Adaptive equalisation for fading digital communication channels

Bradley, Martin James

Adaptive Equalisation for Fading Digital Communication Channels

Martin James Bradley

School of Engineering
University of Durham

September 1996

A thesis submitted for the degree of
Doctor of Philosophy (Ph.D.) of the University of Durham.

Martin James Bradley
Adaptive Equalisation for Fading Digital Communication Channels
Ph.D. 1996.

Abstract

This thesis considers the design of new adaptive equalisers for fading digital communication channels. The rôle of equalisation is discussed in the context of the functions of a digital radio communication system and both conventional and more recent novel equaliser designs are described.

The application of recurrent neural networks to the problem of equalisation is developed from a theoretical study of a single node structure to the design of multinode structures. These neural networks are shown to cancel intersymbol interference in a manner mimicking conventional techniques and simulations demonstrate their sensitivity to symbol estimation errors. In addition the error mechanisms of conventional maximum likelihood equalisers operating on rapidly time-varying channels are investigated and highlight the problems of channel estimation using delayed and often incorrect symbol estimates. The relative sensitivity of Bayesian equalisation techniques to errors in the channel estimate is studied and demonstrates that the structure's equalisation capability is also susceptible to such errors.

Applications of multiple channel estimator methods are developed, leading to reduced complexity structures which trade performance for a smaller computational load. These novel structures are shown to provide an improvement over the conventional techniques, especially for rapidly time-varying channels, by reducing the time delay in the channel estimation process. Finally, the use of confidence measures of the equaliser's symbol estimates in order to improve channel estimation is studied and isolates the critical areas in the development of the technique — the production of reliable confidence measures by the equalisers and the statistics of symbol estimation error bursts.

Declaration

I hereby declare that this thesis is a record of work undertaken by myself, that it has not been the subject of any previous application for a degree, and that all sources of information have been duly acknowledged.

Acknowledgements

The encouragement and incessant good humour of my supervisor, Professor Phil Mars has been greatly appreciated. The technical provisions have been excellent and so too his guidance — save one occasion concerning a crunching tackle on a five-a-side pitch! I have valued the opportunity to return to Durham to undertake a Ph.D. Thank you.

The financial support of the EPSRC and GEC-Marconi has been invaluable. John Milsom, Colin Currie, Chris Dearlove and Clive Tang of the GEC-Marconi Research Centre, to mention but some names, have provided technical advice, constructive criticisms and invaluable direction for which I am sincerely grateful. Many thanks to Ian Alston — his help and tips on SPW have saved, no doubt, many a frustrating hour. David Lee and Barry Cornelius of the University Computing Service could not have provided more prompt support and help on my numerous and often ill-defined computing implementation questions.

Steve Robinson has been a good character to have around; his jokes, humorous gibes and all-round zest has been welcomed. Thanks to all those many postgrads and staff who have made the last three years enjoyable.

Finally, I would like to thank my parents for providing support in more ways than one.

Contents

List of Figures

List of Abbreviations

ADPCM	Adaptive differential pulse code modulation
AGC	Automatic gain control
ASIC	Application specific integrated circuit
AWGN	Additive white Gaussian noise
BCH	Bose-Chaudhuri-Hocquenghem (codes)
BDFE	Bayesian decision feedback equaliser
CAZAC	Constant amplitude zero autocorrelation
CCIR	Comité consultatif international des radiocommunications
CDMA	Code division multiple access
CELP	Code excited linear prediction
C/I	Carrier-to-interference ratio
CIR	Channel impulse response
CPFSK	Constant phase frequency shift keying
CPM	Constant phase modulation
CRC	Cyclic redundancy check
D-AMPS	Digital American mobile 'phone system
DCS	Digital cellular system
DDD-MLSE	Dual decision delay maximum likelihood sequence estimator
DDFSE	Delayed decision-feedback sequence estimator
DDFSE-MCE	Delayed decision-feedback sequence estimator with multiple channel estimators
DECT	Digital European cordless telecommunications
DEQPSK	Differentially encoded quadrature phase shift keying
DFE	Decision feedback equaliser
DFT	Discrete Fourier transform
DM	Delta modulation
DPCM	Differential pulse code modulation
DQPSK	Differential quadrature phase shift keying
DSP	Digital signal processor
ETSI	European telecommunications standards institute

FDD	Frequency division duplexing
FDMA	Frequency division multiple access
FEC	Forward error correction
FIR	Finite impulse response
FSE	Fractionally spaced equaliser
FSK	Frequency shift keying
HF	High frequency
GMSK	Gaussian minimum shift keying
GSM	Global system for mobile communications
I	In-phase
IF	Intermediate frequency
IIR	Infinite impulse response
ISI	Intersymbol interference
ITU	International telegraph union
LCR	Level crossing rate
LMS	Least mean square
LOS	Line-of-sight
LPC	Linear predictive coder
LTE	Linear transversal equaliser
MAP	Maximum *a posteriori*
MCE	Multiple channel estimator
MLSE	Maximum likelihood sequence estimator
MLP	Multilayer perceptron
MMSE	Minimum mean square error
MPE	Multi-pulse excitation
MSE	Mean square error
MSK	Minimum shift keying
MTE	Mean tap error
OQPSK	Offset quadrature phase shift keying
PAM	Pulse amplitude modulation
PCM	Pulse code modulation
PCN	Personal communication network
PCS	Personal communication system
PDC	Personal digital cellular
PLL	Phase locked loop
PMR	Private mobile radio
PRBS	Pseudo-random binary sequence
PSK	Phase shift keying
Q	Quadrature-phase

