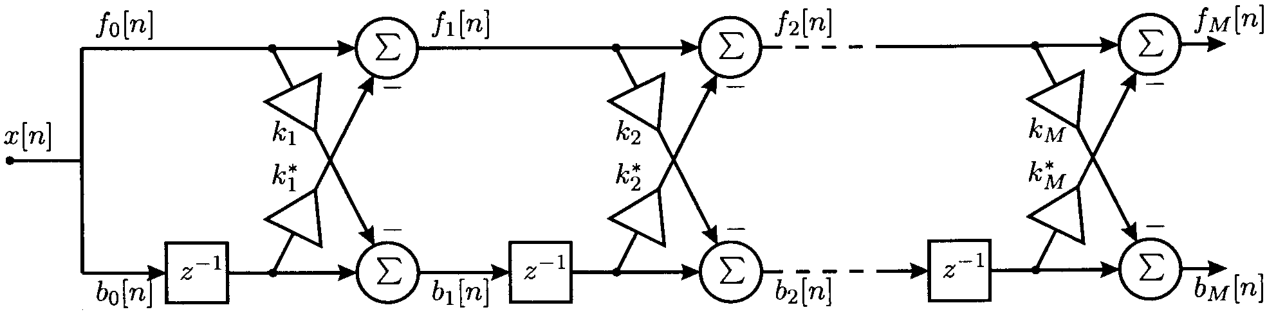

Figure 5.5: Lattice prediction error filter

The order updates for the forward and backward prediction errors are now given by

$$\begin{aligned} f_m[n] &= f_{m-1}[n] + k_m^* b_{m-1}[n-1] && (5.49) \\ b_m[n] &= b_{m-1}[n-1] + k_m f_{m-1}[n] && (5.50) \end{aligned}$$

respectively. The case of $m=0$ is used to obtain the initial condition of

$$f_0[n] = b_0[n] = x[n] \tag{5.51}$$

where $x[n]$ is the input sample at time n. Thus by starting with $m = 0$, and progressively increasing the order of the filter by one, the lattice equivalent model for a prediction error filter of final order N, as well as all lower orders, can be obtained. Only knowledge of the set of reflection coefficients, k_m, is necessary for this operation. The lattice structure is shown in Figure 5.5.

The lattice filter can thus seen to be a highly efficient modular structure for the simultaneous generation of the sequence of forward and backward prediction errors.

5.2.4 Joint-Process Estimation

The linear optimal filtering problem can be simplified significantly by making use of the lattice structure. Once again the problem is to estimate a desired process signal, $d[n]$, from samples of a process $x[n]$ under the assumption that these processes are jointly stationary. Initially use was made of the samples of the process $x[n]$ directly; now the approach will be to use the set of backward prediction errors obtained from feeding samples of the input process, $x[n]$, into a lattice predictor. The fact that the backward prediction errors are orthogonal to each other simplifies the solution of the problem significantly. The joint process estimator is shown in Figure 5.6 below. The lattice predictor, consisting of N stages transforms the sequence of correlated input samples, $\{x[n], x[n-1], \ldots, x[n-N]\}$ into a corresponding sequence of uncorrelated backward prediction errors, $\{b_0[n], b_1[n], \ldots, b_N[n]\}$. The sequence of backward prediction errors is then

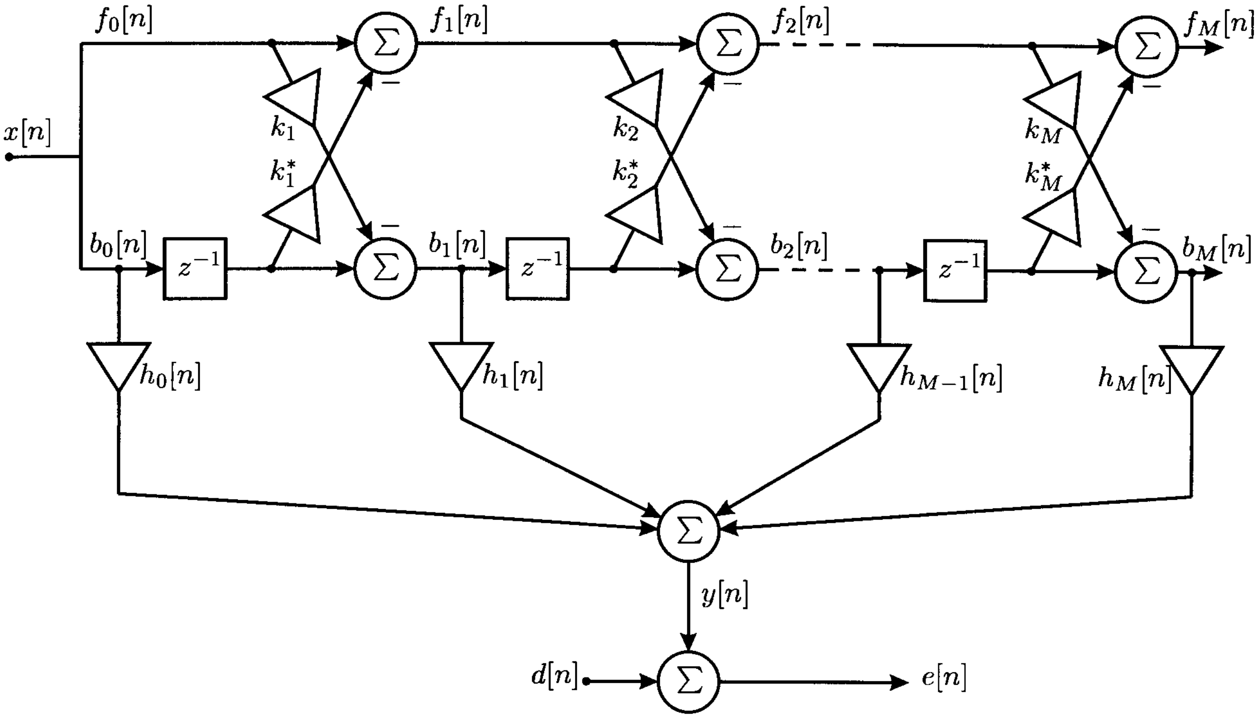

Figure 5.6: Joint-process estimator

used as the input into the so-called multiple regression filter. The estimated desired sequence is then given by

$$\begin{aligned} y[n] &= \sum_{i=0}^{N} h_i^* b_i[n] && (5.52) \\ &= \mathbf{h}^H \mathbf{b}[n] && (5.53) \end{aligned}$$

where $\mathbf{h}$ is the $(M+1) \times 1$ regression coefficient vector defined by

$$\mathbf{h} = [h_0, h_1, \ldots, h_N] \tag{5.54}$$

Now by denoting the correlation matrix of $\mathbf{b}[n]$ as $\mathbf{D}$, and the cross correlation vector between $\mathbf{b}$ and $d[n]$ as $\mathbf{z}$, where

$$\mathbf{z} = E\{\mathbf{b}[n]\ d^*[n]\}. \tag{5.55}$$

the Wiener-Hopf equations yield the optimum regression coefficient vector, which is given by

$$\mathbf{h}_{opt} = \mathbf{D}^{-1}\mathbf{z} \tag{5.56}$$

where the matrix $\mathbf{D}^{-1}$ is a diagonal matrix. This means that unlike the original case, the computation of $\mathbf{h}_{opt}$ is relatively simple to accomplish, as no direct matrix inversion operation is necessary.

5.3 The Recursive Least Squares Algorithm

In the previous section the joint process estimator was developed for the case where all the statistics of the signals involved was known before hand. That is the numerical values of the correlation matrices and vectors was known. If this information is not available, then it also has to be estimated. This is achieved by means of the recursive least squares (RLS) algorithm. This algorithm can be applied to the direct estimation case, or the joint process estimation case. The two realisations of the algorithm are equivalent, but the lattice tends to be more stable and and lends itself better to expansion

In the recursive implementation of the method of least squares, the filter and related statistical variables to be estimated, start off in some known initial condition. The information contained in new samples entering the system is then used to update the old estimates. An important feature of the RLS algorithm is that it utilizes information contained in the input samples, extending back to when the algorithm was initiated.

The derivation of the RLS algorithm proceeds along the same lines as that of for optimal linear filtering, with some minor changes to account for the lack of a-priori information about the statistics of the signals.

The cost function to be minimized is $\xi[n]$, where n is the variable length of the observable data. A weighting function is also introduced into the cost function to give the algorithm the capability to weight current data more than older data, which is especially important in non-stationary applications such as channel equalisation. The general form of the cost function is

$$\xi[n] = \sum_{i=1}^{n} \beta(n,i)|e[i]|^2 \tag{5.57}$$

where $e[i]$ is the error signal defined by

$$\begin{aligned} e[i] &= d[i] - y[i] \qquad (5.58) \\ &= d[i] - \mathbf{w}^H[n]\mathbf{x}[i]. \qquad (5.59) \end{aligned}$$

The desired response is denoted by $d[i]$, and the estimation output by $y[i]$. The filter input vector at time i is defined as

$$\mathbf{x}[i] = [x[i], x[i-1], \ldots, x[i-N+1]]^T \qquad (5.60)$$

and the coefficient vector at time n is defined as

$$\mathbf{w}[n] = [w_0[n], w_1[n], \ldots, w_{N-1}[n]]^T \qquad (5.61)$$

Use is normally made of an exponential weighting factor or ”forgetting” factor defined by

$$\beta(n,i) = \lambda^{n-i} \qquad (5.62)$$

where λ is a positive constant between in the range $[0,1]$. It is usually chosen to be less than, but close to unity. If it is chosen to be unity then the algorithm has infinite memory. The length of the algorithms memory can be calculated roughly by $(1-\lambda)^{-1}$. This leads to the method of exponentially weighted least squares, where the cost function to be minimized is

$$\xi[n] = \sum_{i=1}^{n} \lambda^{n-i}|e[i]|^2 \qquad (5.63)$$

The optimum value of the tap weight vector, for which the cost function attains its minimum value is defined by the normal equations in matrix form

$$\mathbf{\Phi}[n]\mathbf{w}_{opt}[n] = \mathbf{z}[\mathbf{n}] \qquad (5.64)$$

where the $N \times N$ estimated correlation matrix is defined by

$$\mathbf{\Phi}[n] = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}[i]\mathbf{x}^H[i] \qquad (5.65)$$

and the estimated correlation vector between the input and desired sequence by

$$\mathbf{z}[n] = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}[i]d^*(i) \qquad (5.66)$$

Recursions for the updating of these correlation matrices are given as

$$\begin{aligned} \mathbf{\Phi}[n] &= \lambda\mathbf{\Phi}[n-1] + \mathbf{x}[n]\mathbf{x}^H[n] \qquad (5.67) \\ \mathbf{z}[n] &= \lambda\mathbf{z}[n-1] + \mathbf{x}[n]d^*[n] \qquad (5.68) \end{aligned}$$

To compute the optimal set of coefficients, it is necessary to find the matrix inverse of $\mathbf{\Phi}[n]$ at every time step. In practice it can become very computationally intensive to perform this calculation, especially for larger values of N. The ability to calculate the coefficient matrix recursively will also be a desirable feature of such an algorithm. Both these requirements can be met by application of a basic matrix algebra result known as the “matrix inversion lemma”. This leads to many simplified forms of the RLS algorithm without loss in performance, of which the recursive least squares lattice algorithm is a typical example.

5.3.1 The Recursive Least Squares Lattice Algorithm

The full derivation of the recursive least squares lattice algorithm is given in [33], and only the final result will be presented here. The algorithm is of the exponentially weighted type which was described in the previous section. The algorithm is described below, where the order of the lattice stage is denoted by m.

Algorithm 5.1 The RLS Lattice Algorithm

Available at time n :

$k_m[n-1]$: Reflection coefficient
$k^x_m[n-1]$: Filter coefficients
$f_m[n-1]$: Forward prediction errors
$b_m[n-1]$: Backward prediction errors
$r^b_m[n-1]$: Forward prediction error energy
$r^f_m[n-1]$: Backward prediction error energy
$e_m[n-1]$: Joint process error
λ : Weighting Factor

New data at time n :

$x[n]$: Input signal
$d[n]$: Reference signal

Initialisations :

$$b_0[n] = f_0[n] = x[n]$$
$$e_0[n] = d[n]$$
$$r^f_0[n] = r^b_0[n] = \lambda r^f_0[n-1] + x^2[n]$$
$$\alpha_0[n] = 1$$
$$k_m[0] = k^x_m[0] = 0$$

For $i = 0$ to $N-1$:
Predictor section

$$\begin{aligned}
k_{m+1}[n] &= \lambda k_{m+1}[n-1] + \frac{f_m[n]b_m[n-1]}{\alpha_m[n-1]} \\
f_{m+1}[n] &= f_m[n] - \frac{k_{m+1}[n]b_m[n-1]}{r^b_m[n-1]} \\
b_{m+1}[n] &= b_m[n-1] - \frac{k_{m+1}[n]f_m[n]}{r^f_m[n]} \\
r^f_{m+1}[n] &= r^f_m[n] - \frac{k^2_{m+1}[n]}{r^b_m[n-1]} \\
r^b_{m+1}[n] &= r^b_m[n-1] - \frac{k^2_{m+1}[n]}{r^f_m[n]} \\
\alpha_{m+1}[n] &= \alpha_m[n] - \frac{\alpha^2_m[n]}{r^b_m[n]}
\end{aligned}$$

Filter section

$$\begin{aligned}
k^x_{m+1}[n] &= \lambda k^x_{m+1}[n-1] + \frac{e_m[n]b_m[n]}{\alpha_m[n]} \\
e_{m+1}[n] &= e_m[n] - \frac{k^x_{m+1}[n]b_m[n]}{r^b_m[n]}
\end{aligned}$$

The algorithm is initialized, and then executed order recursively for $n = 0$. The next input and desired samples in time are then fed into the algorithm by means of an initialisation, and the algorithm is then executed for $n = 1$. This process is repeated until all the samples of interest have been processed.

This algorithm represents the single channel case, which means that only one complex valued input and one complex valued desired signal sample are processed by the algorithm at each time step.

5.3.2 Multi-Channel Lattice Algorithm

If the application requires the processing of a multi-dimensional input data, to be adapted to a multi-dimensional desired signal, at each time instant, then the cost function used in the derivation of the single channel case will have to be modified. This new cost function can then be used to derive the multi-channel lattice algorithm by following the same technique as for the derivation for the single channel case. The exponentially weighted cost function for the multi-dimensional case is given by

$$\xi[n] = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{e}[i]\ \mathbf{e}^H[i] \tag{5.69}$$

where $\mathbf{e}[i]$ denotes the error signal, which can be expanded as

$$\begin{aligned}
\mathbf{e}[i] &= \mathbf{d}[i] - \mathbf{y}[i] && (5.70) \\
&= \mathbf{d}[i] - \mathbf{w}^H[n]\mathbf{x}[i]. && (5.71)
\end{aligned}$$

where the input vector $\mathbf{x}[i]$, and the desired vector $\mathbf{d}[i]$ have dimensions of KN and L respectively. There are N input samples available in each of K input channels, and there are L simultaneous outputs, all of which are complex. The coefficient vector $\mathbf{w}[n]$ is of dimension $KN \times L$, and is chosen to minimize the trace of the cost function. The input vector format is defined as follows

$$\begin{aligned}\mathbf{x}[i] = [x_1[i], x_1[i-1], \ldots, x_1[i-N+1], x_2[i], x_2[i-1], \ldots, x_2[i-N+1], \ldots \\ x_L[i], x_L[i-1], \ldots, x_L[i-N+1]]\end{aligned} \tag{5.72}$$

where $x_n[i]$ is the ith time sample of the nth input sequence. The optimum solution for the coefficient vector is given by

$$\mathbf{w}_{opt}[n] = \mathbf{\Phi}^{-1}[n]\ \mathbf{z[n]} \tag{5.73}$$

where the $KN \times KN$ estimated correlation matrix is defined by

$$\mathbf{\Phi}[n] = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}[i]\mathbf{x}^H[i] \tag{5.74}$$

and the estimated correlation vector between the input and desired sequence by

$$\mathbf{z}[n] = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}[i]d^H[i] \tag{5.75}$$

These equations are then transformed into an order and time recursive algorithm with a lattice structure.

The final multi-channel lattice algorithm is very similar in form to the single channel case. The only difference is that the variables are extended to matrices. This algorithm will not be presented here, as it forms a subsection of the algorithm which will be presented in the next section.

5.3.3 Multi-Channel Lattice Predictor with Varying Data Lengths

In the previous section the mathematical preliminaries for the derivation of the multi-channel lattice filter were described. This section contains a discussion of the development of a multi-channel lattice filter for data records that do not have the same number of samples, and thus do not span the same amount of time. After the mathematical setup of the cost function has been described, the multi-channel lattice algorithm for non-uniform data records will be given.

For the non-uniform data record length case at hand, the same number of samples are not available in each channel for processing. This means that each channel has a unique number of samples, which will be denoted by N_i, where i is the input channel number. The length of the input vector, which was KN for the equal data record length case, is now denoted by N which is given by

$$N = \sum_{n=1}^{K} N_i. \tag{5.76}$$

The new input data vector, of dimension N, is now defined as

$$\mathbf{x}[i] = [x_1[i], x_1[i-1], \ldots, x_1[i-N_1+1], x_2[i], x_2[i-1], \ldots, x_2[i-N_2+1], \ldots \\ x_L[i], x_L[i-1], \ldots, x_L[i-N_K+1]] \qquad (5.77)$$

The exponentially weighted cost function for the multi-dimensional non-uniform data- record-length case is given by

$$\xi[n] = \sum_{i=1}^{n} \lambda^{n-i} \mathbf{e}[i]\ \mathbf{e}^H[i] \qquad (5.78)$$

where $\mathbf{e}[i]$ denotes the error signal, which can be expanded in terms of the coefficient vector as

$$\begin{aligned} \mathbf{e}[i] &= \mathbf{d}[i] - \mathbf{y}[i] && (5.79) \\ &= \mathbf{d}[i] - \mathbf{w}^H[n]\mathbf{x}[i]. && (5.80) \end{aligned}$$

The coefficient vector $\mathbf{w}[n]$ is of dimension $N \times L$, and is chosen to minimize the trace of the exponentially weighted cost function,

$$\xi[n] = \sum_{i=1}^{n} \lambda^{n-i} \mathbf{e}[i]\ \mathbf{e}^H[i] \qquad (5.81)$$

The optimum coefficient vector is then given by

$$\mathbf{w}_{opt}[n] = \mathbf{\Phi}^{-1}[n]\ \mathbf{z}[\mathbf{n}] \qquad (5.82)$$

where the definitions given in equations (5.74) and (5.75) are used for the estimated correlation matrix and vector respectively.

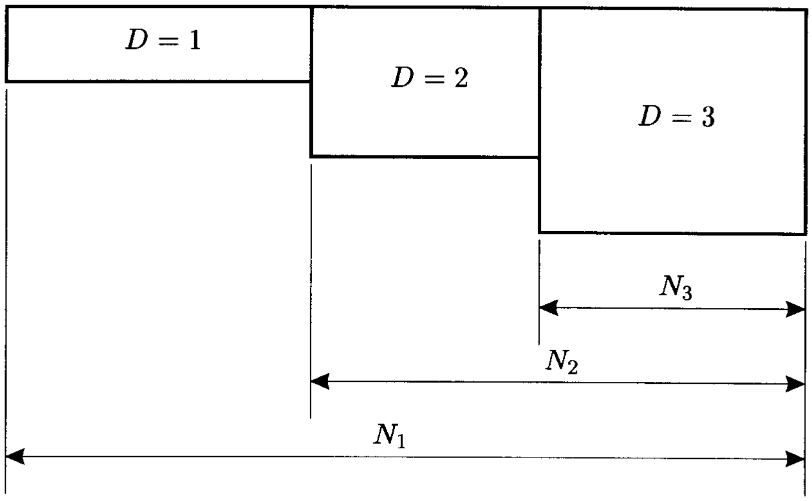

Figure 5.7: Non-uniform record length lattice predictor schematic showing increasing lattice dimension

The equation for the optimal coefficient matrix can now be made order and time recursive to give the lattice predictor for this case. The non-uniform length lattice predictor is composed of multidimensional lattice stages of increasing dimension, as diagrammatically represented in Figure 5.8.

This is the final form of the multi-dimensional lattice predictor which will be necessary for the implementation of the adaptive equaliser for the multi-dimensional modem. The specifications in terms of the data lengths, and the roles of the variables in an equalisation setup will be presented in the next section. The interface between the modem and the equaliser, and the specific equaliser algorithm will also be discussed in the next section.

5.4 MULTI-CHANNEL DECISION FEEDBACK LATTICE EQUALISATION

This section makes use of the results derived in the previous section, as well as the analysis of the effect of multipath on the received Q^2PSK signal to propose a specific form of the multi-dimensional RLS lattice filter as an adaptive method of generating the inverse model of the channel. The adaptive lattice filter is thus used to equalise the channel in terms of phase and frequency response.

5.4.1 Use of Varying Data Length Structure as a Decision Feedback Equaliser

As previously discussed in this chapter, a decision feedback implementation of an equaliser exhibits a significant performance improvement over linear equalisers in terms of bit error probability and can mitigate more severe channel distortion. The use of a DFE thus extends the operating envelope of a modem system in terms of the number of channels which can be used for reliable communication. In a transversal type equaliser, the mean delay is setup to be in the middle of the feed forward section. This means that there are samples before and after (in time) the sample being processed. If decisions are made on the output of this structure to be fed into a feedback section, then it stands to reason that the data record length of the feed forward section will be longer than that of the feedback section, if it is assumed that the feed forward and feedback sections span enough time to account for the maximum time dispersion of the channel. From this principle it can be seen that if the DFE is converted to a lattice structure, then use will have to be made of a lattice structure with two non-equal data record lengths: the first one being the data length of the feed forward section, and the second being the length of the feedback section. The question as to the dimensionality of the lattice algorithm still remains, and will be discussed in the next section.

5.4.2 Integration with Receiver Structure

For the multidimensional modulation technique encompassed by Q^2PSK use will have to be made of a multidimensional equaliser with feed forward dimension equal to that of the modulation technique, and a decision feedback section of twice the dimension of the modulation technique. Due to the fact that the outputs of the matched filters can be combined to form complex samples, the dimensions of the equaliser can be reduced to half of their original value. Thus, if use is made of 4-D modulation, and D_{FF} and D_{FB} are used to denote the dimension of the signals in the feed forward and feedback sections respectively, then the following will be necessary

$$D_{FF} = 2 \tag{5.83}$$
$$D_{FB} = 4 \tag{5.84}$$

It can also be seen from the analysis of the effect of multipath on the output of the matched filters given in Chapter 2 that energy from all the dimensions can be present in the output of any of the matched filters. This implies that the equaliser cannot be partitioned in any way to reduce the complexity, and thus the computational burden involved in implementing this structure cannot be reduced.

The above discussion has defined the dimensionality necessary for the two stages of the equaliser. It was decided to limit the time span of the equaliser to seven symbols in the forward direction, and three in the feedback dimension. If the feed forward time span is represented by N_1, and the feedback time span by N_2, then

$$N_1 = 7 \tag{5.85}$$
$$N_2 = 3 \tag{5.86}$$

This means that there will be four 2-dimensional lattice sections, followed by three 4-dimensional lattice sections. This choice of equaliser time span was mainly due to simulation time considerations, but is also relatively realistic for a 40 *kbit/s* modem operating over a V/UHF channel. Now that the equaliser has been fully specified, all that remains is to define the interface between the demodulator and the equaliser. This is relatively simple, as the outputs of the two matched filters with the same pulse shape, but different carriers, are sampled at the optimum points determined by the synchronization correlator, and then grouped together to form a complex sample. There are thus two complex samples generated per received symbol. These two complex samples are formed into a 2×1 vector, which is then used as the input vector, $\mathbf{x}[n]$, at time n to the lattice filter. The lattice filter output, $\mathbf{y}[n]$, and the desired vector, $\mathbf{d}[n]$ have the

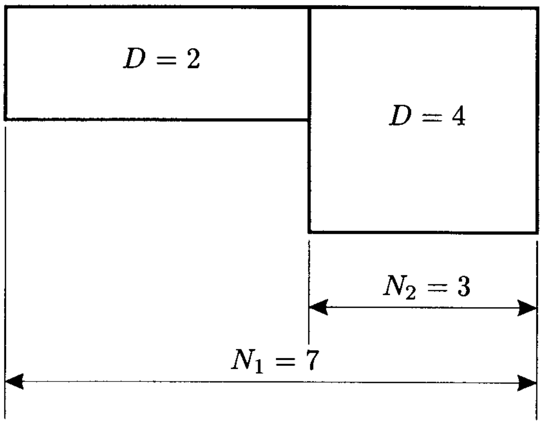

Figure 5.8: Non-uniform record length lattice predictor sections for decision feedback equalisation application.

same dimensions as the input vector, as they also represent a received 4-dimensional symbol. These dimensions are also given in Figure 5.8.

The detail derivation of the lattice algorithm is long and intricate, but does not give more insight into the problem. Use was made of an algebraic approach similar to that used for 1- and 2-dimensional lattice structures in [71, 72, 83, 85], but with the mathematical setup discussed in Paragraph 5.3.3. Only the final multidimensional lattice algorithm will thus be presented here, and is given by:

Algorithm 5.2 The Multidimensional RLS Lattice Algorithm

Available at time n :

$\mathbf{k}_m[n-1]$	: Reflection coefficient
$\mathbf{k}^x_m[n-1]$	: Filter coefficients
$\mathbf{k}^b_M[n-1]$	: Transition stage coefficient
$\mathbf{f}_m[n-1]$	: Forward prediction errors
$\mathbf{b}_m[n-1]$	: Backward prediction errors
$\hat{\mathbf{b}}_M[n-1]$	: Transition stage backward prediction error
$\mathbf{r}^b_m[n-1]$	: Forward prediction error energy
$\mathbf{r}^f_m[n-1]$	: Backward prediction error energy
$\mathbf{r}^b_M[n-1]$	: Transition stage forward prediction error energy
$\mathbf{r}^f_M[n-1]$	: Transition stage backward prediction error energy
$\mathbf{e}_m[n-1]$	: Joint process error
λ	: Weighting Factor

New data at time n :

$\mathbf{x}[n]$	: Input signal
$\mathbf{d}[n]$	: Reference signal

Initialisations :

$$\mathbf{b}_0[n] = \mathbf{f}_0[n] = \mathbf{x}[n]$$
$$\mathbf{e}_0[n] = \mathbf{d}[n]$$
$$\mathbf{r}^f_0[n] = \mathbf{r}^b_0[n] = \lambda \mathbf{r}^f_0[n-1] + \mathbf{x}[n]\mathbf{x}^H[n]$$
$$\alpha_0[n] = 1$$
$$\mathbf{k}_m[0] = \mathbf{k}^x_m[0] = \mathbf{k}^b_M[0] = \mathbf{0}$$
$$\mathbf{r}^f_m[0] = \mathbf{r}^b_m[0] = \delta\mathbf{I}$$

Two-Dimensional Lattice Stages (For $0 < m < N_1 - N_2$)

$$\begin{aligned}
\mathbf{f}_m[n] &= \mathbf{f}_{m-1}[n] - \mathbf{k}^H_m[n-1]\,\mathbf{r}^{-b}_{m-1}[n-2]\,\mathbf{b}_{m-1}[n-1]\\
\mathbf{b}_m[n] &= \mathbf{b}_{m-1}[n-1] - \mathbf{k}_m[n-1]\,\mathbf{r}^{-f}_{m-1}[n-1]\,\mathbf{f}_{m-1}[n]\\
\mathbf{k}_m[n] &= \lambda\,\mathbf{k}_m[n-1] + \alpha_{m-1}[n-1]\,\mathbf{b}_{m-1}[n-1]\,\mathbf{f}_{m-1}[n]\\
\mathbf{r}^f_m[n] &= \lambda\,\mathbf{r}^f_m[n-1] + \alpha_m[n-1]\,\mathbf{f}_m[n]\,\mathbf{f}^H_m[n]\\
\mathbf{r}^b_m[n] &= \lambda\,\mathbf{r}^b_m[n-1] + \alpha_m[n]\,\mathbf{b}_m[n]\,\mathbf{b}^H_m[n]\\
\alpha_m[n] &= \alpha_{m-1}[n] + \alpha^2_{m-1}[n]\,\mathbf{b}^H_{m-1}[n]\,\mathbf{r}^{-b}_{m-1}[n-1]\,\mathbf{b}_{m-1}[n]
\end{aligned}$$

Two-Dimensional Filter Section

$$\begin{aligned}
\mathbf{y}_m[n] &= \mathbf{y}_{m-1}[n] + \mathbf{k}^{xH}_m[n-1]\,\mathbf{r}^{-b}_{m-1}[n-1]\,\mathbf{b}_{m-1}[n]\\
\mathbf{e}_m[n] &= \hat{\mathbf{y}}[n] - \mathbf{y}_m[n]\\
\mathbf{k}^x_m[n] &= \lambda\,\mathbf{k}^x_m[n-1] + \alpha_{m-1}[n]\,\mathbf{b}_{m-1}[n]\,\mathbf{e}^H_{m-1}[n]
\end{aligned}$$

Transitional Lattice Stage ($M = N_1 - N_2$)

$$\begin{aligned}
\hat{\mathbf{b}}_M[n] &= \mathbf{e}_{M-1}[n-1] - \mathbf{k}^b_M[n-1]\,\mathbf{r}^{-f}_{M-1}[n-1]\,\mathbf{f}_{M-1}[n]\\
\mathbf{k}^b_M[n] &= \lambda\,\mathbf{k}^b_M[n-1] + \alpha_{M-1}[n-1]\,\mathbf{e}_{M-1}[n-1]\,\mathbf{f}_{M-1}[n-1]\\
\mathbf{f}^{'}_M[n] &= \begin{bmatrix} \mathbf{f}_M[n] \\ \mathbf{e}_M[n-1] \end{bmatrix}\\
\mathbf{b}^{'}_M[n] &= \begin{bmatrix} \mathbf{b}_M[n] \\ \hat{\mathbf{b}}_m[n] \end{bmatrix}\\
\mathbf{r}^f_M[n] &= \lambda\,\mathbf{r}^f_M[n-1] + \alpha_M[n-1]\,\mathbf{f}^{'}_m[n]\,\mathbf{f}^{'H}_m[n]\\
\mathbf{r}^b_M[n] &= \lambda\,\mathbf{r}^b_M[n-1] + \alpha_M[n]\,\mathbf{b}^{'}_m[n]\,\mathbf{b}^{'H}_m[n]
\end{aligned}$$

Four-Dimensional Lattice Stages (For $N_1 - N_2 < m < N_1$)

$$\begin{aligned}
\mathbf{f}_m[n] &= \mathbf{f}_{m-1}[n] - \mathbf{k}^H_m[n-1]\,\mathbf{r}^{-b}_{m-1}[n-2]\,\mathbf{b}_{m-1}[n-1]\\
\mathbf{b}_m[n] &= \mathbf{b}_{m-1}[n-1] - \mathbf{k}_m[n-1]\,\mathbf{r}^{-f}_{m-1}[n-1]\,\mathbf{f}_{m-1}[n]\\
\mathbf{k}_m[n] &= \lambda\,\mathbf{k}_m[n-1] + \alpha_{m-1}[n-1]\,\mathbf{b}_{m-1}[n-1]\,\mathbf{f}_{m-1}[n]\\
\mathbf{r}^f_m[n] &= \lambda\,\mathbf{r}^f_m[n-1] + \alpha_m[n-1]\,\mathbf{f}_m[n]\,\mathbf{f}^H_m[n]\\
\mathbf{r}^b_m[n] &= \lambda\,\mathbf{r}^b_m[n-1] + \alpha_m[n]\,\mathbf{b}_m[n]\,\mathbf{b}^H_m[n]\\
\alpha_m[n] &= \alpha_{m-1}[n] + \alpha^2_{m-1}[n]\,\mathbf{b}^H_{m-1}[n]\,\mathbf{r}^{-b}_{m-1}[n-1]\,\mathbf{b}_{m-1}[n]
\end{aligned}$$

Four-Dimensional Filter Section

$$\begin{aligned}
\mathbf{y}_m[n] &= \mathbf{y}_{m-1}[n] + \mathbf{k}^{xH}_m[n-1]\,\mathbf{r}^{-b}_{m-1}[n-1]\,\mathbf{b}_{m-1}[n]\\
\mathbf{e}_m[n] &= \hat{\mathbf{y}}[n] - \mathbf{y}_m[n]\\
\mathbf{k}^x_m[n] &= \lambda\,\mathbf{k}^x_m[n-1] + \alpha_{m-1}[n]\,\mathbf{b}_{m-1}[n]\,\mathbf{e}^H_{m-1}[n]
\end{aligned}$$

A detailed block diagram of the structure of this algorithm is given in Figure 5.9, where thicker lines have been used to represent the 4-dimesional signal lines. This diagram has one less two-dimensional stage and one less 4-dimensional stage, so that the diagram can fit on one page. The time index of some of the signal labels in the figure have left out for readability. Some of the weights have been redefined as folows :

$$\begin{aligned}
\mathbf{k}_m^f[n] &= \mathbf{k}_m^H[n-1]\,\mathbf{r}_{m-1}^{-b}[n-2] && (5.87)\\
\mathbf{k}_m^b[n] &= \mathbf{k}_m[n-1]\,\mathbf{r}_{m-1}^{-f}[n-1] && (5.88)\\
\mathbf{k}_m^x[n] &= \mathbf{k}_m^{xH}[n-1]\,\mathbf{r}_{m-1}^{-b}[n-1] && (5.89)\\
\mathbf{k}_M^B[n] &= \mathbf{k}_M^b[n-1]\,\mathbf{r}_{M-1}^{-f}[n-1] && (5.90)
\end{aligned}$$

In this section the parameters of the multidimensional RLS lattice filter to be used as an adaptive decision feedback equaliser in the 4-dimensional Q^{2}PSK modem platform have been specified, as well as the interface between the demodulator and the equaliser. The operation of the equaliser with respect to the frame structure will be discussed in the next section. Specifically the problem of the generation of the desired vector will be considered.

5.4.3 Equaliser Modes of Operation

Up to this stage the only point that has been neglected is the question of where the desired vector $\mathbf{d}[n]$ is obtained from ? A stored, non-distorted replica of the synchronisation sequence is used as the desired vector input to train the equaliser at the start of every frame. After training the equaliser processes the random data in a free running mode where decisions on the equaliser output vector are used as the desired vector. This is referred to as the "decision directed" mode of operation.

Training Mode

The adaptive equaliser has to be trained to build up an inverse model of the channel. The training is accomplished by transmitting a known sequence of symbols over the channel, which are collectively known as the "training sequence". These symbols are received by the demodulator, and stored in a buffer. As this sequence is known, a replica of it can be stored at the receiver. Once the whole training sequence has been received, it can be used as the input vector to the equaliser, and the replica can be used as the desired vector.

In the Q^{2}PSK modem platform the synchronisation sequence doubles as the training sequence. As the correlator searches for the zero-time reference, the output samples of the matched filters are written into a buffer. Once the zero-time reference has been located, the training sequence is buried somewhere in the buffer. This is due to the fact that the peak search algorithm on the output of the correlator checks several symbols after a valid peak has bean detected to make sure that no higher peaks exist. The last symbol of the received training sequence is thus at a position in the buffer which is given by the number of look-ahead symbols plus one. Given this position, it is possible to extract the rest of the training sequence by sampling the buffer

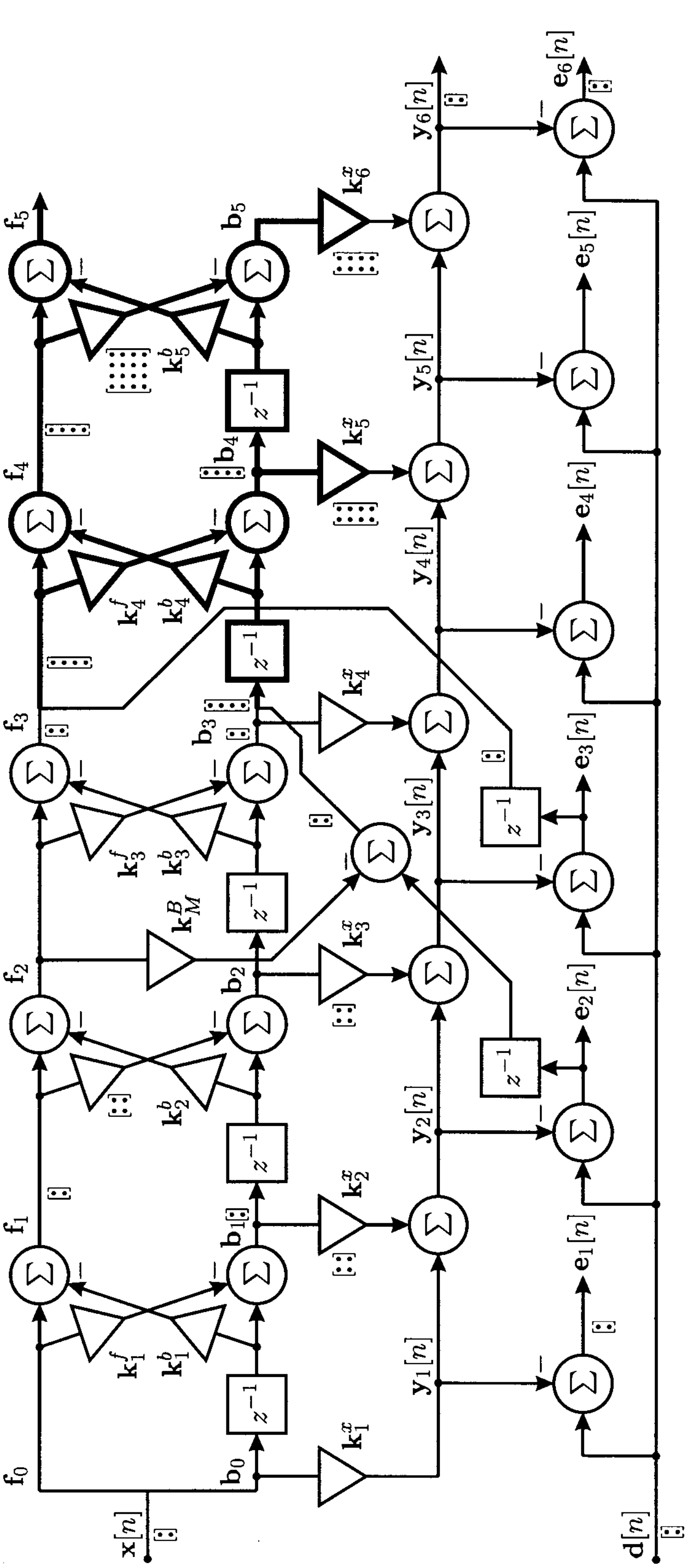

Figure 5.9: Detailed block diagram of lattice structure for 4-D decision feedback equalisation

once per symbol (every 10 samples in the current setup is a symbol as the samples per symbol is equal to ten ($sps = 10$)).

Now that the received training sequence has been extracted, the training of the equaliser can commence. It is desired to develop the main coefficient of the equaliser at a position corresponding to the center of the equaliser, which is at the 4th lattice stage in this case. The stored replica thus has to be delayed by 4 symbols relative to the received sequence to develop the coefficient at the correct time offset within the lattice structure. The length of the training sequence is only 31 symbols, which is not enough to ensure that the adaptive equaliser has converged, so use is made of a cyclic training technique. Once the last symbol of either sequence has been used, the sequence position variable is switched back to the start of the sequence. The sequences are run through the equaliser 16 times in this mode to achieve satisfactory training of the equaliser. Once this has been completed the equaliser can start processing the data payload contained in the frame. An example of the trace of the squared error of the equaliser during cyclic training is shown in Figures 5.10 and 5.11. In both these figures periodic “spikes” can be seen, which indicate the switch over from the end of the training sequence, back to the beginning. It can also be seen that square error for the multipath case is slightly larger than that of the AWGN case.