QAM	Quadrature amplitude modulation
QPSK	Quadrature phase shift keying
QS	Quasi-synchronous
RBF	Radial basis function
RF	Radio frequency
RLS	Recursive least square
R-MCE	Reduced multiple channel estimator
RNN	Recurrent neural network
RPE	Regular pulse excitation
RRC	Root-raised cosine
RSSE	Reduced state sequence estimator
RTRL	Real-time recurrent learning algorithm
SBC	Sub-band coding
SER	Symbol error rate
SNR	Signal-to-noise ratio
TCM	Trellis coded modulation
TDD	Time division duplexing
TDMA	Time division multiple access
TETRA	Trans-European trunked radio
UMTS	Universal mobile telecommunications system
VSELP	Vector sum excited linear prediction
W-MLSE	Windowed maximum likelihood sequence estimator

Chapter 1

Introduction

The growth in the demand for radio communications is increasing as new or improved services become possible. Digital mobile telephony and, more recently, mobile computing has led to the requirement of high quality and ideally large bandwidths to be available to the mobile user. However, providing such data transfers reliably over radio links becomes increasingly difficult due to the characteristics of radio channels. The reasons for the difficulties are that the radio channels have nonconstant frequency responses in amplitude and group delay and that they are time-varying. This frequency selectivity, often arising from the presence of multiple paths within the radio channel, causes a distortion of the transmitted signals and thus accurate determination of the transmitted data becomes more difficult for the receiver. The problem is compounded by the fact that the mechanisms causing the distortion process are often unknown and vary with time in a random manner. The problem is not limited to land mobile radio but also for HF and tropospheric scatter radio links where there is motion in the propagating media of the radio channel.

Transmitting data reliably is of fundamental importance to mobile computing applications whereas some forms of digital mobile telephony can often tolerate some degree of error. The data overheads associated with coding the transmitted data so that it can either be verified and / or corrected in the receiver reduces the available bandwidth of the channel. Thus methods of combating the channel-induced distortion whilst keeping the data overhead as low as possible are very attractive. Equalisation is the term referring to signal processing structures in the radio receiver which are designed to nullify the channel's distorting effects prior to the detection of the received signal. Such structures need to be able to adapt to differing distortion processes as changes in the radio channel occur and, the ability of the adaptation algorithm to operate sufficiently well is often a limiting factor on equaliser performance.

Increasing the symbol rate (baud rate) through a given radio channel causes an increase in the distortion of the transmitted signal and so the use of multilevel modulation schemes

to increase the data rate appears attractive. However, problems of carrier synchronisation in phase modulated systems and of noise interference in multilevel amplidude modulated systems reduce the error rate performance in many applications — typically for rapidly varying channels and for low transmit powers. The development of improved equalisers and detectors which can remove the increased distortion due to higher baud rates would provide an alternative solution for providing higher data capacity radio links. Such a solution would undoubtedly result in a more complex equaliser than current designs and thus would have increased signal processing and power requirements. Current trends in advanced processor development indicate that processing hardware is unlikely to be a technological problem (the cost, however, may be a significant disincentive) but the power demands, especially in systems employing small hand-portable transmitters, may present barriers. In cellular systems the power demand is unlikely to be a problem in base stations allowing advanced techniques to be used on the uplink from mobile terminal to a fixed network.

The aim of this thesis is to identify the causes of the limitations of current equaliser designs whilst operating on time-varying / fading and frequency selective channels and to address the problem of providing solutions to the restrictions. The number of differing radio systems which incorporate equalisation is large and the type of channels for which the equalisers are designed to compensate is varied. Rather than examining a specific system or channel, a general approach to the problems is undertaken followed by investigations into more detailed systems and, in particular, channels.

1.1 Summary of the Thesis

The main body of the thesis is divided into 7 chapters. Chapters 2 and 3 provide an overview of the major components of a radio communications system, showing the relationship of equalisation to other error compensating techniques and associated functions, and a survey of the design and basis of operation of both old and more recent equalisers.

Chapter 4 examines the possibility of applying recurrent neural networks to equalisation. An analysis of the operation of a single node network is undertaken followed by the design of multinode networks and simulation of their performance whilst operating on time-invariant channels.

Chapter 5 characterises the conventional maximum likelihood based equaliser which operates in conjunction with a channel estimator in order to ascertain the main sources of this technique's limitations whilst operating on rapidly time-varying and frequency selective channels.

Chapter 6 investigates Bayesian equalisers, which can be implemented as radial basis function neural networks, with the aim of identifying whether this statistically based equaliser

is less prone to errors in the channel estimate than the maximum likelihood method.

Chapter 7 develops the use of multiple channel estimators for use with maximum likelihood equalisers by modifying the algorithm so that a trade-off between computational complexity and performance is available. The original multiple channel estimator technique has a superior performance but a large computational burden when the channel impulse response becomes large. The modifications developed in this thesis are compared both from a complexity point of view and from a symbol error rate performance aspect.

Finally, Chapter 8 examines the use of soft decisions or confidence information provided by the equaliser on the reliability of its symbol estimates for use in channel estimator tracking. Previously, such confidence methods have been used successfully for aiding channel decoding. The potential for application to channel estimation is investigated and the limitations of the technique identified.

The thesis concludes with a summary of the major findings and identifies further areas of research.

Chapter 2

Radio Communication Systems and Channels

This chapter outlines the functional elements of radio communication systems, the physical attributes of a number of radio channels and discusses methods of modelling these systems and channels. The channel is shown to introduce non-ideal effects on the signal at the radio receiver and this commonly leads to the use of a compensating equaliser in the receiver design. The subject of equalisation is examined more fully in Chapter 3. The air-radio interfaces of a number of current mobile radio communication systems are discussed in Appendix A, highlighting the features which are required in order to compensate for the mobile radio channel.

2.1 Radio Systems

The functional elements of a digital radio communications system are shown in Figure 2.1. The system is designed for the transmission and reception of digital information as opposed to analogue and this has a significant impact on the mechanisms employed in the processing blocks. Digital data systems are ever more increasingly being deployed for reasons ranging from spectral efficiency and system capacity to integration, security and cost.

2.1.1 Source Encoding / Decoding

Depending on the information to be transmitted, the source data may need to be sampled in time and magnitude to produce digital data. For example, a huge use of radio communications is telephony and this requires the analogue voice source to be quantised prior to processing. The function of source coders is to use the statistical redundancy inherent in source signals to

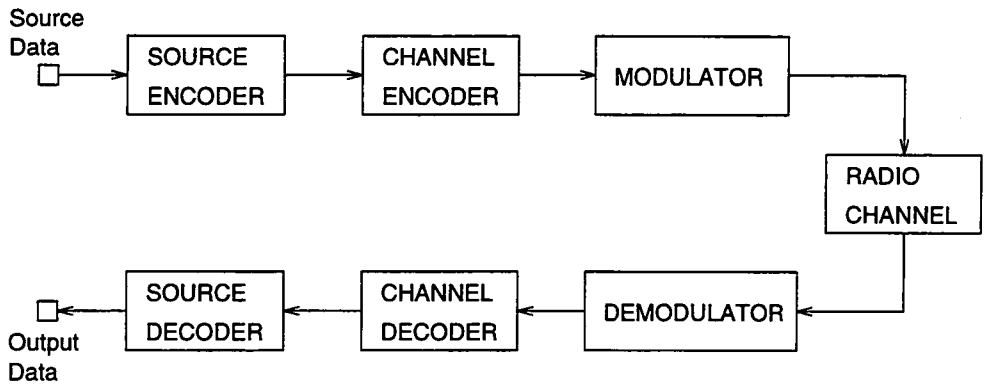

Figure 2.1: Generic representation of the components of a digital radio communications system.

reduce the quantity of data to be transmitted whilst maintaining an acceptable signal quality at the output of the decoder [1]. This allows a greater number of users to use a radio link using the same spectral resource.

The subject of voice coding (vocoders) is large, ranging from pulse code modulation (PCM) to code excited linear prediction (CELP). Coders can be classified by their operation on a sample-by-sample basis (scalar quantisers) or on a block of samples (block or vector quantisers). Scalar quantisers can further be classified as waveform encoders or analysis-by-synthesis coders.

The simplest waveform type encoders (PCM) merely sample and quantise the analogue waveform. Quantising the analogue waveform introduces noise into the signal which impairs the quality of the output signal from the receiver and much of the work in voice coding aims to reduce the bit rate from source encoders whilst maintaining an acceptable quality at the output of the decoder. Improved variants of PCM use the fact that small amplitude speech signals are more common than those with large amplitudes and compress the analogue waveform prior to uniform quantisation. The compression reduces the quantisation noise of the smaller amplitude signals. At the decoder the digitally coded data is converted into analogue and passed through an expander, the inverse of the compressor. Further developments, such as differential PCM (DPCM), quantise the error between the voice signal and a predicted version of it and can reduce the bit rate of the source encoder without degrading the decoded signal quality. The predictor operates on the decoded speech signal (both in the encoder and decoder) and the process uses the fact that successive samples of voice signals sampled at the Nyquist rate (twice the bandwidth of the source) or higher are significantly correlated and thus can be predicted to some degree.

Adaptive DPCM (ADCPM) techniques use the non-stationary statistical properties of speech to adapt the quantisation levels to the current magnitude of the input signal [2]. Thus

if the speech signal is of low magnitude the quantisation step size is reduced together with the magnitude of the quantisation noise. Alternatively the predictor coefficients may be adapted by an algorithm which operates using the encoded differential signal and the decoded source signal. Given that the decoder receives the encoded differential signal without error, the prediction coefficients used in the encoder (and required by the decoder) can be calculated without the need for transmitting the coefficients themselves.

Other forms of waveform encoding include delta modulation (DM), sub-band coding (SBC) and adaptive transform coding — see [2, 3, 4] for further details.

The bit rates of waveform encoders can be reduced to 16kbits/s for speech signals before the quality becomes unacceptable. Analysis-by-synthesis coders can reduce the bit rate further whilst maintaining decoded signal quality and the area is being actively researched. The basis of these types of coders is the modelling of the vocal tract by a linear filter driven by an appropriate input signal to produce the source signal. The voice signal is analysed in terms of this reference model and the coefficients of the resulting model are transmitted to the receiver which then synthesises an approximation of the original voice waveform.

Linear predictive coders (LPC) assumes speech is voiced or unvoiced and models these by passing either a sequence of pulses or random noise through an all-pole, linear, discrete-time filter [5]. The filter coefficients (known as the short term correlation filter) are computed by predicticing the speech sample from the previous few samples (8 to 16 samples) and occurs outside the main encoding loop. The output of the filter is a predicted speech signal and the error between this prediction and the actual speech signals is weighted perceptually in order to shape the noise spectrum so that the voice signal masks the noise. The resulting weighted error signal is minimised, usually to a mean square error criterion and is used to control the excitation of the short term correlation filter and closes the encoding loop (Fig. 2.2). The excitation pulse coefficients and filter coefficients are the model parameters that are transmitted to the receiver.

Further developments of the standard LPC vocoder use multi-pulse excitation (MPE) which does not assume voiced or unvoiced signals. The amplitudes and time positions of the excitation pulses are determined from the result of minimising the perceptually weighted error signal [1]. Regular pulse excitation (RPE) assumes a regular spacing of pulses with variable amplitudes. Code excited linear predictors (CELP) use a codebook of pulse sequences which are chosen from the result of the error minimisation operation.