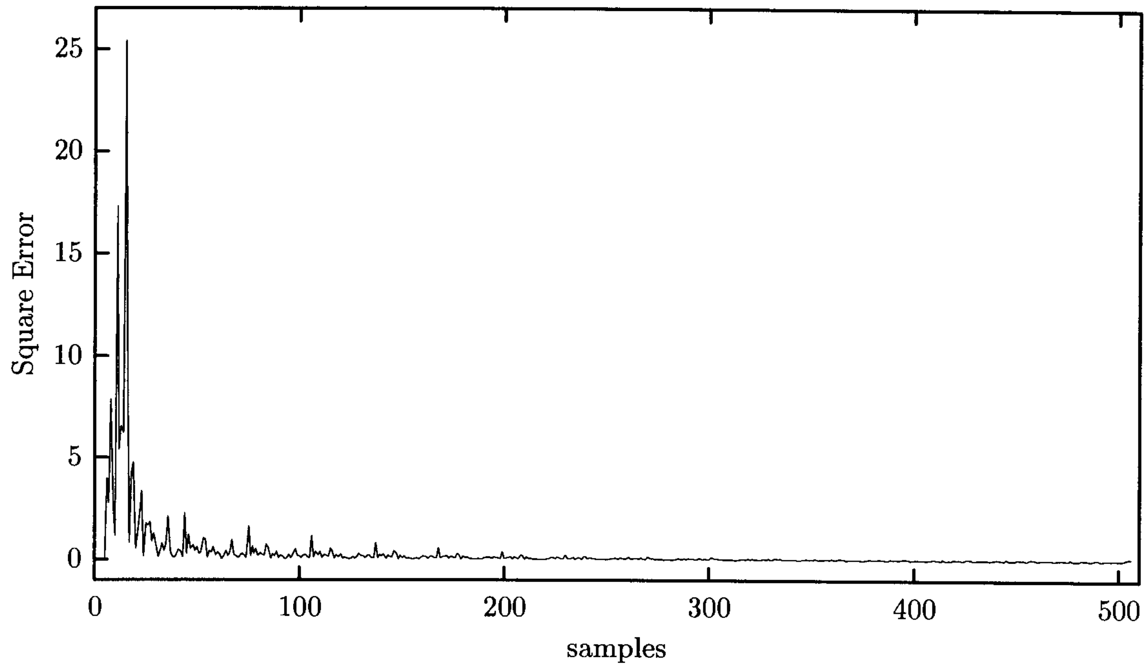

Figure 5.10: Error over cyclic training, no multipath, $E_b/N_0 = 20$ dB

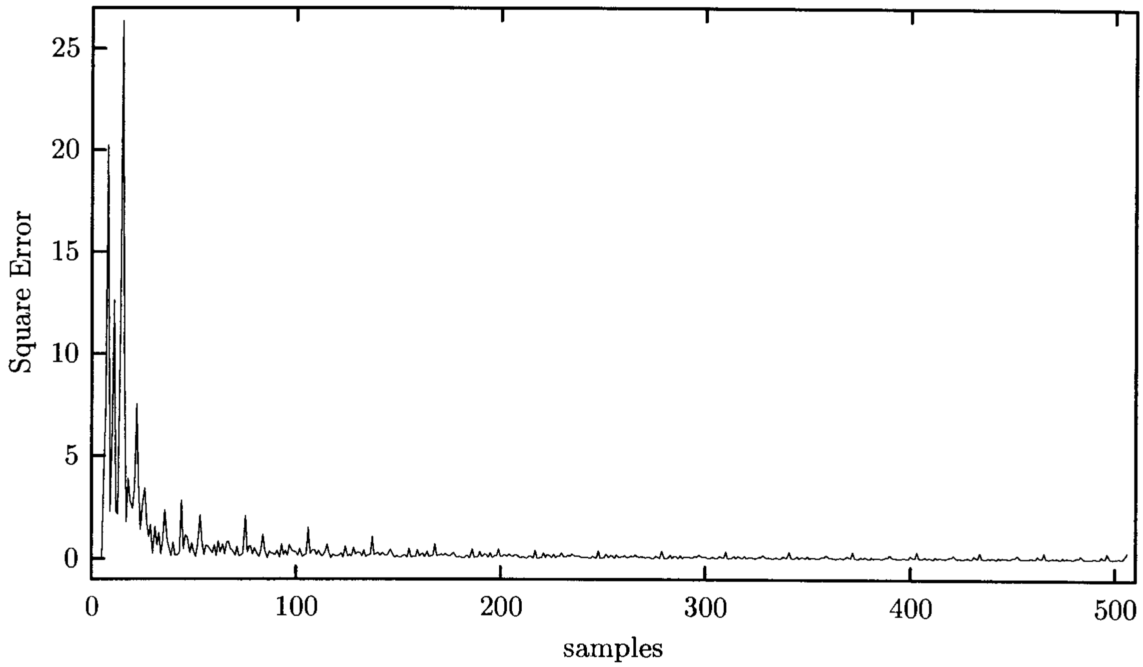

Figure 5.11: Error over cyclic training, with multipath, $E_b/N_0 = 20$ dB

Data Directed Mode

Once the equaliser has been trained successfully, it is switched into a so-called "data directed" mode. In this mode the output of the matched filters is sampled at the optimum position determined by the frame/symbol synchronization algorithm. This sample, which represents one symbol, is used as the input vector to the equaliser. The equaliser processes this symbol with the internal coefficients carried over from the previous symbol, to give an equalised output symbol. It is thus assumed that the channel can only change by a small amount during one symbol period. A decision is made on the output symbol, and this decision is then used as the desired vector with which the equaliser coefficient matrix is then updated.

An example of the trace of the squared error of the equaliser during cyclic training is shown in Figures 5.12 and 5.13. These graphs show the square error during training for the first 506 samples, and then the square error in data directed mode for the remaining samples. The switch over point, from training mode to data directed mode, is visible due to a slight increase in the square error just after the switch over point (sample 506). This is due to the fact that the equaliser was trained on a limited data set, which implies that its inverse model will be a very accurate fit to the training set, but that this will not necessarily be as close as possible to the true inverse model of the channel.

5.4.4 Description of Simulation Software

The whole simulation system was written in C++, and comprises approximately 9800 lines of code in total. Most of the signal processing objects have been written as templates, which means that they describe only the functionality of the object and are not linked to a specific data type. This in turn implies that the effects of fixed point implementation can be evaluated in simulation by adding a fixed point class, and then instantiating the modem with the fixed point class instead of the usual floating point "double" data type.

The adaptive filter class is based on an underlying matrix class which was developed first. This means that the equations for the lattice filter can be programmed in C++ in a form that is very close to the actual algorithm given in this chapter. The listing of the portion of code for the adaptive lattice filter is given in Appendix B.

The whole simulation is setup and controlled by means of configuration files. This aids in keeping track of exactly which experiments have been performed, and where the results were stored. A set of configuration files can also be used to execute batch runs of the simulation for different channel conditions.

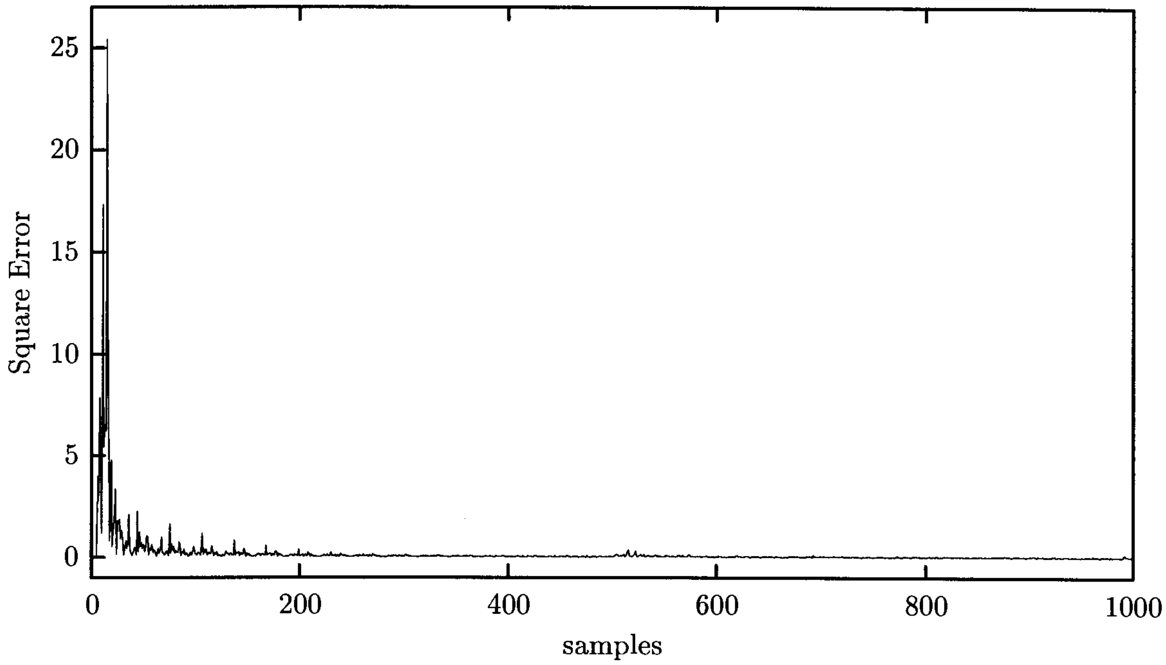

Figure 5.12: Cyclic training and data directed mode error, no multipath, $E_b/N_0 = 20$ dB

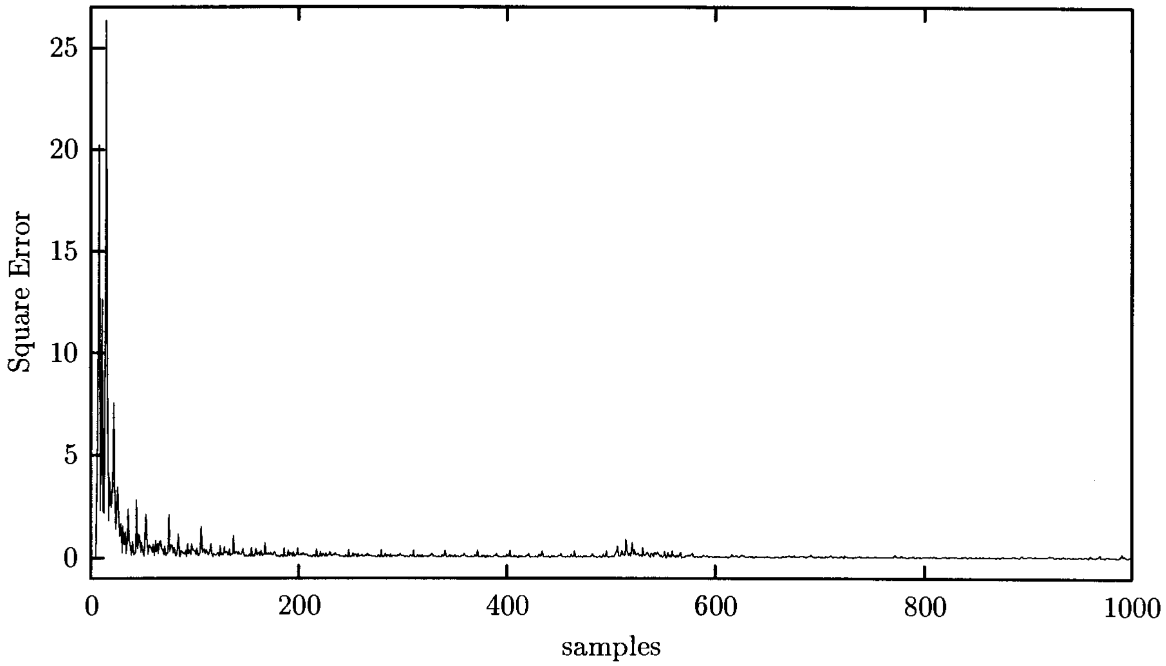

Figure 5.13: Cyclic training and data directed mode error, with multipath, $E_b/N_0 = 20$ dB

5.5 CONCLUSION

In this chapter the multidimensional lattice equaliser with non-uniform data record lengths has been developed for use as an adaptive equaliser for the 4-dimensional Q^{2}PSK modulation technique. The concept of an equaliser was introduced from an inverse modelling point of view, and it was shown that this technique can be used to compensate for amplitude and phase distortion of a signal which has passed through a dispersive channel.

The method of least squares was discussed as a method of predicting a given signal from a sequence of samples from another signal which is correlated in some way to the given signal. The lattice structure was shown to be a more efficient method for the implementation of the predictor. Up until this point it was assumed that the statistics of the signals were known. This restriction has to be lifted in the practical equalisation situation, and the exponentially weighted recursive least squares (RLS) algorithm was discussed as a prediction algorithm which estimates the statistics as they progress in time. This also implies that the algorithm can be used in non-stationary environments such as the mobile channel. This algorithm was extended to a multidimensional lattice form for non-uniform data record lengths.

The interface between the modulator and the equaliser was described, which led to the final specification on the necessary input/output matrix dimensions for the equaliser, as well as the length or time span of the equaliser. Once the equaliser was fully specified the final lattice algorithm was presented. Finally the operational modes of the equaliser were discussed in lieu of the frame structure of the transmission protocol, and examples of the square error were given for each mode.

This chapter concludes the design of the receiver structure. In the following chapter the performance of the equaliser will be evaluated on stationary and fading frequency selective channels.

CHAPTER 6

RESULTS

This chapter presents the performance of multidimensional modem platform, as well as the multidimensional lattice decision feedback equaliser. Results will be presented for the following channel scenarios:

- AGWN only,
- Carrier frequency offset in AWGN,
- Static multipath and AWGN
- Fading multipath channel and AWGN

6.1 AWGN RESULTS

The section presents the results obtained in the presence of AWGN without any other form of distortion of the transmitted signal. The purpose is to check that the noise source scaling factor is correctly set to achieve the desired $\frac{E_b}{N_0}$. The calculation of this gain is discussed in detail in [126], and is given by

$$\begin{aligned} k_n &= \left[\left(\frac{2n}{I}\right)\left(10^{\frac{E_b}{N_0}[\mathrm{dB}]}\right)\right]^{-\frac{1}{2}} && (6.1)\\ &= \left[\left(\frac{8}{10}\right)\left(10^{\frac{E_b}{N_0}[\mathrm{dB}]}\right)\right]^{-\frac{1}{2}} && (6.2)\end{aligned}$$

where I is the interpolation factor, n is the number of bits per symbol, for a noise source with unity variance, and a unity power received signal (measured at output of the channel).

The results of this test are depicted in Figure 6.1. The simulation was run for E_b/N_0 values from 0dB to 11dB in 1dB increments. From this graph it can be seen that the bit error probability measured at the output of the matched filters lies on the theoretical bit error probability curve for Q^2PSK. The very slight variations of the simulation points around the theoretical curve are due to the limited run length of the simulation.

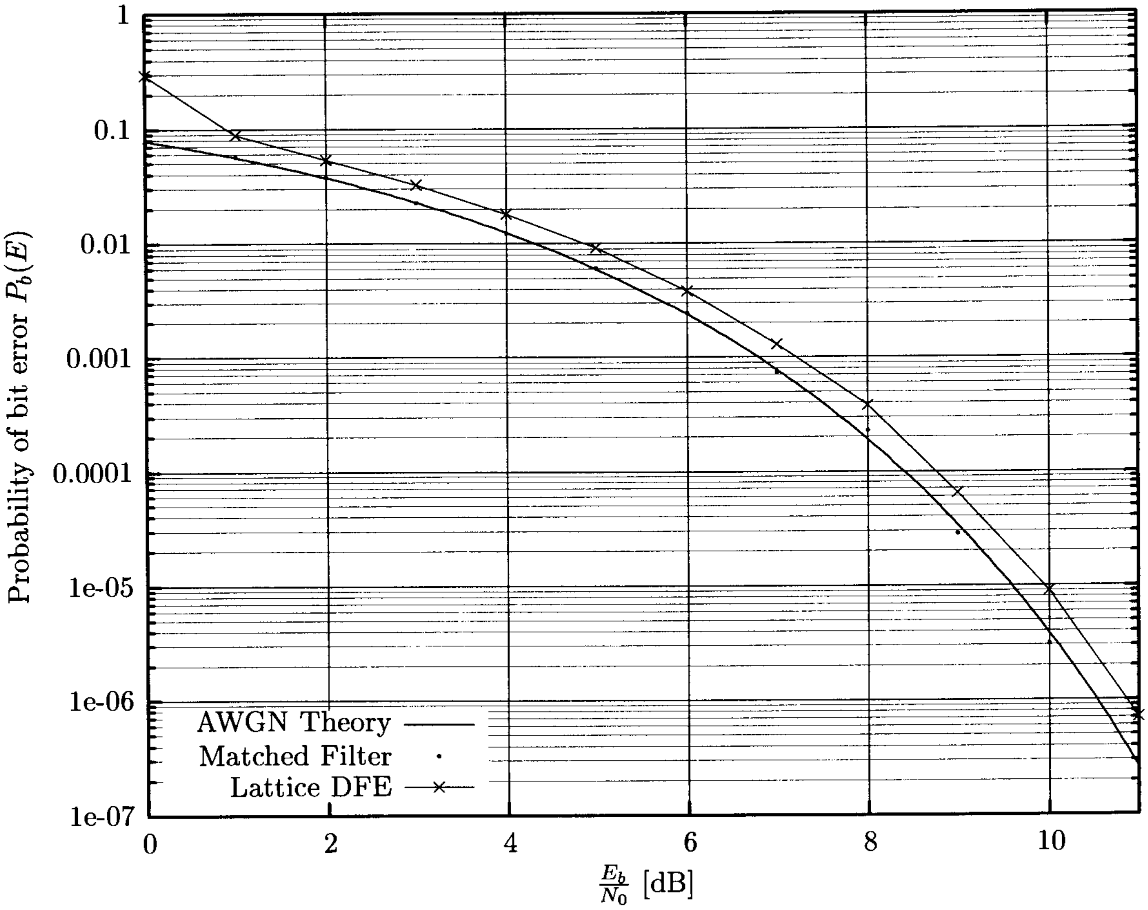

Figure 6.1: AWGN Result

The bit error probability test for the equaliser shows about a 1dB degradation over the matched filter case for $P_b > 7 \times 10^{-2}$. This is due to the fact that the number of incorrect decisions in the feedback part of the equaliser is large enough to induce a significant loss in performance. For $P_b < 10^{-2}$ it can be seen that the equaliser only performs 0.4dB worse than the matched filter case. This is due to the noise in the system, which causes the equaliser to form an inverse model which still has some variance on its parameters. This implies that the equaliser forms a inverse model which is not exactly matched to the channel model. The channel model in this case is distortionless, so the equaliser actually introduces distortion through its modeling inaccuracy.

This test thus shows that the receiver structure is optimal, and the objective of achieving the theoretical performance for the matched filter case has been achieved. The "insertion loss" of the equaliser has also been measured, and found to be 0.4dB for $P_b < 10^{-2}$.

6.2 CARRIER RECOVERY RESULTS

The RLS digital phase locked loop was evaluated for three frequency offsets, 0 Hz, 100Hz and 250 Hz, and the BEP results in AWGN are plotted in Figure 6.2 below.

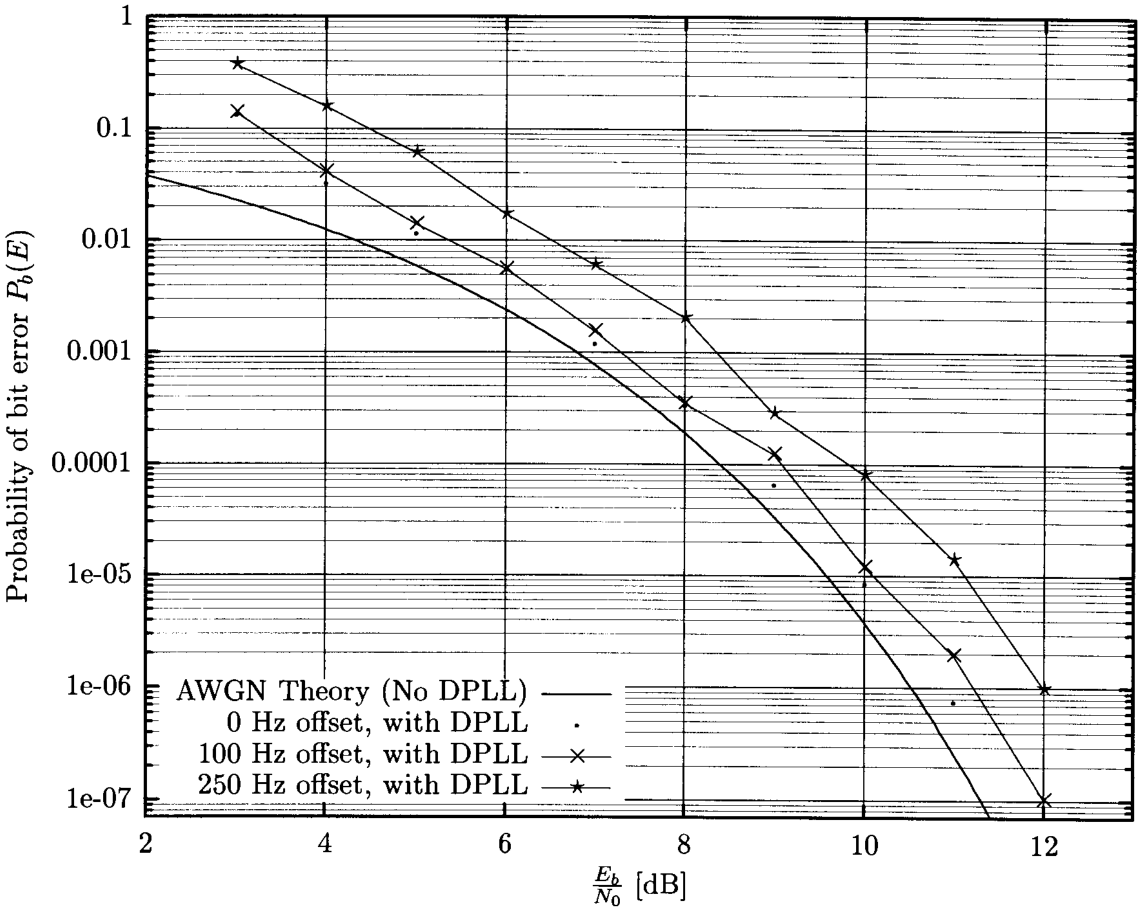

Figure 6.2: Carrier recovery results for various frequency offsets

From these results it can be seen that this carrier recovery technique is successful, especially in the light of the fact that without the DPLL, the BEP for all E_b/N_0 is 0.5. The results show that the DPLL induces a 0.3 dB loss for the 0 Hz offset case, a 0.6 dB loss for the 100 Hz offset case and a 1.6 dB loss for the 250 Hz offset case. This is deemed to be sufficient for the application at hand, which will only have relatively low Doppler offsets.

6.3 STATIC MULTIPATH RESULTS

In this section results are presented for the static multipath tests performed using the multidimensional modem simulation platform in conjunction with the multidimensional lattice equaliser. The word "Static" implies that the multipath environment is not changing as a function of time. Use was made of a two path model as discussed in Paragraph 3.6. Each bit error probability plot shows the results for the matched filter as well as the equaliser. For each of these cases minimum phase (MP)and non-minimum phase (NMP) tests were conducted.

6.3.1 Symbol Spaced Case

In this section the results for symbol spaced multipath are presented. These results all pertain to two-path multipath channels, where the weaker path has a relative strength denoted by α relative to the unity strength main path. These paths are then normalized to maintain unity power at the channel output.

Results for $\alpha = 0.2$

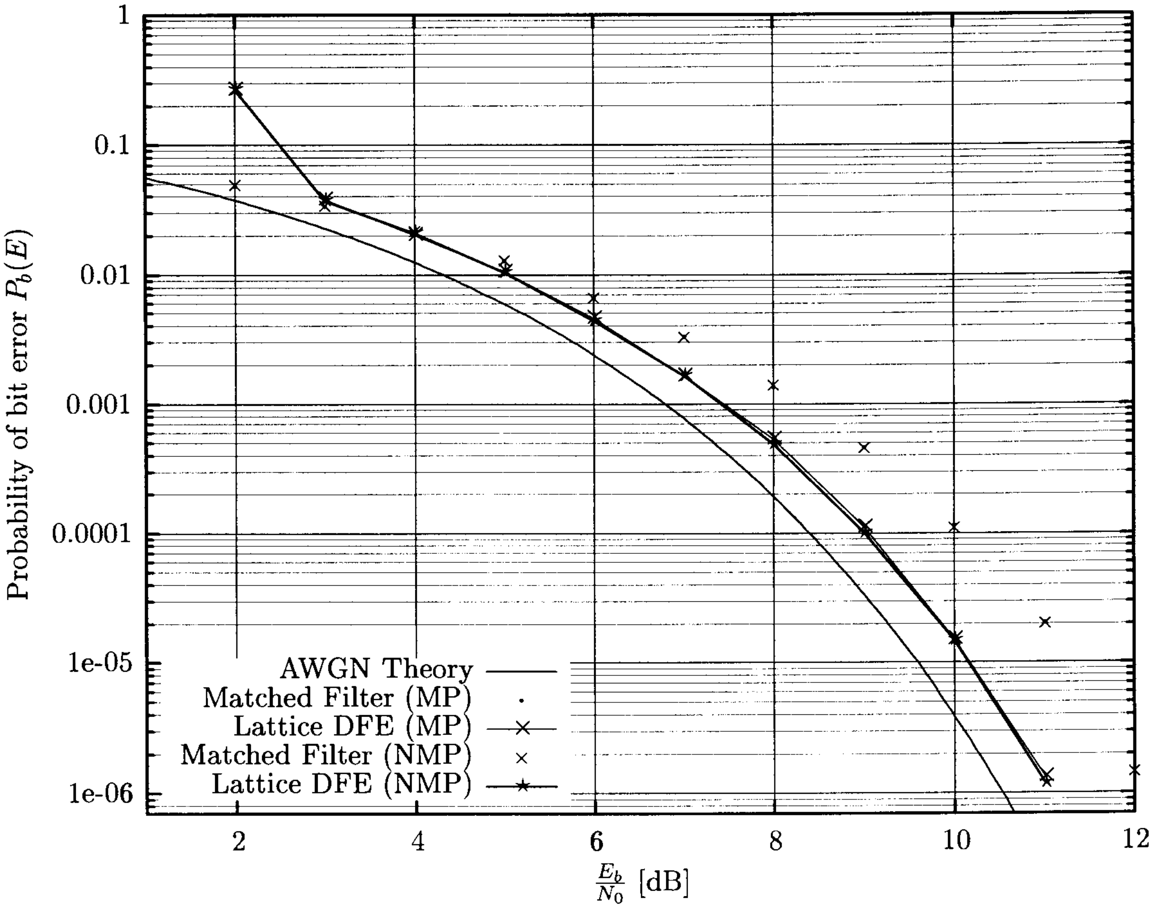

Figure 6.3: Static multipath result for $\alpha = 0.2$, $\tau = T_{symbol}$.

In Figure 6.3 it can be seen that both the matched filter cases lie on nearly the same points. This trend will continue for all of the static multipath tests. The DFE results are only slightly degraded as compared to the AWGN case in Figure 6.1. The DFE performs worse than the matched filter case for low signal to noise ratios. This is due to the error propagation effect in the feedback part of the equaliser.

Results for $\alpha = 0.5$

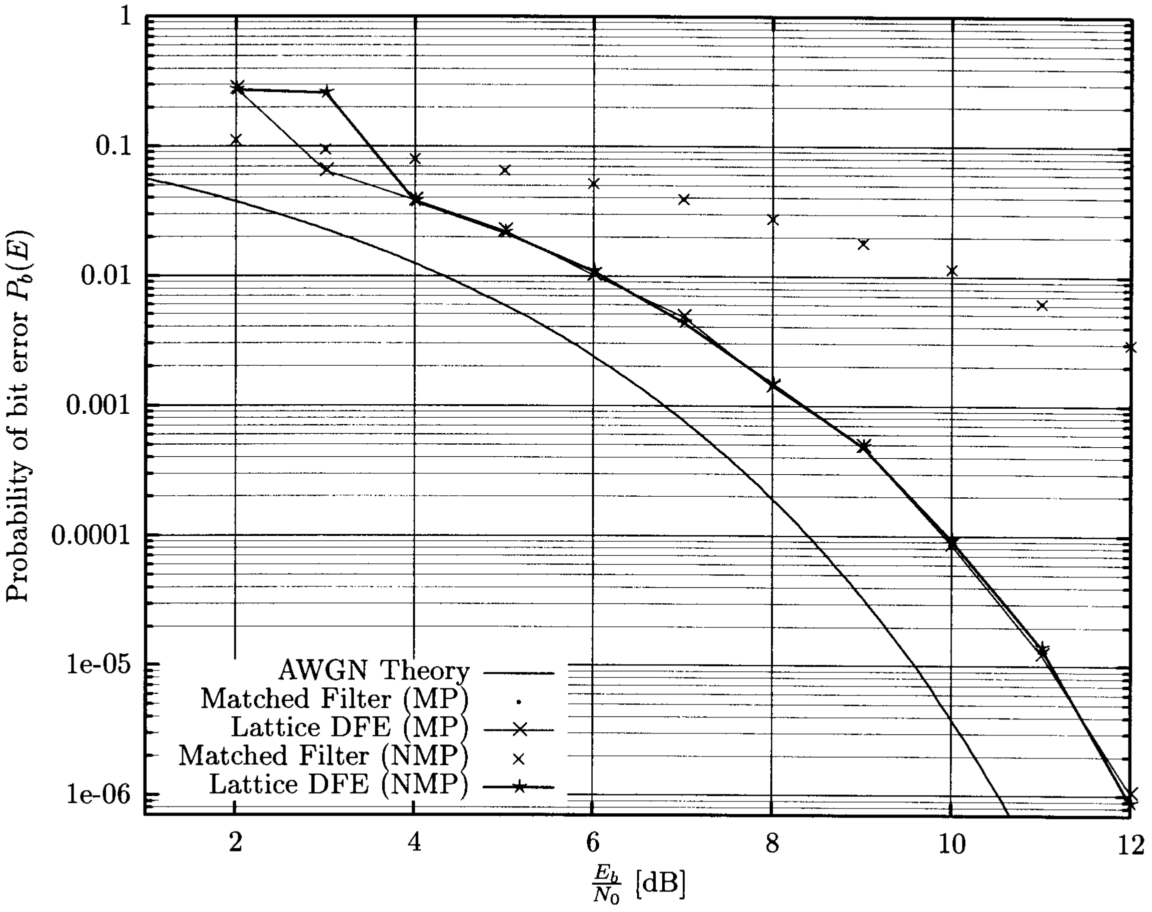

Figure 6.4: Static multipath result for $\alpha = 0.5$, $\tau = T_{symbol}$.

In Figure 6.4 it can once again be seen that the matched filter results are nearly identical. The DFE performance is still much better than that of the matched filters, but the DFE performance has degraded relative to the $\alpha = 0.2$ case of Figure 6.3, as would be expected. The error propagation effect can also be seen to have increased relative to the $\alpha = 0.2$ case.

Results for $\alpha = 0.8$

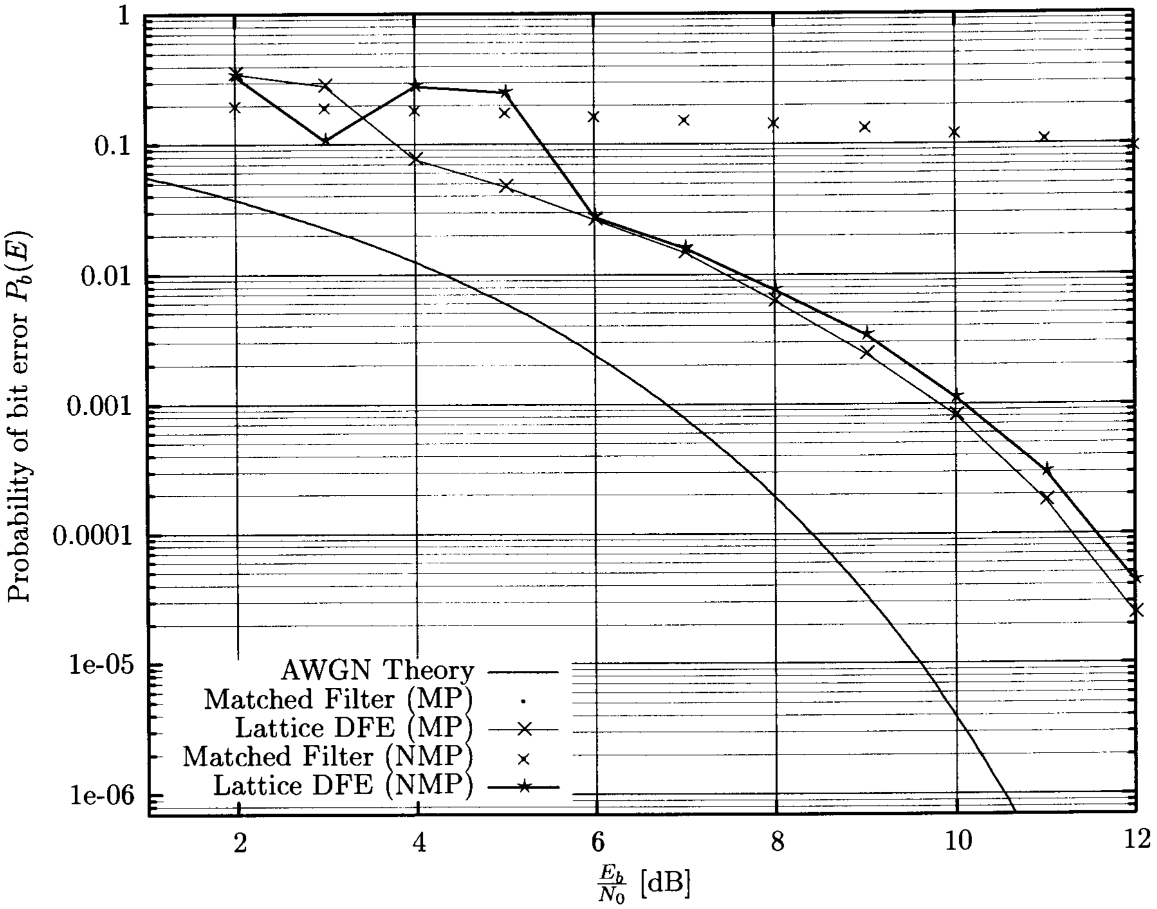

Figure 6.5: Static multipath result for $\alpha = 0.8$, $\tau = T_{symbol}$.

In Figure 6.5 it can once again be seen that the matched filter results are nearly identical, and that the matched filter BER hardly improves with increasing SNR. The DFE performance is still much better than that of the matched filters, but the DFE performance has degraded relative to the $\alpha = 0.5$ case of Figure 6.4. The error propagation effect has once again increased relative to the $\alpha = 0.5$ case.

Results for $\alpha = 1.0$

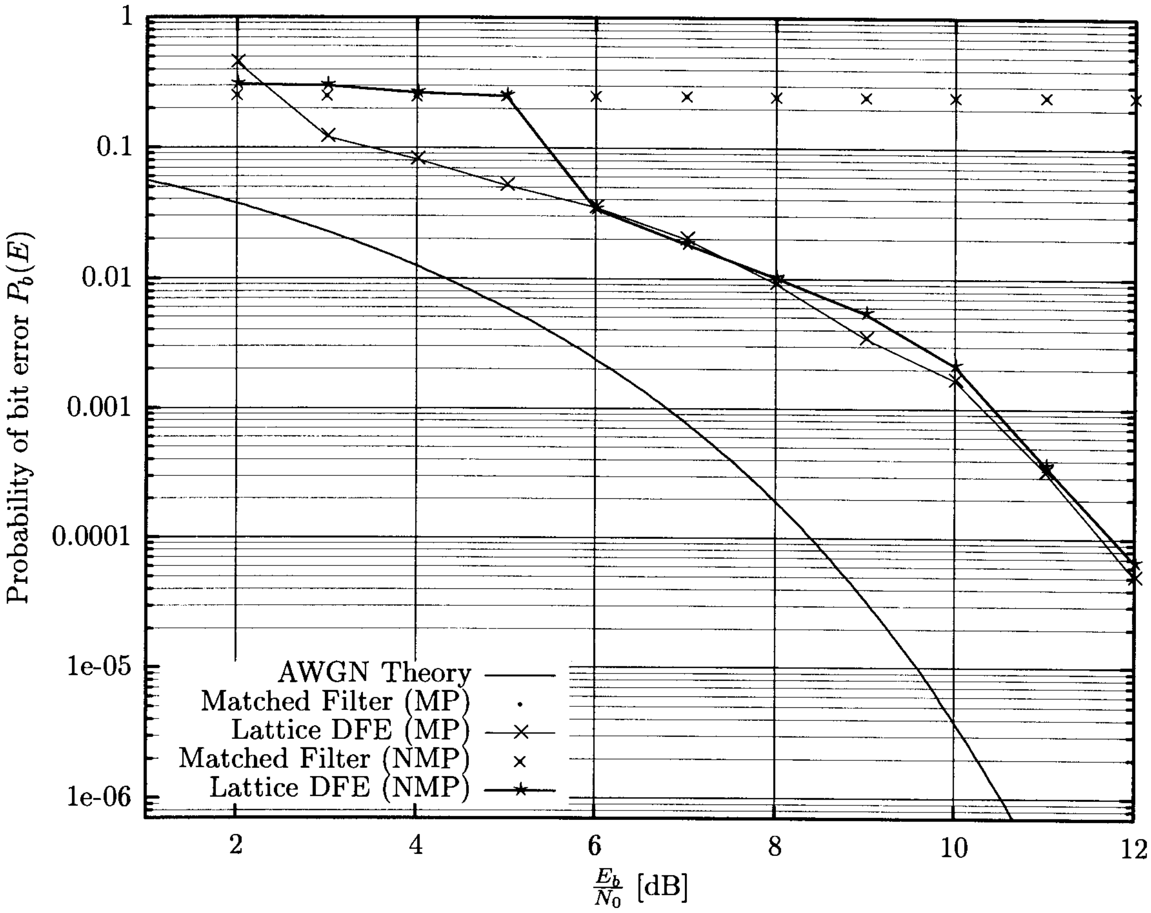

Figure 6.6: Static multipath result for $\alpha = 1.0$, $\tau = T_{symbol}$.

The $\alpha = 1.0$ is the most difficult case for the equaliser, as both multipaths have the same amplitude. This implies an infinitely deep zero in the received signal. In Figure 6.6 be seen that the matched filter are slightly worse than the $\alpha = 0.8$ case. The DFE performance is still much better than that of the matched filters, but the DFE performance has degraded relative to the $\alpha = 0.8$ case of Figure 6.5. The error propagation effect has not increased relative to that of the $\alpha = 0.8$ case.

6.3.2 Non-Symbol Spaced Results

In this section the results for various non-symbol spaced multipath cases will be presented to evaluate the equaliser performance in such situations. All simulations were run for the $\alpha = 0.5$ case.

Results for $\tau = 0.5T_{symbol}$

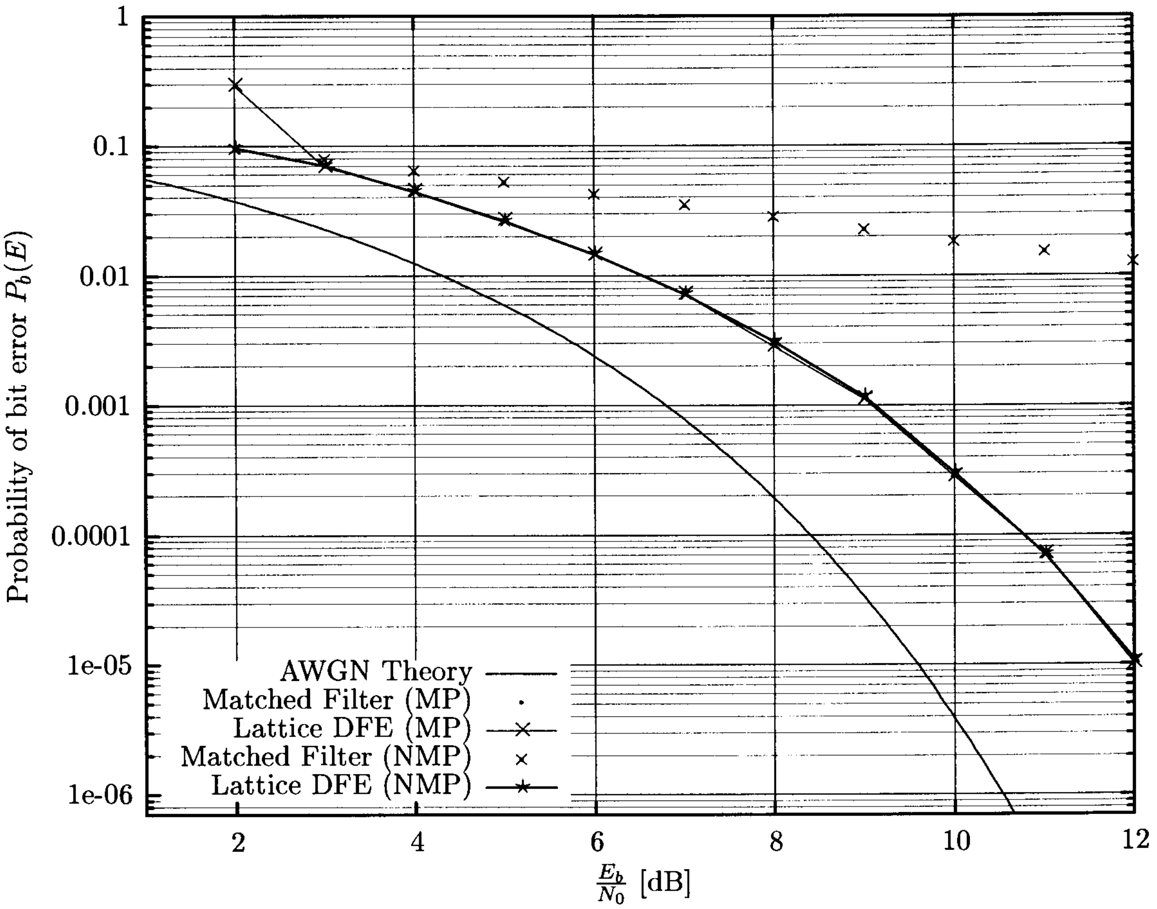

Figure 6.7: Static multipath result for $\alpha = 0.5$, $\tau = 0.5T_{symbol}$.

By comparing Figure 6.7 with Figure 6.4, which is the symbol spaced $\alpha = 0.5$ result, it can be seen that the performance of the DFE has degraded. This is due to the fact that the equaliser is only a symbol spaced equaliser, and will thus not be able to equalise sub-symbol spaced multipath channels as well as it can equalise symbol spaced multipath channels [34]. A fractionally spaced equaliser (FSE) samples the incoming signal at at least the Nyquist rate. The FSE can thus compensate for channel distortion before aliasing effects caused by sampling at the symbol rate occur.