Block or vector quantisers operate on blocks of samples at a time rather than on a sample-by-sample basis and use statistical dependencies in the source to produce low bit rate coders [2, 4]. The process of vector quantisation may be interpreted as one of pattern matching where the input vector is classified as being in a distinct subdomain of the input vector signal space. The classification process operates by choosing the best subdomain which matches the

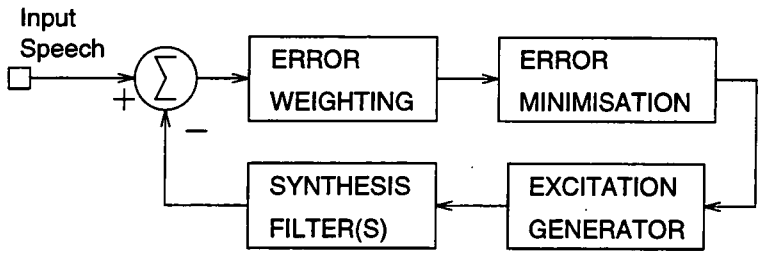

Figure 2.2: Components of an analysis-by-synthesis Linear Predictive Coder (LPC) source coder.

input vector according to some error criterion and algorithms exist which can partition the input signal space in an efficient / 'optimum' manner. These quantisers are not restricted to coding of the input signal. For instance the prediction filter coefficients in LPC vocoders can be quantised as a block such that the coded output has low distortion and this can further reduce the bit rate of LPC coders. Vector quantisation is often applied to source data which is inherently discrete in time and / or magnitude.

2.1.2 Channel Coding / Decoding

The function of channel coding is to facilitate the accurate reception of the source encoded data at the source decoder in the receiver. Errors in the data sequence being fed into the receiver source decoder will degrade, at least, or invalidate the output of the source decoder. Typical communication channels are non-ideal and radio channels are no exception. Errors in the transmitted data will occur due to thermal noise, atmospheric noise and interference amongst others and it is the function of the channel coder / decoder to transform the source encoded data such that the immunity to the types of signal degradation introduced by the channel is maximised.

Some references treat modulation as part of channel coding but for the purposes of this overview the two are separated as per Figure 2.1.

In contrast to source encoders which decrease the bit rate, channel coders introduce redundancy into the signal and hence increase the bit rate. Efficient coders exist for power limited and bandwidth limited applications. If the transmitted signal power is unlimited then the degrading effects of noise can be reduced and the probability of error be made arbitrarily small by increasing the transmitter power. If bandwidth is unlimited then the use of many orthogonal signalling waveforms, i.e. waveforms which are uncorrelated with each other, can reduce the error probability provided the signal-to-noise ratio (SNR) is kept above certain levels. However, both power and spectrum are carefully regulated for radio channels and so channel coders are employed to circumvent the restrictions on power and bandwidth.

There are two main types of channel coders - block and convolutional [2]. Block codes take a block or sequence of input data samples and code them into larger length blocks — the ratio of the input to output block length is known as the code rate, R_c. Convolutional encoders feed their input into a linear shift register and output linear combinations of the shift register's contents. The code rate is defined as the inverse of the ratio of the number of output bits per input bit.

Linear block codes perform operations of addition and multiplication on blocks of input data (containing k information bits) to produce output codes with n information bits and are denoted as (n, k) codes. If the elements of the output code take symbols from a binary alphabet then the coder is known as a binary coder. An important subset of linear block codes are cyclic codes which have properties that enable efficient decoding. Examples of block codes are Hamming, Golay, Bose-Chaudhuri-Hocquenghem (BCH) and Reed-Solomon codes. In the receiver the channel decoder has an increased performance if the samples from the output of the demodulator block (Fig. 2.1) are not quantised — soft-decision decoding, but this increases the computational load on the demodulator. Block codes have error detecting and usually error correcting capabilities. This latter property allows forward error correction (FEC).

Convolutional coders consist of an L stage shift register with each stage containing k information bits. The parameter L is known as the constraint length. Each input symbol (containing k information bits) is fed into the delay line and n binary symbols (bits) are formed by modulo-2 additions of combinations of the Lk bits in the delay line and are output from the coder sequentially. Decoding in the receiver is performed via the Viterbi algorithm if the constraint length is small and via a sequential decoding process if the constraint length is large. The trade-off is between the improved performance of the Viterbi versus the reduced complexity of the sequential decoder. As with block codes, soft-decisions from the output of the demodulator can be used to improve quality of the Viterbi decoder.

Both block and convolutional coders increase the bandwidth of the transmitted signal as they increase the bit rate — the expansion is equal to $1/R_c$. For bandwith limited communication channels alternative channel coding techniques exist which increase the number of signalling symbols rather than the bit rate in order to introduce the coding redundancy and hence they have an increased spectrum efficiency. This form of coding combines some of the functionality of the modulator (Fig. 2.1) as the mapping of data bits onto signalling symbols is performed in a coded manner by making the choice of waveform dependent on preceding symbols / data bits. Trellis coded modulation (TCM) is an example of this form of coding and uses set partitioning principles to maximise the Euclidean distance between consecutive symbols and hence reduce the probability that the symbols will be erroneously classified in the receiver [6]. The decoder can be implemented by application of the Viterbi algorithm and can use soft-decisions from the 'demodulator' to increase performance.

Radio channels often have characteristics which result in bursts of errors from the output of the demodulator particularly when the received signal power is subject to fading. Both channel decoders have a limited capability to correct errors from the demodulator and if the errors are clustered together then the performance of the channel decoders is reduced. Interleaving is a technique used to spread the errors out over a number of blocks or disperse them across the demodulated symbol sequence. This reduces the burstiness of the error events and enables an increased performance from the channel decoder. Interleaving occurs in the transmitter at the output of the channel coder, prior to modulation, and deinterleaving occurs in the receiver at the output of the demodulator prior to channel decoding. A number of different interleavers exist — diagonal, block and convolutional [5].