Results for $\tau = 2.0T_{symbol}$

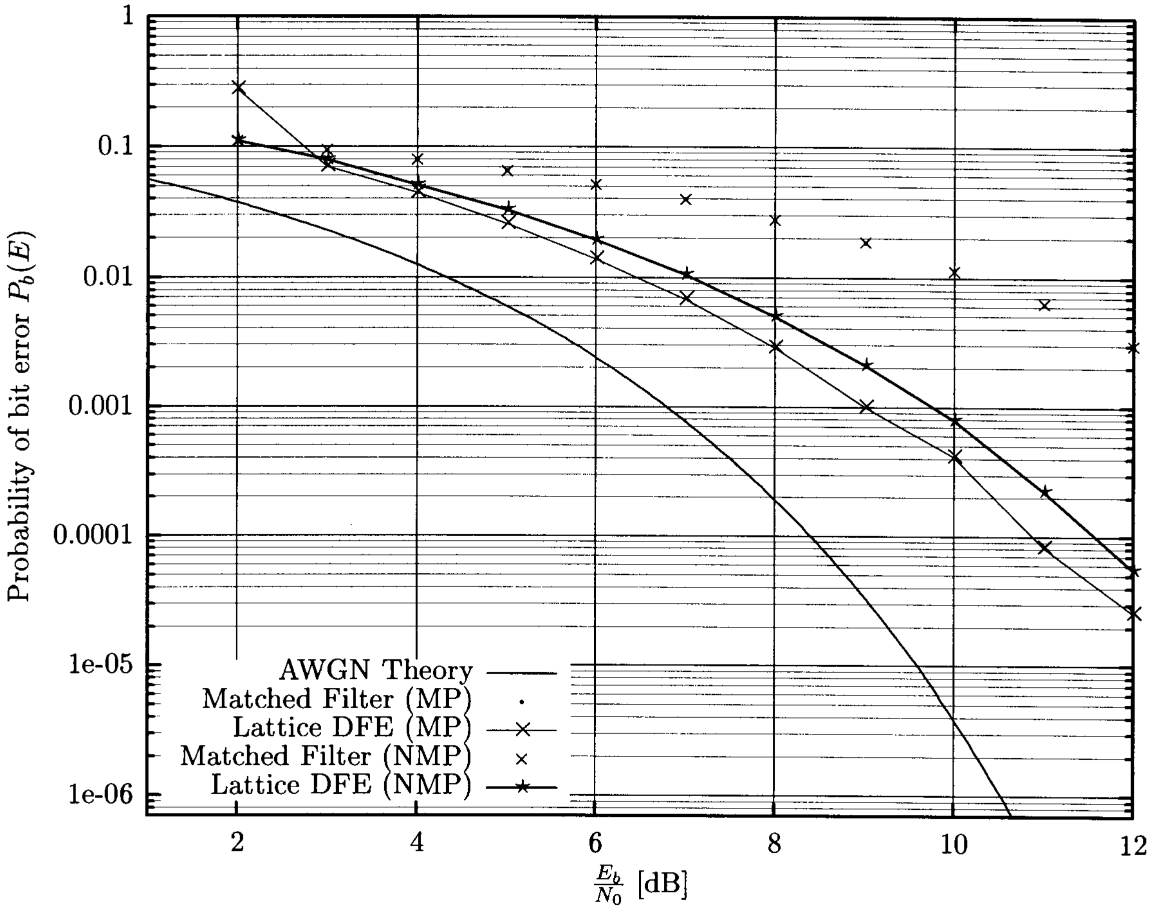

Figure 6.8: Static multipath result for $\alpha = 0.5$, $\tau = 2.0T_{symbol}$.

In Figure 6.8 it can be seen that both the DFE results have degraded slightly as compared to the $\tau = 0.5T_{symbol}$ case in Figure 6.7. This is due to the fact that for the $\tau = 0.5T_{symbol}$ case there is only one spectral notch which coincides with a null in the Q^{2}PSK spectrum, but for the $\tau = 2.0T_{symbol}$ case there are multiple notches within the main lobe of the Q^{2}PSK spectrum. The performance of the DFE for the non-minimum phase case is worse than that of the minimum phase case. This is due to the fact that the first multipath component is corrected for in the 2nd half of the lattice structure, and that the structure has less remaining resources available to eliminate the effect of the multipath, as there are only three feedback stages. It was also shown in Chapter 5 that non-minimum phase systems are not invertible.

Results for $\tau = 3.0T_{symbol}$

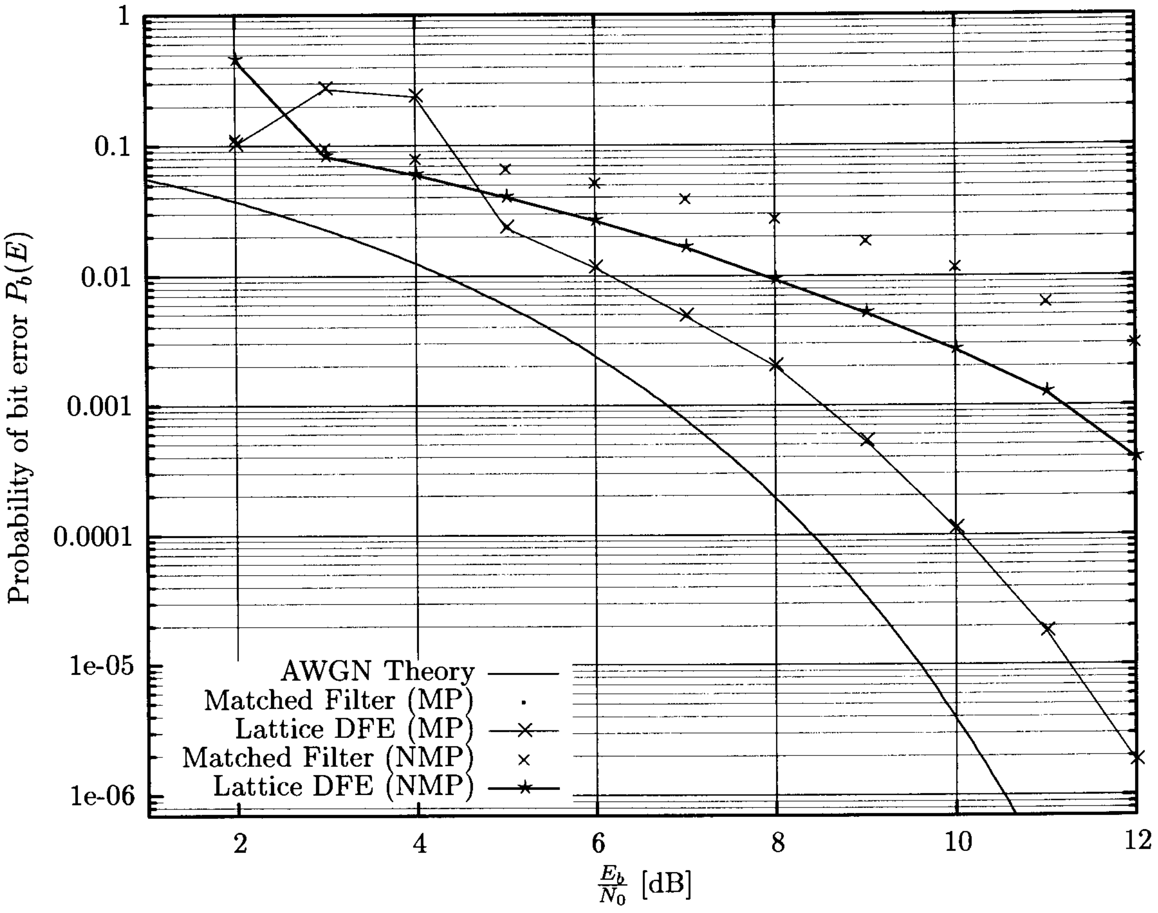

Figure 6.9: Static multipath result for $\alpha = 0.5$, $\tau = 3.0T_{symbol}$.

In Figure 6.9 it can be seen that the DFE non-minimum phase case has once again degraded as compared to the $\tau = 2.0T_{symbol}$ case in Figure 6.8. The cause is the same as that discussed for the $\tau = 2.0T_{symbol}$ case. It is interesting to note that the minimum phase case has better performance, due to equaliser having more resources available to mitigate the effect of the first multipath, as well as the fact that the NMP channels are not invertible.

6.4 V/UHF FADING CHANNEL RESULTS

In this section the equaliser performance in slow fading, frequency selective channels, will be presented. The tests were designed in such a manner as to keep the total probability density function of the envelope, of the received signal, Ricean distributed with a Ricean factor of $K = 10$ dB. This factor for the pdf of the received signal envelope will hence forth be referred to as the effective K of the channel, and will be denoted by K_{eff}. In all the simulations the main path was always a Ricean faded signal, and the multipath a rayleigh faded signal. As the strength of the Rayleigh path is increased, the K factor of the main path has to be increased to maintain $K_{eff} = 10$ dB. Once again the relative strength of the multipath component relative to the main, unity strength, component will be denoted by α.

6.4.1 Simulation Strategy for Fading Channels

Use was made of an automated process to ensure that the simulation was run for long enough at each chosen E_b/N_0 point. This process consists of a number of checks made on certain simulation parameters, which all have to be true before the next simulation point is initiated.

For the fading channel tests, the first test checks that the simulation has processed a minimum number of samples, which is given by :

$$N_{samples\ min} = \frac{75}{f_d} \tag{6.3}$$

This means that as the fading rate is reduced, the simulation will run for more samples. If the correct number of samples have been processed, then checks are made as to wether errors have occurred. If no errors have occurred the simulation continues running. If errors have occurred, then a check is made as to wether the power output of the channel has converged to approximately unity. This value must be within the following bounds :

$$P_{channel\ out} = 1.0 \pm 0.025 \tag{6.4}$$

If all three the above requirements are met, then the convergence of the BEP is checked. To achieve this, use is made of the following equation :

$$\delta_{BEP} = \left| 1 - \frac{BEP_{current}}{BEP_{previous}} \right| \tag{6.5}$$

where this equation is evaluated for the BEP of the equaliser. The value of δ_{BEP} has to be less than 0.007 for five consecutive single error events before the simulation, for the current E_b/N_0, is terminated, and the simulation for the next E_b/N_0 point is initiated.

6.4.2 Fixed Fading Rate Results

The fading rate for the results presented in this section was held constant at $f_d = 10^{-5} f_{sample}$. This implies that the velocity of the mobile user is constant.

Fading Channel Results for $\alpha = 0.0$ and $\alpha = 0.3$

The case where $\alpha = 0$, represents the no multipath case, with the Ricean factor of the main path being $K_{path1} = 10$ dB. This result will be the best possible performance of the equaliser as it has no frequency selective multipaths which it has to mitigate. This result will thus be used as the benchmark for the other fading tests.

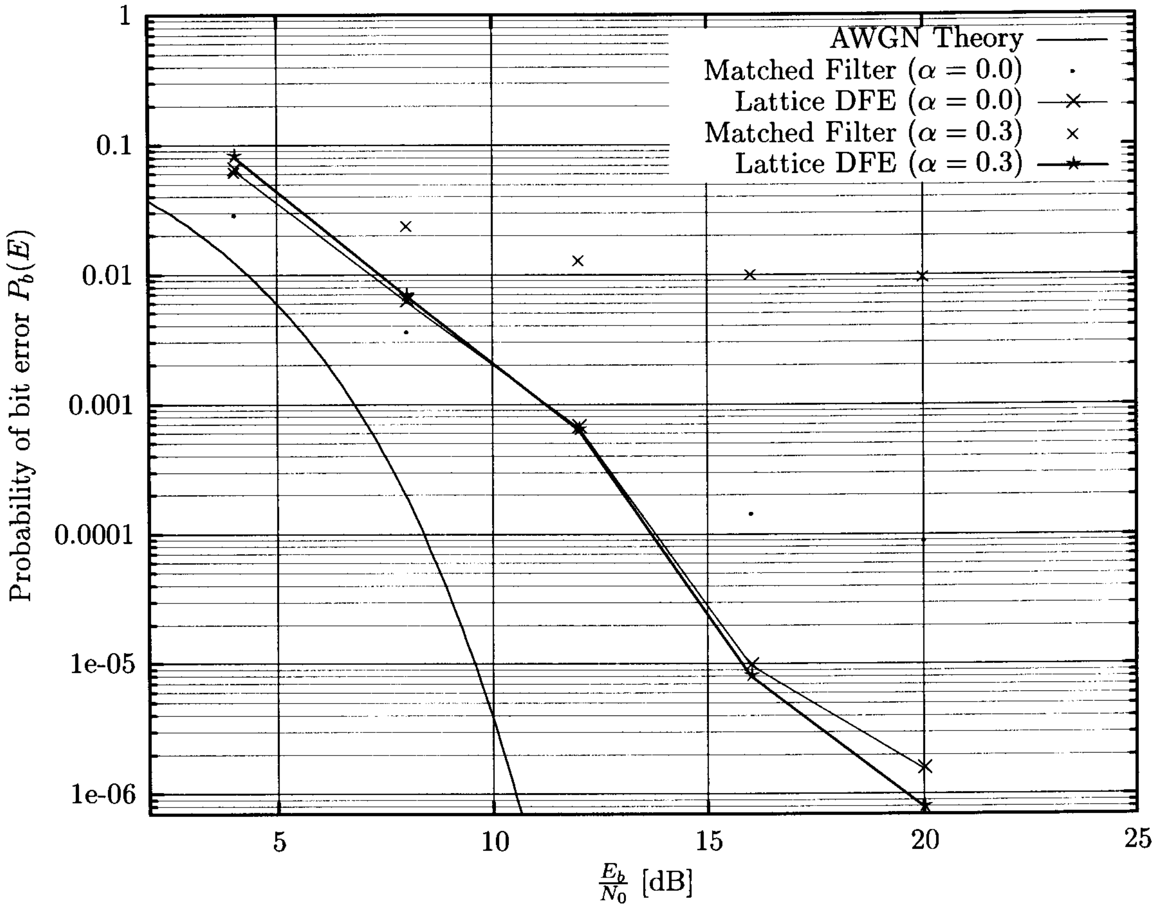

Figure 6.10: Fading multipath result for $\alpha = 0.0$ and $\alpha = 0.3$.

Figure 6.10 shows the results for the $\alpha = 0$ and $\alpha = 0.3$ cases. It is interesting to note the large loss in performance, for the matched filters, due to the added multipath. It can be seen, for the matched filter case, that the irreducible error floor for the $\alpha = 0.3$ case is two orders of magnitude worse than the $\alpha = 0.0$ case. The equaliser can be seen to be very effective in the mitigation of the multipath effects as its performance is very similar for both the minimum phase and non-minimum phase cases. The beginnings of an irreducible error floor are also apparent for the equaliser case.

Fading Channel Results for $\alpha = 0.5$

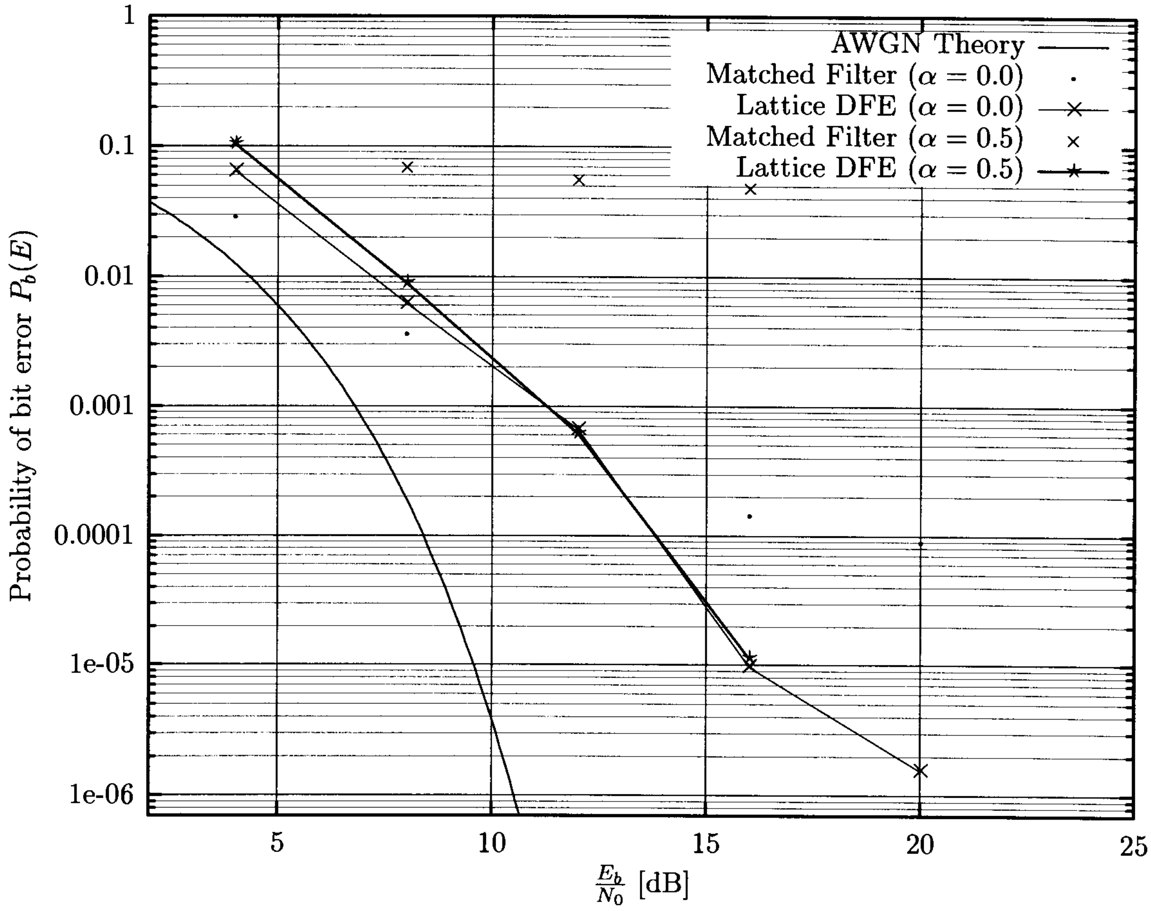

Figure 6.11: Fading multipath result for $\alpha = 0.0$ and $\alpha = 0.5$.

From Figure 6.11 it can be seen that the irreducible error floor for the matched filter case is worse than for the $\alpha = 0.3$ case. The performance of the equaliser can once again be seen to match that of the $\alpha = 0.0$ case.

Fading Channel Results for $\alpha = 0.7$

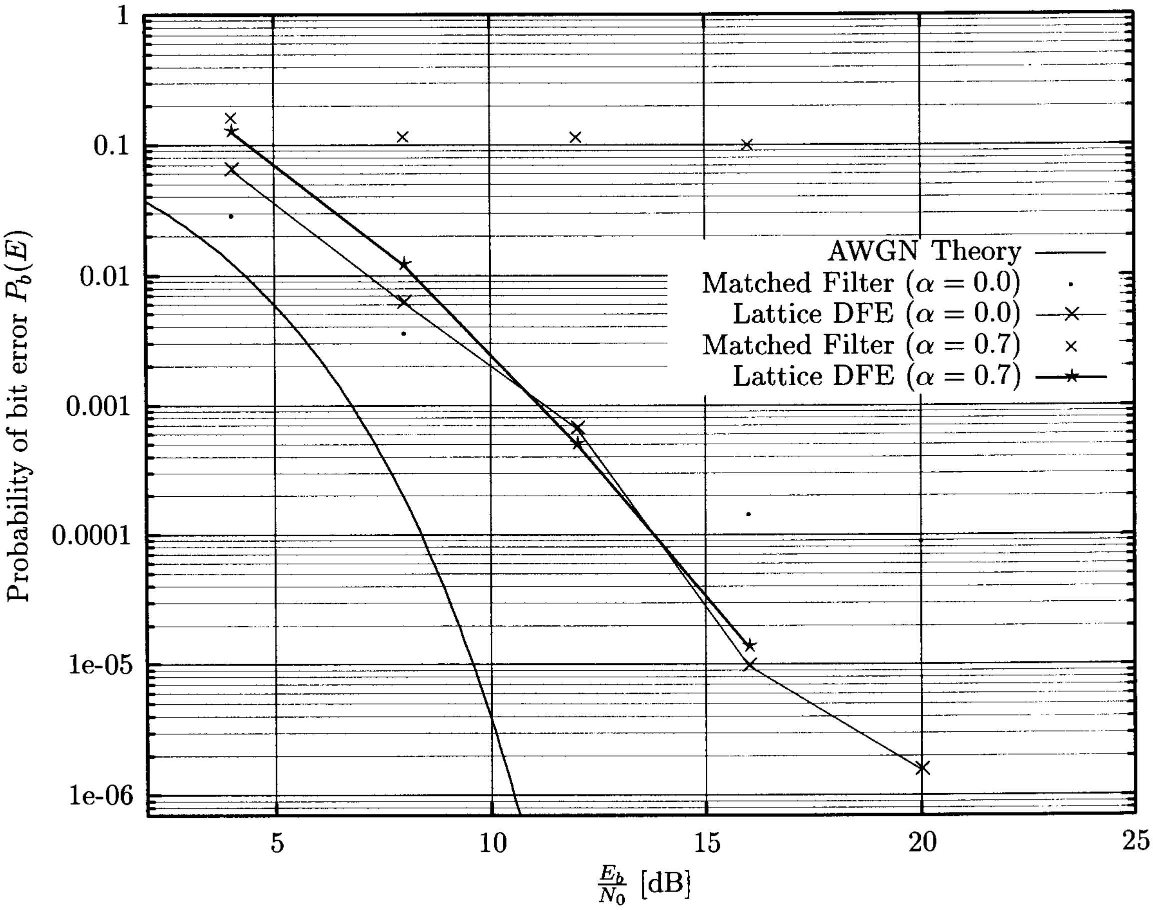

Figure 6.12: Fading multipath result for $\alpha = 0.0$ and $\alpha = 0.7$.

From Figure 6.12 it can be seen that the channel is not usable if use is only made of a matched filter receiver. It can once again be seen that the equaliser performance is close to the Ricean single path case.

Fading Channel Results for $\alpha = 1.0$

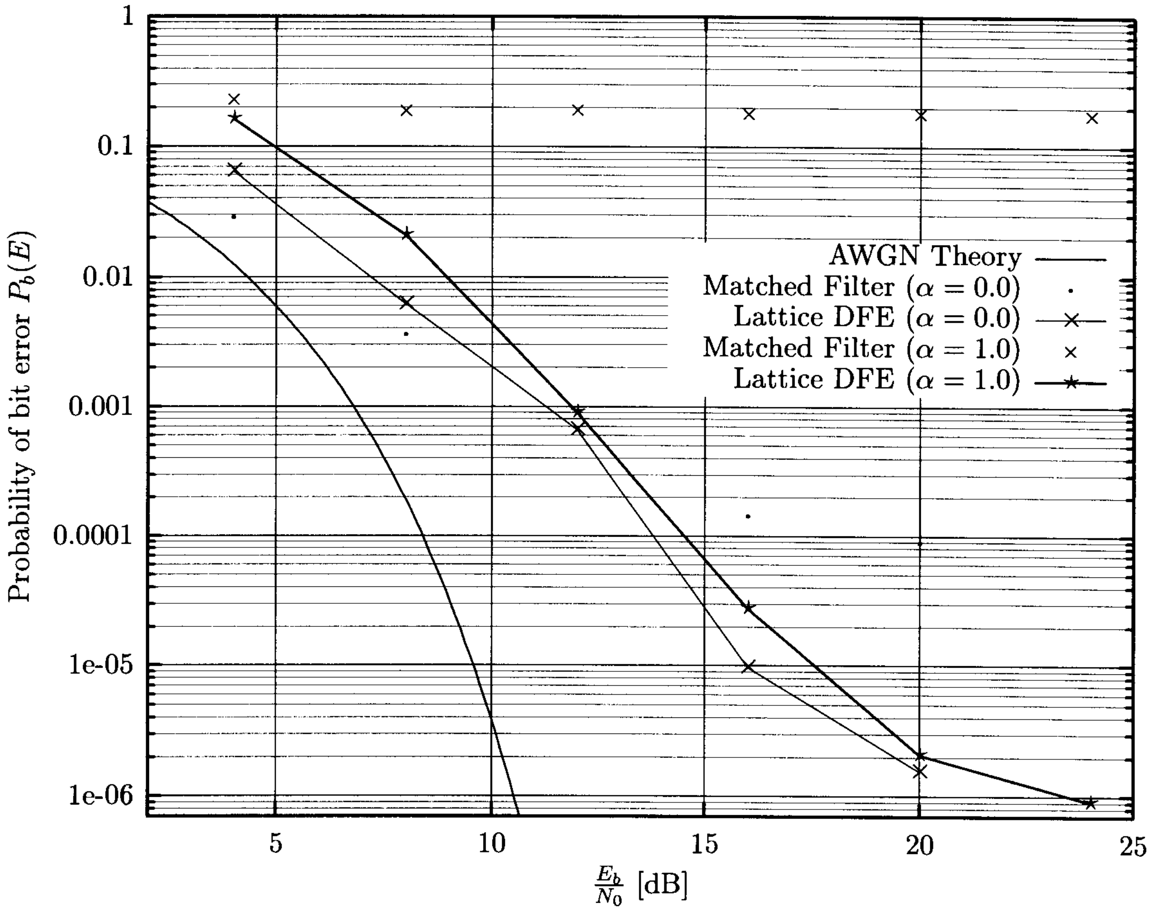

Figure 6.13: Fading multipath result for $\alpha = 0.0$, $\alpha = 1.0$.

From Figure 6.13 it can be seen that the matched filter for the $\alpha = 1.0$ case is useless, unless some form of error correction is added. The equaliser performance is once again relatively close to that of the $\alpha = 0.0$ case, as was expected. This simulation was run longer to evaluate the irreducible error floor, which can be seen for the equaliser to be below the 1.0×10^{-6} bit error probability level. It should be noted that it is an extremely arduous exercise to measure BEP's below this level by means of simulation.

6.4.3 Minimum and Non-minimum Phase Fading Results

In this section the effect of non-minimum phase channels on equaliser performance will be evaluated by means of comparison with the results for the minimum phase channels. This will be done for the cases of $\alpha = 0.7$ and $\alpha = 1.0$

Non-minimum Phase Fading Channel Results for $\alpha = 0.7$

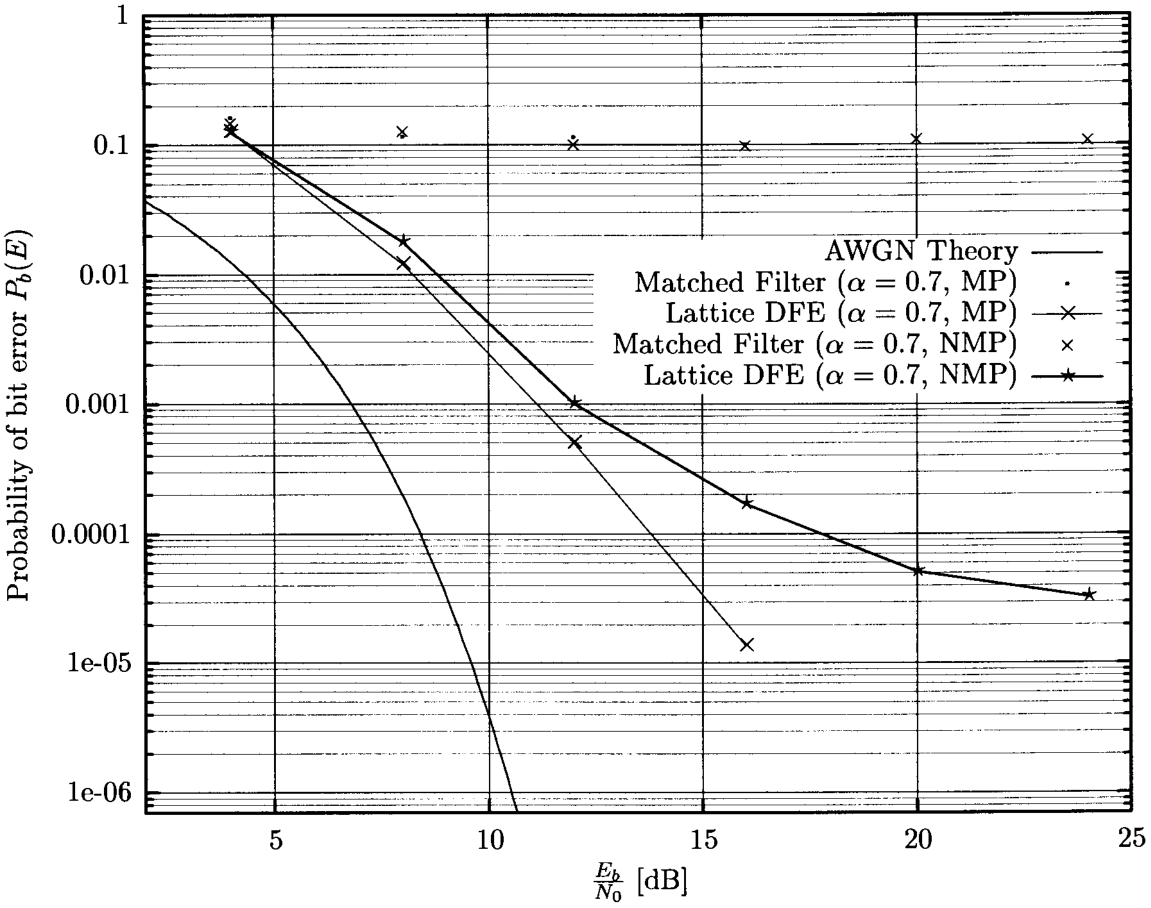

Figure 6.14: Fading non-minimum phase multipath result for $\alpha = 0.7$.

Figure 6.14 shows that for the non-minimum phase fading case the equaliser exhibits an irreducible error floor. The cause of this phenomenon will be discussed after the next result has been presented.

Non-minimum Phase Fading Channel Results for $\alpha = 1.0$

These tests were conducted by forcing the synchronisation algorithm to choose the first correlation peak for the minimum phase case, and the second peak for the non-minimum phase case.

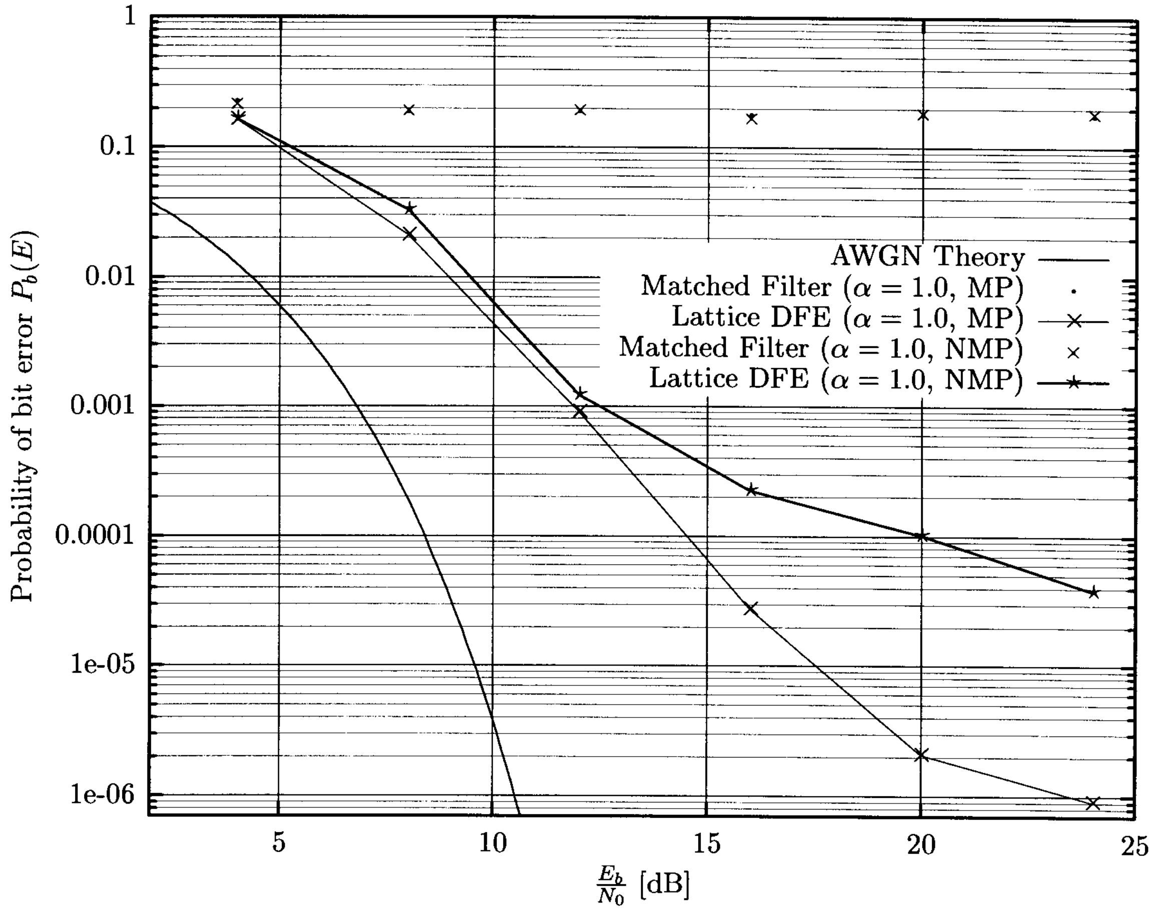

Figure 6.15: Fading non-minimum phase multipath result for $\alpha = 1.0$.

Figure 6.15 shows that both the minimum phase, and non-minimum phase (NMP) channels cause the equaliser to exhibit an irreducible error floor. This can be attributed to the fact that multipath fades cause notches in the spectrum of the received signal. The equaliser tries to compensate for these notches by increasing its gain at the frequency of the notches. This has the effect of increasing the noise level in the received signal, which is responsible for the irreducible error floor. It should also be kept in mind that an inverse filter does not exist for NMP channels, which would explain why under NMP conditions the equaliser always exhibits worse performance as compared to a similar minimum phase case.

6.4.4 Variable Fading Rate Results

In this section the equaliser performance will be evaluated on channels which have higher fading rates than the results presented in the previous sections. The fading rates for which the equaliser will be evaluated are $f_d = 10^{-4} f_{sample}$ and $f_d = 10^{-3} f_{sample}$, which are one and two orders of magnitude faster than the $f_d = 10^{-5} f_{sample}$ case. The multipath setup will be held constant at $\alpha = 0.7$, and $K_{eff} = 10$ dB, for a minimum phase channel. The spacing between the multipaths is once again T_{symbol}.

Fading Rate Results for $f_d = 10^{-4} f_{sample}$

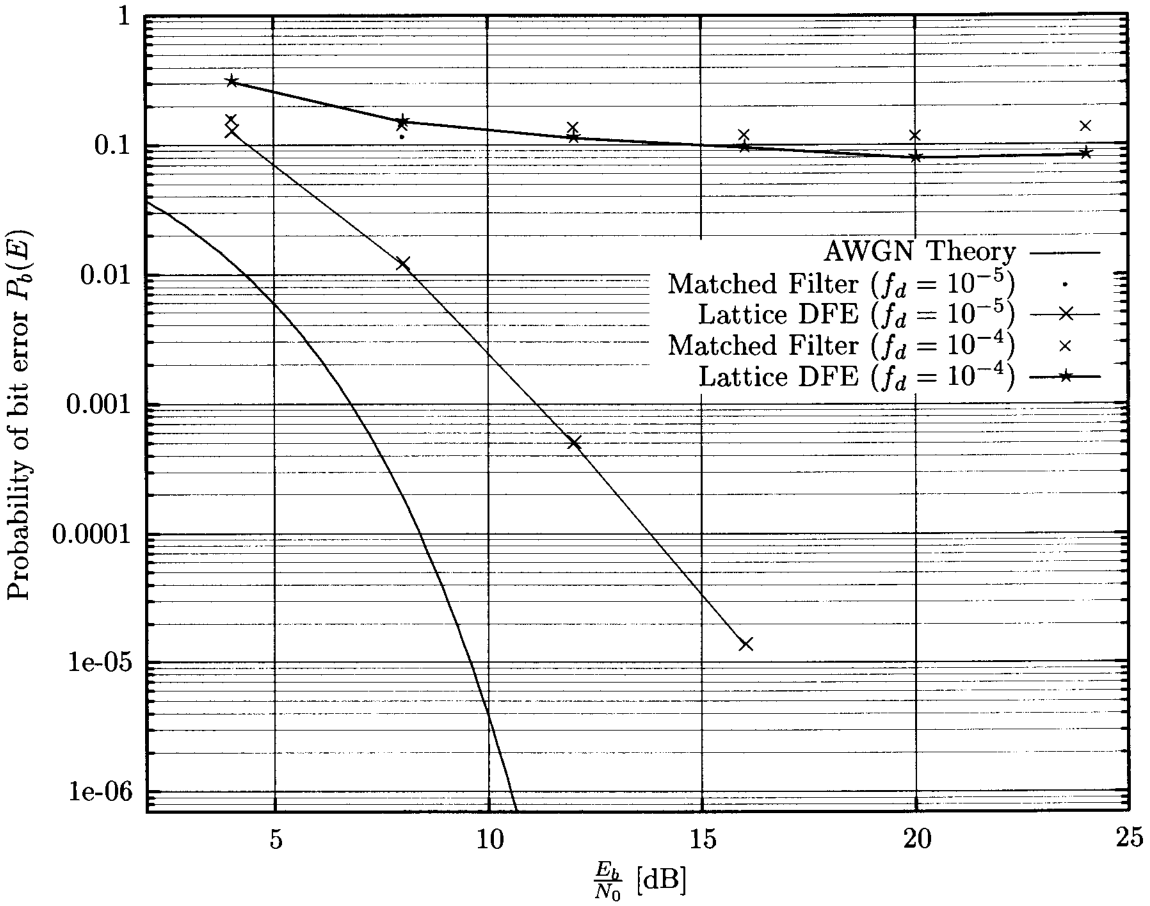

Figure 6.16: Fading rate result for $f_d = 10^{-4} f_{sample}$.

From Figure 6.16 it can be seen that the equaliser fails to give any usable improvement over the matched filter case. The reason for this will be discussed in more detail after the next result has been presented.

Fading Rate Results for $f_d = 10^{-3} f_{sample}$

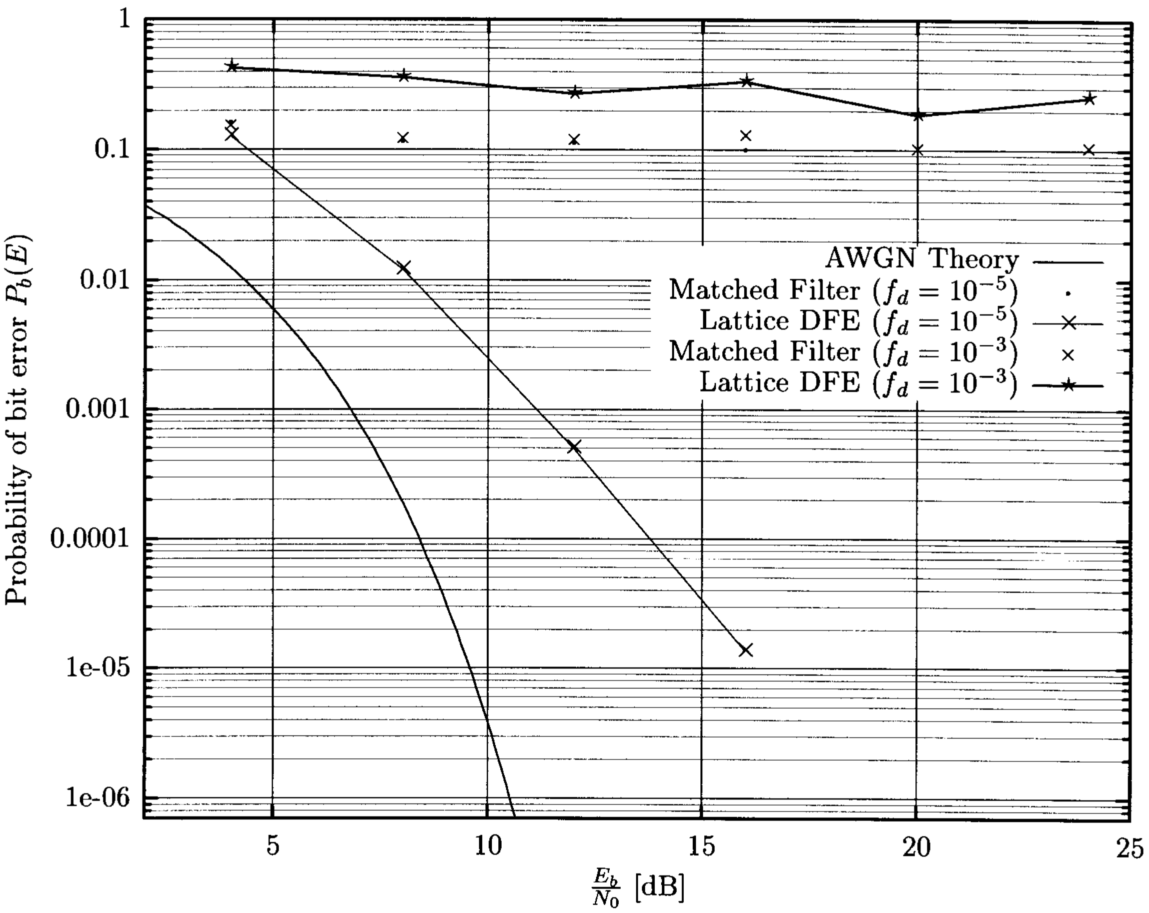

Figure 6.17: Fading rate result for $f_d = 10^{-3} f_{sample}$.

The result in Figure 6.17 shows that the equaliser performs worse than the matched filter case. From this result, and the prvious result for the $f_d = 10^{-4} f_{sample}$ case, it can be seen that the equaliser cannot follow fades of these speeds. This can be attributed to the weighting factor used in the lattice algorithm. The exponential weighting factor was set to $\lambda = 0.99$ for all the simulations run. This means that the algorithm averages over a relatively long period of time, which in turn means that if the channel is not approximately stationary during this time frame, the equaliser will form an erroneous inverse model of the channel.

6.5 CONCLUSION

In this chapter various results, in terms of bit error probability were presented to illustrate the performance characteristics of the adaptive equaliser.

It should be noted that the generation of one BEP curve takes approximately 3 days on a 660 MHz Pentium II personal computer. This chapter thus represents many hours of simulation work in total. This is not even to mention the amount of time taken to develop the simulation software.

Firstly results were presented showing the equaliser performance in the presence of AWGN only. These results showed that the equaliser does not induce more than a 0.4 dB loss, as compared to the theoretical case. These results also showed that the simulation is accurately calibrated, as the matched filter BEP result matched the theoretical case very accurately.

Results showing the operation of the adaptive carrier recovery loop for constant frequency offsets were presented to show the effectiveness of this technique.

Secondly a set of results were presented to illustrate the performance of the equaliser in static multipath (non-fading) environments. These results showed that the equaliser gives a large performance gain over the matched filter case. It was also shown that the equaliser performance degrades as the multipath situation becomes more severe.