2.1.3 Modulation / Demodulation

The principle functions of a digital modulator are to map data bits to signalling symbols / waveforms, shape the symbols and frequency translate the baseband signal to either an intermediate frequency (IF) or the carrier radio frequency (RF). The modulation process transforms the data bits from the discrete to the continuous time domain. The modulated symbols are often represented by a constellation diagram which shows, in a complex baseband format, the mappings of the data bits to a discrete set of amplitudes, phases, frequencies or a combination of amplitude and phase (Fig. 2.3). Demodulators transform the corrupted received waveforms back to discrete data bits.

The systems used to transmit and receive the signals and the channels used to convey them influence a number of factors affecting the choice of modulation. Radio channels have a limited bandwidth (both physically and regulatory) and in order to acheive a high data capacity, the spectral efficiency (the ratio of the information rate to the bandwidth required — measured in bits/s/Hz) of the modulator must be taken into account. Spectral efficiency is increased by using a larger number of symbols or modulation levels whilst keeping the bandwidth of each element the same. However, this is achieved at the expense of modulator / demodulator complexity. In land cellular mobile radio networks, bandwidth reductions obtained by using multilevel modulation are offset by a larger frequency reuse factor for sufficient carrier-interference (C/I) ratio. In order to achieve an adequate received SNR for the lower energy symbols, the transmit power must be increased resulting in the higher energy symbols causing increased interference in nearby cells [7].

Bandwidth constraints affect the shape and length of the signalling symbol as waveforms occupying only the data bit period have large spectral sidelobes. Application of Nyquist's criterion leads to the design of bandlimited signal elements which span several symbol periods but do not introduce intersymbol interference (ISI) when transmitted across otherwise ideal channels. Such waveforms have zero crossing points at all symbol-period sampling instances

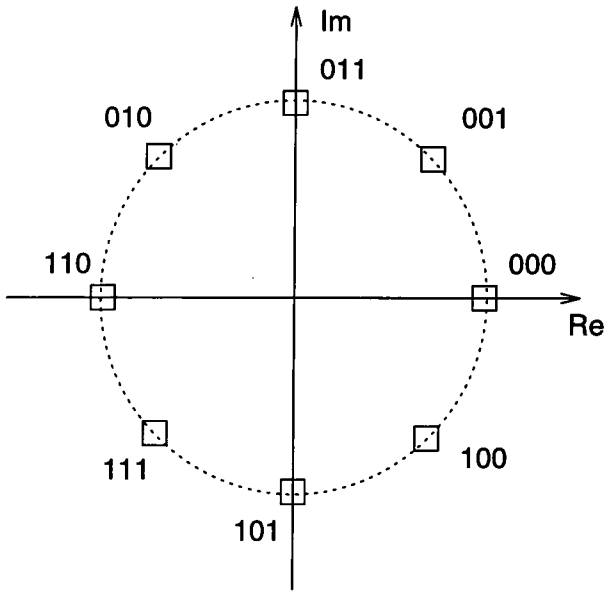

Figure 2.3: Constellation diagram for an 8-PSK modulator.The numbers represent the Gray coded bit sequences resulting in each symbol.

other than the instant pertaining to the symbol itself. Thus when the demodulator samples the received waveform at the required sampling instances, waveforms of prior and future symbols do not affect the current sample. Pulse shaping occurs in the modulation process after the mapping the data bits to the symbol amplitudes, phases or frequencies. A typical pulse shape is the raised cosine:

$$g(t) = \frac{\cos(\alpha\pi t/T_s)}{1 - 4\alpha^2 t^2/T_s^2} \operatorname{sinc}(t/T_s) \tag{2.1}$$

where T_s is the symbol period and α ($0 \leq \alpha \leq 1$) is the roll-off parameter. When $\alpha = 0$ the pulse occupies the Nyquist minimum bandwidth ($= 1/2T_s$) giving the theoretical maximum spectrum efficiency [3]. The filter creating this pulse shape is typically split between the modulator and demodulator filters, each implementing a truncated approximation of the root-raised cosine filter. In practical implementations of such systems, ISI appears due to sampler phase jitter and the inaccuracy of the pulse shaping filter.

Other bandwidth related factors include the high spectral sidelobes caused by abrupt switching of the signalling waveforms as occurs in memoryless modulation methods such as pulse amplitude modulation (PAM), quadrature amplitude modulation (QAM) and phase shift keying (PSK). Modulators with memory lead to continuous phase modulation (CPM) techniques such as continuous phase frequency shift keying (CPFSK), minimum shift keying (MSK), and offset quadrature phase shift keying (OQPSK) which have reduced spectral sidelobes. See [8] for further details of digital modulation techniques for mobile radio systems.

Power efficient amplifiers prior to the transmitting antenna are usually nonlinear leading to spectral leakage. Modulation schemes which have constant envelope power such as Gaussian MSK (GMSK) allow such nonlinear amplifiers to operate at constant power and thus do not cause high levels of spectral leakage.

The methods of detecting and differentiating the received signals heavily influences the choice of signalling waveform. By choosing a set of mutually orthogonal waveforms (i.e. waveforms which have zero cross-correlation) the receiver can correlate the received signal with each of the posssible transmitted waveforms and select the one with the highest correlation. This method also maximises the received SNR ($\frac{E_b}{N_0}$ — energy per bit/noise power) and is known as matched filter or correlation demodulation [2]. The most common detection error is between adjacent symbols in the constellation diagram and by choosing a Gray coded mapping of data bits to symbols in the modulator, the data bit error rate can be reduced since adjacent symbols differ only by a single bit.

Typical radio receivers are required to detect low power signals and the effect of noise generated in the receiver analogue components is to degrade the receiver's ability to detect these weak signals and to differentiate between the received symbols. Analogue components (e.g. antennæ, filters, mixers and amplifiers) will introduce noise into the received signal and it is important to keep noise figures in the amplifiers low. The noise figure is defined as the ratio of the input SNR to the output SNR [4]. The noise added in the receiver is modelled as an uncorrelated or white process. In a matched filter demodulator, the filtering process colours the additive white noise and this has a limiting effect on the detection mechanisms as the noise becomes correlated with previous noise samples. The situation may be alleviated by introducing a discrete-time noise whitening filter which decorrelates the noise between successive samples. This filter may be cascaded with the matched filter or incorporated with a sampled matched filter.