Thirdly a set of results illustrating the BEP performance of the equaliser over various slow fading (non-stationary) channels were presented. From these results it can be seen that the equaliser once again gives a large performance increase over the matched filter case. The results also showed that if the Ricean factor of the fading channel is kept constant as the multipath component is increased, that the equaliser performance remains close to the Ricean line of sight (LOS) channel which is not frequency selective.

Fourthly a set of results for non-minimum phase fading channels, for relatively severe multipath cases is presented. These results show that the equaliser shows an irreducible bit error probability, due to noise enhancement.

Finally a set of results was presented illustrating the performance of the equaliser for fast fading channels. These final results show that the equaliser cannot track faster varying channels with the same fidelity as the $f_d = 10^{-5} f_{sample}$ case. This is attributed to the setup of the exponential weighting constant, λ in the lattice algorithm.

The results presented in this chapter thus illustrate the performance gain of the equaliser over the matched filter case, as well as the limitations of the equaliser.

CHAPTER 7

CONCLUSION

In this chapter a summary of the research performed during this study is presented. This work has been focussed on achieving the goal of designing an adaptive lattice decision feedback equaliser for the 4-D Q^{2}PSK modulation technique. A discussion of topics for future study is also presented.

7.1 Conclusions

In Chapter 1 it was stated that multipath propagation in a wireless mobile communication environment is probably one of the most hostile channel effects which can be encountered by an engineer designing a communication system. Bandwidth efficiency was also identified as a very important attribute of emerging modulation techniques.

The aim of this dissertation was the development of a multidimensional adaptive lattice equaliser for the spectrally efficient modulation technique referred to as Q^{2}PSK. To achieve this goal an accurate Q^{2}PSK modulation simulation platform was developed and a multidimensional adaptive lattice decision feedback equaliser designed, and integrated into this platform. Along with this an accurate frequency selective fading channel simulator was developed and integrated with the modem simulation platform to allow seamless (non file based) evaluation of the modem in conjunction with the adaptive equaliser. The equaliser was validated, and its performance tested by means of a rigorous set of simulation tests presented in Chapter 6. These simulations were conducted for additive white Gaussian noise, Doppler offsets, static multipath and fading multipath scenarios. The results of these tests show that in most cases the adaptive equaliser considerably increases the performance of the modem over a wide variety of scenarios in which the modem can be applied for data transfer.

7.1.1 Contributions

In this section specific contributions are discussed in more detail. The contributions of this dissertation are listed below :

- This dissertation presents the design and testing of multidimensional adaptive filtering as a means of adaptive channel equalisation for four-dimensional digital communication signals corrupted by ISI. This is the first time that this problem has successfully been addressed by making use of adaptive filters. The specific topology of the lattice filter also seems to be unique, but this is difficult to verify as such filters are used in a wide variety of unrelated feilds, for example geophysics.

- From the simulation results presented in Chapter 6 it can be seen that the adaptive equaliser performs exceptionally well under most multipath conditions. This includes static multipath channels and fading multipath channels. The performance improvement can be difficult to measure in dB's due to the fact that the baseline performance often shows an irreducible error floor. From the simulation results it can be seen that the equaliser often breaks this irreducible error floor, or translates it to a much lower BEP level.

- The BEP results presented in Chapter 6 thus show that the lattice equaliser has been correclty designed. Better performance could have been achieved if use was made of a longer equaliser, but this would have meant drastically increased simulation time. This was not deemed necessary as the relatively short length of the equaliser was enough to demonstrate the performance improvement brought about by equalisation, as well as the factors which limit the performance of the equaliser.

- A rigorous derivation of the effect of multipath transmission on the Q^2PSK signal was presented in Chapter 2. The result of this derivation gave insight into the correct design of the adaptive equaliser. An in depth discussion of dimensionality presented in Chapter 2, as well as the correct calculation of the spectral efficiency of Q^2PSK led to the correction of certain misleading graphs and statements presented by the originators of Q^2PSK, namely Saha and Birdsall [10, 11].

- An accurate (to within 0.1 dB of theoretical performance) Q^2PSK simulation platform was developed, to which the adaptive lattice equaliser was interfaced. A recursive least squares (RLS) based adaptive digital phase-locked loop (DPLL) was implemented and tested for carrier recovery in the Q^2PSK simulation platform.

- An accurate, software, fading multipath channel simulator was developed to evaluate the performance of the equaliser in mobile wireless channel environments. The channel simulator is capable of simulating Ricean, as well as Rayleigh fading multipaths. This included the development of a new set of Doppler filter design equations, which were shown, in Chapter 3, to be accurate over a range of bandwidths spanning five orders of magnitude. During the testing of the Doppler filters, the fast startup algorithm presented by Kay [89], was shown to become unstable for very small bandwidths relative to the sampling frequency of the system.

- A closed form expression was derived for the sum of the squared impulse response of an infinite impulse response (IIR) filter. This was necessary to achieve the correct normalization of the doppler filters. Up until now this has usually been achieved emperically, by means of simulation.

7.2 Topics for Future Study

The purpose of this dissertation was the development of a lattice equaliser for Q^2PSK. This means that several topics for future study were identified during the design process. These topics will be divided into subsections and discussed separately.

7.2.1 Modulation

The Q^2PSK modem used for simulation purposes made use of the most elementary waveforms. The optimization of these waveforms has not been covered fully in the open literature, so this is an open area for research. This research should not only be restricted to the 4-D case, but should be generalized to the N-dimensional case. Rotationally invariant and differential multidimensional modulation schemes also represent a wide area of research. The use of wavelets to create bandwidth efficient modulation techniques [127] is also a relatively young area of research.

7.2.2 Equalisation

This study focussed on the development of a symbol spaced decision feedback lattice equaliser. Other forms of adaptive algorithms can be used for the equalisation of multidimensional signals. The use of so-called "blind" equalisation algorithms [128], which do not require the use of a training sequence can also be researched. The other area open for investigation is that of fractionally spaced (less than symbol spaced) equaliser topologies. Theoretical performance bounds for adaptive equalisers is also a daunting field which is open for investigation.

7.2.3 Channel Modelling

The instability of the fast startup method for IIR filters presented by Kay [89] presents itself as a well contained generally applicable area of research. Various other areas pertaining to the realisation of efficient doppler filters for channel simulators can also be investigated. A more accurate phase splitter design can be developed for the generation of complex samples in the channel simulator.

7.2.4 Simulation Techniques

Recently several new simulation techniques, collectively referred to as "importance sampling " [129], which can be used to speed up Monte Carlo type simulations have been published. The application of these techniques to a modem structure containing an adaptive equaliser will be a very challenging field of research.

7.2.5 Performance Calculation

The performance of multidimensional signalling techniques in various types of fading channels also represents a very challenging field of research. This is especially true if analytical solutions are sought for the various fading scenarios [46].

7.3 Concluding Remarks

This chapter concludes this dissertation, and has presented a summary of what has been achieved in this dissertation. A discussion of future topics for research which have been identified has also been included.

APPENDIX A

BEP Expressions

This appendix contains a list of bit error probability (BEP) expressions used in the generation of the channel capacity graph. Gray coding was assumed in the mapping to the signal space, which implies that the following expression can be used to convert the symbol error probability to a bit error probability:

$$P_{bit} = \frac{P_{symbol}}{\log_2(M)} \tag{A.1}$$

where M is number of signal constellation points, P_{bit} is the bit error probability and P_{symbol} is the symbol error probability.

Modulation Format	P_{symbol}
BPSK	$P_e = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$
QPSK/MSK	$P_e = \text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) - \frac{1}{4}\text{erfc}^2\left(\sqrt{\frac{E_b}{N_0}}\right)$
M-PSK	$P_e \simeq \text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\sin\left(\frac{\pi}{2M}\right)\right) \quad M \geq 4$
M-QAM	$P_e \simeq 2\left(1 - \frac{1}{\sqrt{M}}\right)\text{erfc}\left(\sqrt{\frac{3E_{avg}}{2(M-1)N_0}}\right)$
M-FSK	$P_e \leq \frac{M-1}{2}\text{erfc}\left(\sqrt{\frac{E_s}{2N_0}}\right)$

APPENDIX B

Code Listing for Multi-dimensional Lattice

This appendix contains the C++ code listing for the multi-dimensional adaptive lattice filter. The compactness and readability of the lattice equations was brought about by the use of the matrix class.

```
              /*************************
               *************************
               **                     **
               **  LATTICE  TEMPLATE  **
               **                     **
               *************************
               *************************/



/***************************************
 *                                     *
 *  Author : J.E. Cilliers             *
 *  Year   : 1996                      *
 *                                     *
 ***************************************
 *                                     *
 *  Update record :                    *
 *                                     *
 *     Started     : 23-10-1996        *
 *                                     *
 *     Last update : 10-01-1998        *
 *                                     *
 ***************************************/


/*
      Approach:
      =========

         - Template Lattice adaptive filter class.
         - Optimize for speed.
         - Can be inherited into a safer class.
         - Test program for all features.
         - Do adaptive algorithms first, then inherit into equaliser class.


      Mathematical / Structural notes :
      =================================

         - Lattice class needs two class types for the template
            one for the scalar parameters, and one for the actual
            data type used in the lattice.


      File names:
      ===========

         - latt_t - Lattice adaptive filter template.
*/


#include <conio.h>
#include "c:\usr\jacques\masters\c\q2_new\mat_lib\mat2d_t.h"
#include "c:\usr\jacques\masters\c\q2_new\math_lib\math_lib.cpp"
#include "c:\usr\jacques\masters\c\q2_new\lattice\latt_t.h"


/****************************
 *                          *
 *  Double Channel Lattice  *
 *                          *
 ***************************/


template<class Scalar>
void DC_Lattice<Scalar>::alloc(void)
{
   // Input/output order and length must be initialised !
   a = alloc_scalar(length,"a");
   a_p = alloc_scalar(length,"a_p");

   // Delay line for actual input data.
   FF_delay_line = alloc_matrix(length,s_input_order,1,"FF_delay_line");
   Desired.set_dim(output_order,1);
   Predict = alloc_matrix(length,output_order,1,"Predict");
   E = alloc_matrix(length,output_order,1,"E");

   // Single Channel Variables.
   F = alloc_matrix(s_length,s_input_order,1,"F");
   B = alloc_matrix(s_length,s_input_order,1,"B");
   B_p = alloc_matrix(s_length,s_input_order,1,"B_p");
   K = alloc_matrix(s_length,s_input_order,s_input_order,"K");
   K_p = alloc_matrix(s_length,s_input_order,s_input_order,"K_p");
   Kx = alloc_matrix(s_length,s_input_order,output_order,"Kx");
   Kx_p = alloc_matrix(s_length,s_input_order,output_order,"Kx_p");
   Rf = alloc_matrix(s_length,s_input_order,s_input_order,"Rf");
   Rf_p = alloc_matrix(s_length,s_input_order,s_input_order,"Rf_p");
   Rb = alloc_matrix(s_length,s_input_order,s_input_order,"Rb");
   Rb_p = alloc_matrix(s_length,s_input_order,s_input_order,"Rb_p");
   Rb_pp = alloc_matrix(s_length,s_input_order,s_input_order,"Rb_pp");

   // Transitional Stage Variables.
   TF_p.set_dim(s_input_order,1);
   TE_p1.set_dim(output_order,1);
   TE_p2.set_dim(output_order,1);
   TKb.set_dim(output_order,s_input_order);
   TKb_p.set_dim(output_order,s_input_order);
   TB.set_dim(output_order,1);

   // Double Channel Variables.
   DF = alloc_matrix(d_length,d_input_order,1,"DF");
   DB = alloc_matrix(d_length,d_input_order,1,"DB");
   DB_p = alloc_matrix(d_length,d_input_order,1,"DB_p");
   DK = alloc_matrix(d_length,d_input_order,d_input_order,"DK");
   DK_p = alloc_matrix(d_length,d_input_order,d_input_order,"DK_p");
   DKx = alloc_matrix(d_length,d_input_order,output_order,"DKx");
   DKx_p = alloc_matrix(d_length,d_input_order,output_order,"DKx_p");
   DRf = alloc_matrix(d_length,d_input_order,d_input_order,"DRf");
   DRf_p = alloc_matrix(d_length,d_input_order,d_input_order,"DRf_p");
   DRb = alloc_matrix(d_length,d_input_order,d_input_order,"DRb");
   DRb_p = alloc_matrix(d_length,d_input_order,d_input_order,"DRb_p");
   DRb_pp = alloc_matrix(d_length,d_input_order,d_input_order,"DRb_pp");

}  // Check 1


template<class Scalar>
```

```
void DC_Lattice<Scalar>::de_alloc(void)
{
   // Total length.
   delete[] a;
   delete[] a_p;
   delete[] Predict;
   delete[] E;

   // Single Channel.
   delete[] F;
   delete[] B;
   delete[] B_p;
   delete[] K;
   delete[] K_p;
   delete[] Kx;
   delete[] Kx_p;
   delete[] Rf;
   delete[] Rf_p;
   delete[] Rb;
   delete[] Rb_p;
   delete[] Rb_pp;

   // Double Channel.
   delete[] DF;
   delete[] DB;
   delete[] DB_p;
   delete[] DK;
   delete[] DK_p;
   delete[] DKx;
   delete[] DKx_p;
   delete[] DRf;
   delete[] DRf_p;
   delete[] DRb;
   delete[] DRb_p;
   delete[] DRb_pp;
}  // Check 1


template<class Scalar>
void DC_Lattice<Scalar>::set_const(void)
{
  w = .99;
  delta = 1;
}  // Check 1


template<class Scalar>
void DC_Lattice<Scalar>::init(void)     // init vars.
```

```
{
  unsigned int     x;

   set_const();
   for(x=0; x<length; x++)
   {
      FF_delay_line[x] = 0;
      Predict[x] = 0;
      E[x] = 0;
      a[x] = 1;
      a_p[x] = 1;
   }

   // Single channel.
   for(x=0; x<s_length; x++)
   {
      F[x] = 0;
      B[x] = 0;
      B_p[x] = 0;
      K[x] = 0;
      K_p[x] = 0;
      Kx[x] = 0;
      Kx_p[x] = 0;
      Rf[x].I();
      Rf_p[x].I();
      Rb[x].I();
      Rb_p[x].I();
      Rb_pp[x].I();
   }

   // Transitional stage;
   TKb = 0;
   TKb_p = 0;
   TF_p = 0;
   TE_p1 = 0;
   TE_p2 = 0;

   // Double channel.
   for(x=0; x<d_length; x++)
   {
      DF[x] = 0;
      DB[x] = 0;
      DB_p[x] = 0;
      DK[x] = 0;
      DK_p[x] = 0;
      DKx[x] = 0;
      DKx_p[x] = 0;
      DRf[x].I();
      DRf_p[x].I();
      DRb[x].I();
      DRb_p[x].I();
      DRb_pp[x].I();
   }
}  // Check 1


template<class Scalar>
void DC_Lattice<Scalar>::shift(void)
{
  unsigned int            x;

  for(x=length-1; x>=1; x--)
  {
    FF_delay_line[x] = FF_delay_line[x-1];
  }
  FF_delay_line[0] = 0;

  for(x=0; x<length; x++)
  {
      a_p[x] = a[x];
  }

  for(x=0; x<s_length; x++)
  {
      B_p[x] = B[x];
      K_p[x] = K[x];
      Kx_p[x] = Kx[x];
      Rf_p[x] = Rf[x];
      Rb_pp[x] = Rb_p[x];
      Rb_p[x] = Rb[x];
  }

  TKb_p = TKb;         // Transitional Stage.
  TF_p = F[s_length-2];
  TE_p1 = E[s_length-1];
  TE_p2 = E[s_length-2];

  for(x=0; x<d_length; x++)
  {
      DB_p[x] = DB[x];
      DK_p[x] = DK[x];
      DKx_p[x] = DKx[x];
      DRf_p[x] = DRf[x];
      DRb_pp[x] = DRb_p[x];
      DRb_p[x] = DRb[x];
  }
```

```
} // Check 1


template<class Scalar>
DC_Lattice<Scalar>::DC_Lattice
        (int s_len, int d_len, int s_inp, int out)
{
  setup(s_len, d_len, s_inp, out);
} // Check 1


template<class Scalar>
void DC_Lattice<Scalar>::setup
        (int s_len, int d_len, int s_inp, int out)
{
  s_length = s_len;
  d_length = d_len;
  length = s_length + d_length;
  s_input_order = s_inp;
  d_input_order = s_inp + out;
  output_order = out;
  alloc();
  init();
} // Check 1


template<class Scalar>
void DC_Lattice<Scalar>::desired_in(Matrix_2d_s<Scalar> des)
{
   Desired = des;
} // Check 1


template<class Scalar>
Matrix_2d_s<Scalar> DC_Lattice<Scalar>::process(Matrix_2d_s<Scalar> in)
{
   unsigned        m, mt, x, y;

   // Input stage.
   a[0] = 1;
   FF_delay_line[0] = in;
   F[0] = in;
   B[0] = in;
   Rf[0] = Rf_p[0]*w + in*CT(in);
   Rb[0] = Rf[0];

   // Single Channel Stage.
   for(m=1; m<s_length; m++)
```

```
{
  F[m] = F[m-1] - CT(K_p[m])*Inv(Rb_pp[m-1])*B_p[m-1];
  B[m] = B_p[m-1] - K_p[m]*Inv(Rf_p[m-1])*F[m-1];
  K[m] = K_p[m]*w + a_p[m-1]*B_p[m-1]*F[m-1];
  if(m < s_length-1)
  {
    Rf[m] = w*Rf_p[m] + a_p[m]*F[m]*CT(F[m]);
    Rb[m] = w*Rb_p[m] + a[m]*B[m]*CT(B[m]);
  }
  s_temp = CT(B[m-1])*Inv(Rb[m-1])*B[m-1];
  a[m] = a[m-1] + sqr(a[m-1])*(s_temp(1,1));
  Predict[m] = Predict[m-1] + (CT(Kx_p[m])*Inv(Rb_p[m-1])*B[m-1]);
}

// Transitional Stage.
m = s_length-1;
TB = TE_p2 - TKb_p*Inv(Rf_p[m-1])*F[m-1];
TKb = w*TKb_p + a_p[m-1]*TE_p2*TF_p;
for(x=1; x<=s_input_order; x++)
{
  DF[0](x,1) = F[m](x,1);
}

y=1;
for(x=s_input_order+1; x<=d_input_order; x++, y++)
{
  DF[0](x,1) = TE_p1(y,1);
}
for(x=1; x<=s_input_order; x++)
{
  DB[0](x,1) = B[m](x,1);
}
y=1;
for(x=s_input_order+1; x<=d_input_order; x++, y++)
{
  DB[0](x,1) = TB(y,1);
}
mt = s_length-1;
DRf[0] = w*DRf_p[0] + a_p[mt]*DF[0]*CT(DF[0]);
DRb[0] = w*DRb_p[0] + a[mt]*DB[0]*CT(DB[0]);

// Double Channel Stage.
mt = s_length;
for(m=1; m<d_length; m++, mt++)
{
  DF[m] = DF[m-1] - CT(DK_p[m])*Inv(DRb_pp[m-1])*DB_p[m-1];
  DB[m] = DB_p[m-1] - DK_p[m]*Inv(DRf_p[m-1])*DF[m-1];
  DK[m] = DK_p[m]*w + a_p[mt-1]*DB_p[m-1]*DF[m-1];
```

```
      DRf[m] = w*DRf_p[m] + a_p[mt]*DF[m]*CT(DF[m]);
      DRb[m] = w*DRb_p[m] + a[mt]*DB[m]*CT(DB[m]);
      d_temp = CT(DB[m-1])*Inv(DRb[m-1])*DB[m-1];
      a[mt] = a[mt-1] + sqr(a[mt-1])*(d_temp(1,1));
      Predict[mt] = Predict[mt-1] + (CT(DKx_p[m-1])*Inv(DRb_p[m-1])*DB[m-1]);
   }
   return(Predict[length-2]);
} // XXX Check 1


template<class Scalar>
void DC_Lattice<Scalar>::update_taps(void)
{
   unsigned int                    m, mt;

   E[0] = Desired;
   mt = 1;
   for(m=1; m<s_length; m++, mt++)
   {
      E[mt] = E[0] - Predict[mt];
      Kx[m] = w*Kx_p[m] + a[mt-1]*B[m-1]*CT(E[mt-1]);
   }
   for(m=1; m<d_length; m++, mt++)
   {
      E[mt] = E[0] - Predict[mt];
      DKx[m-1] = w*DKx_p[m-1] + a[mt-1]*DB[m-1]*CT(E[mt-1]);
   }
} // XXX Check 1


template<class Scalar>
Scalar DC_Lattice<Scalar>::get_sqr_err(unsigned int pos)
{
   Scalar                sqr_err;
   Matrix_2d_s<Scalar>   M_err(1,1);

   M_err = CT(E[pos])*E[pos];
   sqr_err = M_err(1,1);
   return(sqr_err);
} // XXX Check 1


// END of FILEE
```

BIBLIOGRAPHY

[1] W. P. Robins, *Phase Noise in Signal Sources (Theory and Applications).* Peter Peregrinus Ltd., 1982.

[2] C. E. Shannon, "Communication in the presence of noise," *Proc. IRE*, vol. 37, pp. 10–21, January 1949.