Synchronisation of the receiver to the transmitted waveform has a significant effect on the quality of the detection process. The synchronisation issues for the demodulator are of carrier and symbol synchronisation [9]. Carrier synchronisation involves generating a local carrier reference with a phase ideally locked to the transmitter carrier and allows the bandpass received signal to be mixed down to baseband without frequency or phase offsets. Coherent demodulators require the carrier phase (and therefore frequency) to be known and provide a better performance than incoherent demodulators which do not require such constraints. The receiver carrier may be generated by phase locked loops (PLLs) operating on the received signal when an unmodulated carrier component is present. This requires power to be used on transmitting a carrier / pilot tone and so alternative techniques have been developed to derive the carrier from signals transmitted in a suppressed carrier format. These methods involve squaring the received signal to generate a component at twice the carrier frequency and using a PLL followed by a frequency divider to derive the carrier reference. Alternative

methods use decision directed or maximum likelihood (ML) techniques.

Constant phase or frequency offsets between the receiver and transmitter carriers result in a fixed phase rotational offset or a constant rotation of the baseband signal constellation diagram and leads to crosstalk between the in-phase (I) and quadrature-phase (Q) components of the complex baseband signal. In general, higher bandwidth-efficient modulation schemes require better carrier synchronisation — i.e. their demodulators are less tolerant of carrier phase errors, but extracting the phase information from the received signal becomes more onerous. The phase ambiguity resulting from a constant carrier phase error can be compensated for by differential encoding in the modulator, e.g. differentially encoded quadrature phase shift keying (DEQPSK). This method encodes the data bits in the phase difference between succesive symbols rather than in the phase itself. In the receiver the decoding is performed after coherent detection in DEQPSK systems but the received signal can be detected incoherently using a differential detector (DQPSK systems).

Symbol synchronisation involves generating a timing clock to sample the continuous time domain signal after it has been mixed down to baseband. The sampling process transfers the signal back into the discrete time domain. The correct timing of the sampler ensures that the baseband waveform is sampled at the peaks of the symbol waveforms for maximum immunity to noise and intersymbol interference resulting from time dispersion of the communication channel or modulation waveforms which span more than one symbol period. In the case of correlation type demodulators the symbol timing is required to synchronise the sampling of the output of the correlators. Common methods of symbol synchronisation use decision directed or maximum likelihood techniques.

Radio communication channels corrupt the transmitted signal in a number of aspects and are detailed in Section 2.2. Compensation techniques can be implemented in both the modulator and demodulator. Partial response signalling is a modulation technique which introduces a controlled amount of ISI between symbols and is suitable for radio channels with large amplitude variations within their bandwidth. A subcomponent of the demodulator, the equaliser, is introduced to compensate for the time dispersive characteristics of radio channels which would otherwise degrade the detection process. Equalisation is discussed in detail in Chapter 3 onwards. Another method which can be used independently or in conjunction with equalisation is the use of diversity. This technique is particularly useful when the channel does not introduce ISI but when the signal power level varies in time. This fading is discussed more fully in Section 2.2.1. There are a number of diversity methods including spatial, time, polarisation and frequency diversity. Each method is based on the idea that if the received signal power drops due to a radio path fading, then another, independent, path is unlikely to be mid-fade at the same time. Thus the received signal can be taken from this alternative path whose SNR is greater than the original. The most common form is space /antenna diversity where two or more antennæ are used to receive the signal.

2.2 Radio Channels

Radio communication channels introduce a number of nonideal effects on the transmitted signal. The ideal channel introduces an attenuation and phase shift which remain constant across the signalling bandwidth and over time. Coupled with additive noise in the receiver 'front-end' components, this ideal channel is called the additive white Gaussian noise (AWGN) channel and is often the reference channel to which many systems are compared. The attenuation is due to path loss — the power density of the transmitted signal decreases with an inverse square law with distance from the transmitter antenna under free-space conditions but in many radio channels (e.g. indoor) the decrease is of a higher power than 2 [10]. The attenuation may be compensated for by automatic gain control (AGC) circuitry. The phase shift of the received signal is affected by the length of the electrical path between transmitter and receiver and may be partially compensated for by carrier synchronisation techniques.

The nonideal effects may be subdivided into deterministic and nondeterministic. The latter includes additive noise-like interference such as atmospheric noise (common in HF radio systems) and co-channel interference arising from spectral leakage from adjacent frequency bands, geographically remote users transmitting on the same frequency band (land cellular radio systems) or users using the same bandwidth (direct sequence spread-spectrum systems e.g. code division multiple access, CDMA systems). As these types of interference are random, little can be done to remove them from the received signal once they have affected the signal and care is taken in system design to limit the areas and the magnitude in which the noise can impinge on the transmitted signal.

Deterministic effects arise from the transmitted signal itself and generally result in intersymbol interference (ISI) due to nonuniform amplitude and phase responses of the channel over the signalling bandwidth. In the time domain these nonuniformities manifest themselves as a spreading of the channel impulse response (CIR) which, ideally, is just a delta function with a delay due to the propagation time. The cause of the variations in amplitude and phase responses is multipath propagation — the existence of more than one path between transmitter and receiver along which the transmitted signal may propagate. Variations in the electrical length of these paths result in slightly different propagation times and the signals arrive with a phase shift relative to one another resulting in constructive or destructive interference [5]. The transmitted signal contains frequencies across a certain bandwidth and thus the relative phase of the multipath signals varies across the band leading to variations in amplitude and phase response. Small variations in propagation time between paths result in larger deviations in amplitude and phase response the larger the signal bandwidth and carrier frequency. Time delays between multipath components which are significantly less than the transmitted symbol period do not cause ISI.

Multipath conditions arise in many radio channels from line-of-sight (LOS) microwave,

high frequency (HF) and land mobile radio channels. The microwave multipath channel occurs when atmospheric conditions lead to stratification of the troposphere with variations in moisture, temperature and refractive index between the layers resulting in many discrete paths along which the radio wave can propagate [11]. HF radio channels have multiple paths arising from multiple reflections between layers in the ionosphere and/or the earth. Land mobile radio channels often do not have a line-of-sight path between transmitter and receiver and rely on the scattering, diffraction and reflection of the radio waves by obstacles to acheive a communication channel.

Another classification of radio channels is whether they are time-variant or time-invariant. Time-invariant channels remain constant over the duration of a large number of symbol periods and are, in general, easier to correct. However, many practical radio channels vary much more rapidly compared to the symbol duration and result in a significant degradation in the bit error rate from conventional detectors. Receiver structures designed for the time-varying nature of the channel exist but provide far from ideal performance.

2.2.1 Time-Variant Radio Channels

The characterisation and modelling of time-varying channels has received considerable attention and is of practical significance particularly in the optimisation of mobile radio networks [12, 13, 14]. This section summarises the salient features of time-varying channels and compares the character of a number of such channels.

Time-varying channels may be classified as frequency selective or frequency non-selective [2]. In the latter, the impulse response of the channel at any given instance in time does not cause intersymbol interference, i.e. the CIR delay spread is significantly less than the transmitted symbol period. The spectrum of the channel over the transmission bandwidth is flat and thus frequency non-selective channels are also termed 'flat fading' channels. As the transmitted bit rate increases (decreasing the symbol period) this type of effective sampled channel becomes less common and ISI becomes more prevalent.

Due to the relative motion between transmitter and receiver or within the propagating medium itself, the time delay, amplitude, phase and Doppler shift of the received signal varies with time and in an essentially random manner. Multipath effects cause the radio waves to interfere either constuctively or destructively resulting in varying signal strengths as motion occurs. This is primarily due to the significant differences in phase between the multipath signals which occur with only a small movement of the transmitter, receiver or propagation medium at the transmission frequencies used. If the statistical properties of these random variations are constant in time then the channel is termed 'fading'. This type of fading should be distinguished from 'fades' in the spectrum — frequency selectivity. Fading radio channels which introduce ISI (i.e. are frequency selective) are not neccesarily detrimental. Indeed the

diversity introduced by the paths causing ISI can be used to provide signal power when the attenuation of other paths increase during fades.

The baseband representation of the channels have complex paths to represent the amplitude and phase variations of the bandpass channel [2]. An individual path is typically modelled as a white, zero mean complex Gaussian process with independent real and imaginary components. This results in a signal envelope with a Rayleigh probability density function and this type of channel is termed a Rayleigh fading channel. The time correlations are described by filtering the Gaussian process components with a filter whose frequency response matches the square root of the Doppler power spectrum of the channel. If this filter is linear, then the filtered Gaussian components will also be Gaussian and the resulting complex process will remain Rayleigh distributed. If the channel has fixed scatterers, reflectors or a dominant line-of-sight path in addition to random Gaussian variations then the channel is termed a Ricean fading channel. The validity of these models to various applications is discussed later in the section.

The multipath nature of channels is often characterized, in the time domain, by a parameter T_m, the multipath delay spread of the channel, which is the range of delay values over which the CIR is much greater than zero. If the symbol period is greater than this value then the channel is treated as frequency non-selective. The frequency domain parameter ($\approx 1/T_m$) is called the coherence bandwidth of the channel. The time-varying nature of individual paths is characterised by the coherence time, T_c, in the time domain or by the Doppler spread of the channel in the frequency domain. For a maximum Doppler shift of f_d, the coherence time is often defined as $T_C = \frac{0.432}{f_d}$ [4]. A channel with large Doppler spread has more rapid variations in amplitude and phase. The Bello system of functions relate these channel characteristics in the time and frequency domains with a 2-dimensional scattering function representing both time and frequency dispersion of the channel [5].

Land Mobile Radio Channels

Land mobile radio channels have varying characteristics depending on the local terrain [15]. In urban areas the CIR is modelled as having a typical multipath delay spread of 5 μs which results in a significant amount of ISI for symbol rates in the order of 270 kBaud (e.g. GSM — Global System for Mobile communications). The paths within the channel are Rayleigh fading with the time variations of all paths being modelled by the classical Doppler spectrum (a simplification as some paths have Ricean-type characteristics). This Doppler spectrum model is a worst case scenario which assumes that there are a large number of uncorrelated signals arriving with a uniformly distributed angle of incidence at the isotropic receive antenna and with similar time delays [5]. This large number of local paths is due to multiple reflectors and scatterers present in an urban environment. The assumption of uncorrelated paths is due to the fact that small changes in path length result in significant changes in phase of the