[3] C. E. Shannon, "Probability of error for optimal codes in Gaussian channels," *Bell Sys. Tech. Journal*, vol. 38, pp. 611–656, May 1959.

[4] L. F. Toth, "On the densest packing of spherical caps," *Am. Math. Monthly*, vol. 56, pp. 330–331, 1949.

[5] C. L. Weber, "New solutions to the signal design problem for coherent channels," *IEEE Trans. Information Theory*, vol. IT-12, pp. 161–167, April 1966.

[6] C. A. Stutt, "Information rate in a continuous channel for regular simplex codes," *IRE Trans. Information Theory*, vol. IT-6, pp. 516–522, December 1960.

[7] M. S. Ulstad, "A high-performance digital data-tranmission system," *IEEE Trans. Comm. Tech.*, vol. COM-16, pp. 115–119, February 1968.

[8] A. Gersho and V. B. Lawrence, "Multidimensional signal constellations for voiceband data transmission," *IEEE Journal on Selected Areas in Communications*, vol. SAC-2, pp. 687–702, September 1984.

[9] D. Saha and T. G. Birdsall, "Quadrature-Quadrature phase shift keying: a constant envelope modulation scheme," in *Conf. Inform. Sci. Syst.*, pp. 585–590, March 1986.

[10] D. Saha, *Quadrature-Quadrature Phase Shift Keying.* PhD thesis, University of Michagin, 1986.

[11] D. Saha and T. G. Birdsall, "Quadrature-Quadrature phase shift keying," *IEEE Trans. Comms.*, vol. 37, no. 5, pp. 437–448, 1989.

[12] M. Visintin, E. Biglieri, and V. Castellani, "Four-dimensional signals for bandlimited channels," *IEEE Trans. Comms.*, vol. 42, pp. 403–409, March 1994.

[13] O. M. El-Ghandour and D. Saha, "Differential detection in quadrature-quadrature phase shift keying (Q^2PSK) systems," *IEEE transactions on communications*, vol. 39, pp. 703–712, May 1991.

[14] I. Korn and L. Wei, "Q^2PSK in the satellite mobile channel with ISI and CSI," *IEEE Trans. Vehic. Tech.*, vol. 43, pp. 69–78, February 1994.

[15] D. J. van Wyk, "Four-dimensional Q^2PSK modulation and coding for mobile digital communication," Master's thesis, University of Pretoria, 1996.

[16] V. Acha and R. A. Carrasco, "A new digital implementation of quadrature-quadrature phase shift keying," in *3rd IEE Conference on Telecommunications*, 1991.

[17] L. Wei, "Q^2PSK with new pulse shaping pair in the satellite mobile channel," *Electronic Letters*, vol. 29, pp. 2014–2015, November 1993.

[18] R. A. Khalona, G. E. Atkin, and J. L. LoCicero, "On the performance of a hybrid frequency and phase shift keying modulation technique," *IEEE Trnas. Comms.*, vol. 41, pp. 655–659, May 1993.

[19] J. Lui, J. Kim, S. C. Kwatra, and G. H. Stevens, "Rotative quadrature phase shift keying," *Electronic Letters*, vol. 28, pp. 1095–1097, June 1992.

[20] L. W. Hughes, "A simple upper bound on the error probability for orthogonal signals in white noise," *IEEE Trans. Comms.*, vol. 40, p. 670, April 1992.

[21] I. Korn, "Performance of digital communication schemes with multidimensional signals in Gaussian noise," *IEEE Trans. Comms.*, vol. 44, pp. 1070–1072, September 1996.

[22] I. Korn, "Multiamplitude and multidimensional digital communication system in a Gaussian channel," *IEEE Trans. Comms.*, vol. 45, pp. 159–163, February 1997.

[23] L. Wei, "Commonalities of several modulation techniques," *IEEE Trans. Comms.*, vol. 44, pp. 298–300, March 1996.

[24] V. Acha and R. A. Carrasco, "Trellis coded Q^2PSK signals, part 1 : AWGN and nonlinear satellite channels," *IEE Proc. Commun.*, vol. 141, pp. 151–158, June 1994.

[25] V. Acha and R. A. Carrasco, "Trellis coded Q^2PSK signals, part 2 : Land mobile satellite fadign channels," *IEE Proc. Commun.*, vol. 141, pp. 159–167, June 1994.

[26] D. Saha, "Channel coding with quadrature-quadrature phase shift keying (Q^2PSK) signals," *IEEE Trans. Comms.*, vol. 38, no. 4, pp. 409–417, 1990.

[27] R. D. Gauddenzi and M. Luise, "Synchronization of quadrature-quadrature phase-shift keying signals," *IEEE Trans. Comms.*, vol. 40, pp. 1532–1539, September 1992.

[28] V. Acha and R. A. Carrasco, "Maximum-likelihood synchronisation algorithms for Q^2PSK," *IEE Proceedings*, vol. 139, pp. 597–606, December 1992.

[29] D. J. van Wyk, J. E. Cilliers, and L. P. Linde, "Performance of an adaptive carrier Kalman estimator for Q^2PSK," in *IWTS*, pp. 282–287, 1997.

[30] J. E. Cilliers and L. P. Linde, "Application and peformance of a dual-loop adaptive RLS Kalman estimator for carrier recovery in a Q^2PSK system," in *AFRICON '99*, vol. 1, pp. 153–158, September 1999.

[31] D. J. van Wyk and L. P. Linde, "Multidimensional frame synchronisation for 4D-Q^2PSK," *Electronics Letters*, vol. 31, 1995.

[32] S. Feiz and S. S. Soliman, "Maximum-likelihood receiver for four-dimensional signals corrupted by ISI," *IEEE Transactions on Communications*, vol. 40, pp. 265–277, February 1992.

[33] J. G. Proakis, *Digital Communications.* McGraw-Hill Book Company, 1989.

[34] T. S. Rappaport, *Wireless Communications Principles and Practice.* Prentice Hall, 1996.

[35] K. Feher, *Wireless Digital Communications Modulation and Spread Spectrum Applications.* Prentice Hall, 1995.

[36] H. Stark and J. W. Woods, *Probability, Random Processes, and Estimation Theory for Engineers.* Prentice-Hall, 1986.

[37] W. C. Y. Lee, *Mobile Communications Design Fundamentals.* John Wiley & Sons, 1993.

[38] B. Sklar, "Rayleigh fading channels in mobile digital communication systems Part 1: characterization," *IEEE Comms. Mag.*, 1997.

[39] B. Sklar, "Rayleigh fading channels in mobile digital communication systems Part 2: mitigation," *IEEE Comms. Mag.*, 1997.

[40] C. A. Siller, "Multipath propagation," *IEEE Comms. Mag.*, vol. 22, pp. 6–15, February 1984.

[41] U. Dersch and R. J. Ruegg, "Simulations of the time and frequency selective outdoor mobile radio channel," *IEEE Trans. Vehic. Tech.*, vol. 42, pp. 338–344, August 1993.

[42] A. K. Salkintzis, "Implementation of a digital wide-band mobile channel simulator," *IEEE Trans. Broadcasting*, vol. 45, pp. 122–128, March 1999.

[43] S. Qu and T. Yeap, "A three-dimensional scattering model for fading channels in land mobile environment," *IEEE Trans. Vehic. Tech.*, vol. 48, pp. 765–781, May 1999.

[44] J. J. Olmos, A. Gelonch, F. J. Casadevall, and G. Femenias, "Design and implementation of a wide-band real-time mobile channel emulator," *IEEE Trans. Vehic. Tech.*, vol. 48, pp. 746–764, May 1999.

[45] P. Petrus, J. H. Reed, and T. S. Rappaport, "Effects of directional antennas at the base station on the doppler spectrum," *IEEE Communications Letters*, vol. 1, pp. 40–42, March 1997.

[46] M. K. Simon and M.-S. Alouini, "A unified approach to the performance ananlysis of digital communications over generalized fading channels," *Proceedings of the IEEE*, vol. 86, pp. 1858–1876, September 1998.

[47] H. P. Stern, J. G. Hermes, and S. Darbha, "An adaptive propagation prediction program for land mobile radio systems," *IEEE Trans. Broadcasting*, vol. 43, p. 1, March 1997.

[48] A. P. R. Opperman, "VHF/UHF channel simulator for mobile communications," Master's thesis, University of Pretoria, 1995.

[49] J. S. Swarts, "Aspects of multipath channel characterization," Master's thesis, Rand Afrikaans University, 1998.

[50] R. W. Lucky, "Automatic equalizer for digital communication," *Bell System Technical Journal*, vol. 44, pp. 547–588, April 1965.

[51] R. W. Lucky, "Techniques for adaptive equalization of digital communication systems," *Bell System Technical Journal*, vol. 45, 1966.

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

[53] G. Ungerboeck, "Theory on the speed of convergence in adaptive equalizers for digital communication," *IBM J. Res. Develop.*, 1972.

[54] R. W. Harris, D. M. Chabries, and F. A. Bishop, "A variable step (VS) adpative filter algorithm," *IEEE Trans. ASSP*, vol. ASSP-34, pp. 309–316, April 1986.

[55] R. H. Kwong and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Transactions on Signal Processing*, vol. 40, pp. 1633–1642, July 1992.

[56] R. H. Kwong, "Robust design of the LMS algorithm," *IEEE Transactions on Signal Processing*, vol. 40, pp. 2613–2616, October 1992.

[57] V. J. Mathews and Z. Xie, "A stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Trans. Signal Processing*, vol. 41, pp. 2075–2086, June 1993.

[58] D. Godard, "Channel equalization using a Kalman filter for fast data transmission," *IBM Journal of Research and Development*, pp. 267–273, May 1974.

[59] J. Botto and V. Moustakides, "Stabilizing the fast Kalman algorithms," *IEEE Trans. ASSP*, vol. 37, pp. 1342–1348, September 1989.

[60] D. W. Lin, "On digital implementation of fast Kalman algorithms," *IEEE Trans. ASSP*, vol. ASSP-32, pp. 998–1005, October 1984.

[61] B. Mulgrew and F. N. Cowan, "An adaptive Kalman equalizer: structure and performance," *IEEE TRans. ASSP*, vol. ASSP-35, pp. 1727–1735, December 1987.

[62] J. G. Proakis, "Adaptive equalization for TDMA digital mobile radio," *IEEE Trans. Vehic. Tech.*, vol. 40, pp. 333–341, May 1991.

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

[64] G. Carayannis, D. G. Manolakis, and N. Kalouptsidis, "A fast sequential algorithm for least squares filtering and prediction," *IEEE Trans. ASSP*, vol. ASSP-31, 1983.

[65] J. M. Cioffi and T.Kailath, "Fast recursive-least-squares transversal filters for adaptive filtering," *IEEE Trans. ASSP*, vol. ASSP-32, 1984.

[66] S. T. Alexander, "A derivatio of the complex fast Kalman algorithm," *IEEE Trans. ASSP*, vol. ASSP-32, pp. 1230–1232, December 1984.

[67] M. Morf, D. Lee, J. Nickolls, and A. Vierra, "A classification of algorithms for ARMA models and ladder realizations," in *IEEE Int. Conf. on ASSP*, 1977.

[68] M. Morf, A. Vierra, and D. Lee, "Ladder forms for identification and speech processing," in *Proc. 1977 IEEE Conf. on Decision and Control*, 1977.

[69] M. Morf, "Ladder forms in estimation and system identification," in *11th Annual Asilomar Conf. on Circuits, Sytems and Computers*, 1977.

[70] M. Morf and D. Lee, "Recursive least squares ladder forms for fast parameter tracking," in *Proc. 1978 Conf. on Decision and Control*, 1979.

[71] E. H. Satorius and S. T. Alexander, "Channel equalization using adaptive lattice algorithms," *IEEE Trans. Comms.*, vol. COM-27, pp. 899–905, June 1979.

[72] E. H. Satorius and J. D. Pack, "Application of least squatres lattice algorithms to adaptive equalization," *IEEE Trans. Comms.*, vol. COM-29, pp. 136–142, February 1981.

[73] T. L. Lim and M. S. Mueller, "Rapid equalizer start-up using least-squares algorithms," in *Int. Conf. Commun*, 1980.

[74] D. T. L. Lee, M. Morf, and B. Friedlander, "Recursive least squares laddder estimation algorithms," *IEEE Trans. ASSP*, vol. ASSP-29, pp. 627–641, June 1981.

[75] B. Porat, B. Friedlander, and M. Morf, "Square root covariance ladder algorithms," *IEEE Trans. Automatic Control*, vol. AC-27, pp. 813–829, August 1982.

[76] F. Ling and J. G. Proakis, "Numerical accuracy and stability: two problems of adaptive estimation caused by round-off error," in *Proc. ICASSP '84*, 1984.

[77] F. Hsu, "Square-root kalman filtering for high-speed data received over fading dispersive HF channels," *IEEE Trans. Inform. Theory*, vol. IT-28, 1982.

[78] C. G. Samson and V. U. Reddy, "Fixed point error analysis of the normalised ladder algorithm," *IEEE Trans. ASSP*, vol. ASSP-31, pp. 1177–1191, October 1983.

[79] M. J. Shensa, "Recursive least squares lattice algorithms - a geometrical approach," *IEEE Trans. Automatic Contol*, vol. AC-26, pp. 695–702, June 1981.

[80] J. M. Cioffi, "An unwindowed RLS adaptive lattice algorithm," *IEEE Trans. ASSP*, vol. 36, pp. 365–371, March 1988.

[81] M. D. Miranda and M. Gerken, "A hybrid least squares QR-lattice algorithm using a priori errors," *IEEE Trans. Signal Processing*, vol. 45, pp. 2900–2911, December 1997.

[82] M. S. Mueller, "Least-squares algorithms for adaptive equalizers," *Bell Syst. Tech. Journ.*, vol. 60, 1981.

[83] F. Ling and J. G. Proakis, "A generalized multichannel least squares lattice algorithm based on sequential processing stages," *IEEE Trans. ASSP*, vol. ASSP-32, pp. 381–388, April 1984.

[84] F. Ling and J. G. Proakis, "A generalized least square latice algorithm and its application to decision-feedback equalization," in *Proc. ICASSP*, 1982.

[85] F. Ling and J. G. Proakis, "Adaptive lattice deceision-feedback equalizers - their performance and application to time-variant multipath channels," *IEEE Trans. Comms.*, vol. COM-33, pp. 348–356, April 1985.

[86] H. Lev Ari, "Modular architectures for adaptive multichannel lattice algorithms," *IEEE Trans. ASSP*, vol. ASSP-35, pp. 543–552, April 1987.

[87] S. Karaboyas, N. Kalouptsidis, and C. Caroubalos, "Highly parallel multichannel LS algorithms and application to decision-feedback equalizers," *IEEE Trans. ASSP*, vol. 37, pp. 1380–1396, September 1989.

[88] A. P. Petropulu, L. Nikias, and J. G. Proakis, "Efficient realization structure of the a priori lattice-ladder recursive least squares algorithm," *IEEE Trans. Signal Processing*, vol. 39, pp. 992–996, April 1991.

[89] S. M. Kay, "Efficient generation of coloured noise," *Proceedings of the IEEE*, vol. 69, pp. 480–481, April 1981.

[90] H. Urkowitz, *Signal Theory and Random Processes.* Artech House, Inc., 1983.

[91] D. Slepian, H. J. Landau, and H. O. Pollak, "Prolate speriodal wave functions, fourier analysis, and uncertainty principle (i and ii)," *Bell System Technichal Journal*, vol. 40, pp. 43–84, January 1961.

[92] H. J. Landau and H. O. Pollack, "Prolate spheroidal wave functions, fourier analysis, and uncertainty - iii:the dimension of the space of essentially time- and band-limited signals," *Bell Syst. Tech. Journal*, vol. 41, pp. 1295–1336, July 1962.

[93] A. Papoulis, *Signal Analysis.* McGraw-Hill, 1977.

[94] S. Haykin, *Digital Communications.* John Wiley & Sons, 1988.

[95] S. G. Wilson and H. A. Sleeper, "Four dimensional modulation and coding – an alternative to frequency-reuse," *Technical Report*, 1983.

[96] B. Westra, D. J. van Wyk, J. E. Cilliers, and L. P. Linde, "Performance evaluation of multi-level four-dimensional Q^2PSK in gaussian noise," *IEEE COMSIG '97*, pp. 141–146, 1997.

[97] S. L. Drakul and E. Biglieri, "A class of constant envelope multidimensional modulation schemes for nonlinearly amplified channels," *IEEE Singapore ICCS*, pp. 114–118, 1994.

[98] W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering.* Prentice-Hall, 1973.

[99] W. B. Davenport and W. L. Root, *An Introduction to the Theory of Random Signals and Noise.* Mc Graw-Hill Book Co., 1968.

[100] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering.* John Wiley & Sons, 1965.

[101] E. Kreyszig, *Introductory Functional Analysis with Applications.* John Wiley and Sons, 1989.

[102] Y. Okumura, E. Ohmori, and K. Fukuda, "Field strength and its variability in VHF and UHF land mobile radio service," *Rev. Elec. Commun. Lab*, vol. 16, no. 9 and 10, pp. 825–873, 1968.

[103] M. Hata, "Emperical formulae for propagation loss in land mobbile radio services," *IEEE Trans. Vehic. Tech.*, vol. VT-29, no. 3, pp. 317–325, 1980.

[104] R. H. Clarke, "A statistical theory of mobile radio reception," *Bell Sys. Tech. Journ.*, vol. 47, pp. 957–1000, 1968.

[105] A. W. Lam and F. M. Ozluturk, "Performance bounds for ds/ssma communications with complex signature sequences," *IEEE Transactions on Communications*, vol. 40, pp. 1607–1614, October 1992.

[106] S. O. Rice, "Mathenatical analysis of random noise (part ii)," *Bell Systems Technical Journal*, vol. 23, pp. 310–333, July 1944.

[107] S. O. Rice, "Mathenatical analysis of random noise (part i)," *Bell Systems Technical Journal*, vol. 23, pp. 282–310, July 1944.

[108] M. J. Gans, "A power spectral theory of propagation in the mobile environment," *IEEE Transactions on Vehicular Technology*, vol. VT-21, pp. 27–38, February 1972.

[109] R. A. Goubran, H. M. Hafez, and A. U. H. Sheikh, "Implementation of a real-time mobile channel simulator using a dsp chip," *IEEE Trans. Instrumentation and measurement*, vol. 40, pp. 709–714, August 1991.

[110] E. Casas and C. Leung, "A simple digital fading simulator for mobile radio," *IEEE Trans. Veh. Technol*, vol. 39, pp. 205–212, August 1990.

[111] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing.* Prentice-Hall Inc., 1989.

[112] J. F. Kaiser, "Nonrecusive digital filter design using the I_0-sinh window function," in *Proceedings 1974 IEEE International Symposium on Circuits and Systems*, pp. 20–23, April 1974. San Francisco.

[113] B. Wichmann and D. Hill, "Building a random generator," *BYTE Magazine*, pp. 127–128, March 1987.

[114] G. E. P. Box and M. E. Muller, "A note on the generation of random normal deviates," *Annals of Mathematical Statistics*, vol. 29, pp. 610–611, 1958.

[115] J. I. Smith, "A computer generated multipath fading simulation for mobile radio," *IEEE Transactions on Vehicular Technology*, vol. VT-24, pp. 39–40, August 1975.

[116] F. Swarts, "Markov characterization of fading channels," Master's thesis, Rand Afrikaans University, 1991.

[117] R. Haeb, *Kohaerenter Empfang bei Datenuebertragung uber nichtfrequentselective Schwundkanaele.* PhD thesis, Aachen University of Technology, 1986.

[118] C. K. Pauw, *Differential 16-ary Phase-Amplitude Modulation Schemes for Data Transmission over Fading Channels.* PhD thesis, City University of New York, 1983.

[119] B. D. O. Anderson and J. R. Moore, *Optimal Filtering.* Prentice Hall, 1979.

[120] J. D. Markel and A. H. Gray, *Linear Prediction of Speech.* Springer-Verlag, 1976.

[121] M. H. Hayes, *Statistical Digital Signal Processing and Modelling.* John Wiley & Sons Inc., 1996.

[122] S. M. Kay, *Modern Spectral Estimation, Theory and Application.* Prentice Hall, 1988.

[123] W. D. Rummler, "A new selective fading model: application to propagation data," *Bell sytems Technical Journal*, vol. 59, pp. 1037–1071, May-June 1979.

[124] B. Kim, "Dual-loop DPLL gear-shifting algorithm for fast synchronization," *IEEE Trans. Circuits and Systems.*, pp. 577–586, 1997.

[125] B. Kim, "Digital carrier recovery with adaptive dual loop DPLL for mobile communication applications," in *ICASSP*, 1993.

[126] J. E. Cilliers, "E_b/N_0 and SNR," Tech. Rep. DEF 2001/152 Rpt A, Defencetek, 1997.

[127] J. N. Livingston and C. Tung, "Bandwidth efficient PAM signalling using wavelets," *IEEE Trans. Comms.*, vol. 44, pp. 1629–1631, December 1996.

[128] M. Ghosh, "Blind decision feedback equalization for terrestrial television receivers," *Proc. IEEE*, vol. 86, pp. 2070–2081, October 1998.

[129] D. Lu and K. Yao, "Improved importance sampling technique for efficient simulation of digital communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 67–75, January 1988.