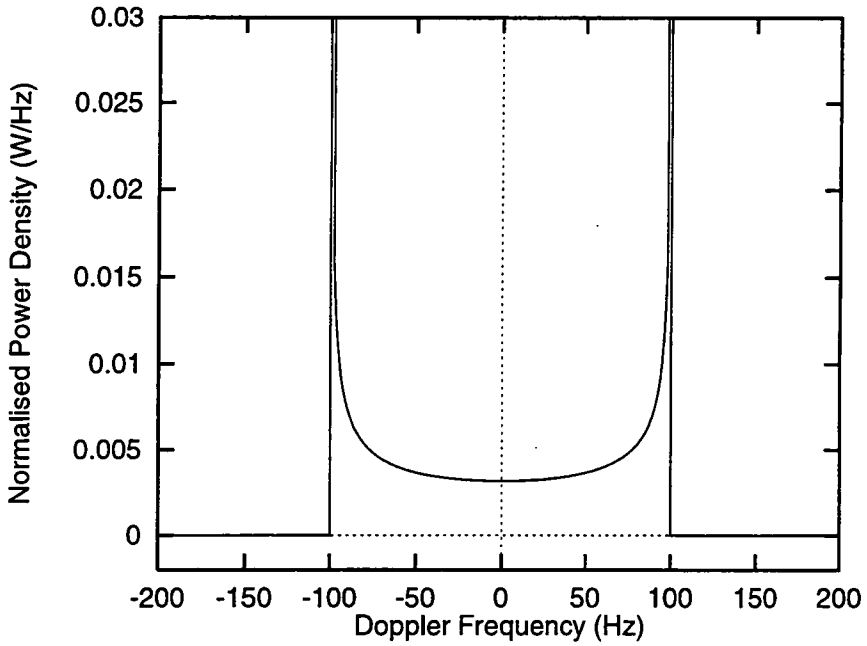

Figure 2.4: Power spectrum of baseband classical Doppler spreading. In this example the maximum Doppler shift is ±100 Hz.

RF signal — typical wavelengths being in the order of 0.3 m. As the mobile receiver moves directly towards / away from the transmitter, the RF experiences a maximum Doppler shift of $f_d = \pm\frac{f_t v}{c}$, where f_t is the RF carrier frequency and c, v are the velocity of light and mobile velocity (in ms^{-1}) respectively. For a carrier frequency of 900 MHz and a mobile velocity of 120 km/hr, the maximum Doppler shift is approximately 100 Hz. At other angles of incidence the magnitude of the Doppler shift is not as great and the spectrum of combined signal from the local paths, together with the assumption of a uniform distribution of angles of incidence, leads to the baseband classical Doppler power spectrum, (Fig. 2.4),

$$S(f) = \begin{cases} \frac{1}{\pi f_d \sqrt{1-\frac{f^2}{f_d^2}}} & -f_d < f < +f_d \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

For rural areas the channel is modelled as having a multipath delay spread in the order of 0.5 μs, resulting in negligible ISI and with a dominant main path modelled as having Ricean characteristics — a specular component / Doppler shift together with a Doppler spread (see Figure 2.5). The Hilly Terrain model has a burst of local paths with time delays in the region of 1 μs and a burst of paths with delays from 15 μs to 20 μs resulting from distant reflections. All paths in this model have classical Doppler spread characteristics. Although the multipath spread causes ISI, it can also be viewed as diversity which enables receivers to operate even when an individual path's signal strength drops during a fade.

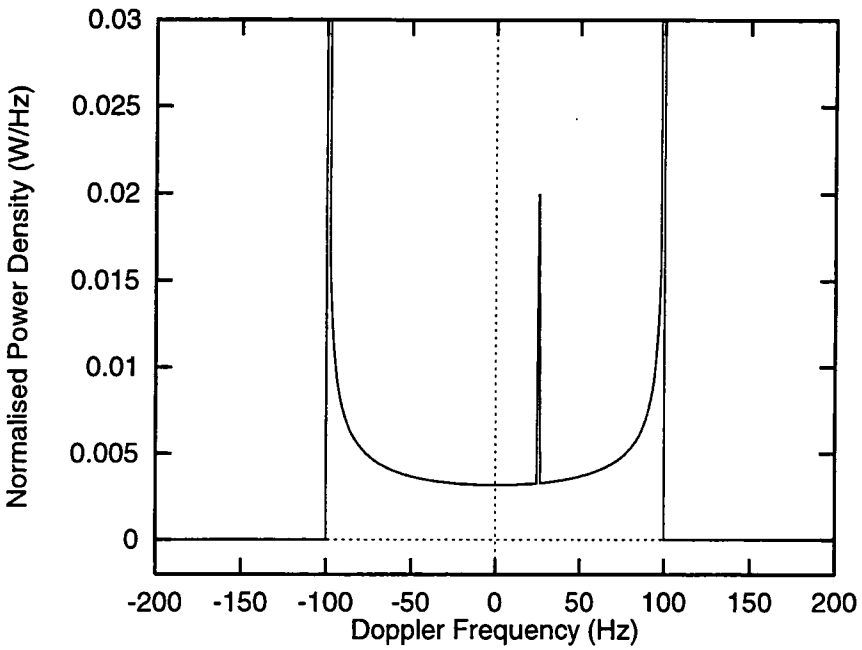

Figure 2.5: Power spectrum of a Ricean spreading function. In this example the maximum Doppler shift is ± 100 Hz with a specular component at $+25$ Hz.

Strictly, fading channels occur only when the relative velocity between transmitter and receiver is constant and thus practical land mobile communication channels are statistically non-stationary. However, the variations in velocity are significantly slower than the transmit bit rate, allowing the channel to be reasonably modelled with constant statistical parameters (Doppler spread etc.).

Tropospheric Scatter Channels

Tropospheric scatter communication links suffer from multipath fading due to the changing characteristics of the troposphere [16, 17]. These radio systems provide link lengths from 100 to 1000 km (i.e. much greater than line-of-sight systems) by radiating power of the order of 1 — 10 kW into the trophosphere in a narrow beam a few degrees above the horizon with RF carriers in the UHF band (900 MHz — 5 GHz). The troposphere extends from the earth's surface up to 8 to 10km high. Scattering occurs due to turbulence in the troposphere creating many small volumes of air with varying refractive index which reflect or refract incident radio waves in all directions. Alternatively, stratification of the troposhere creates layers of air with differing refractive indicies and affects the propagation of radio waves through refraction and partial reflection at each layer / boundary.

There are no resolvable discrete paths, unlike some land mobile radio channels and HF channels, but a continous range of paths with differing propagation delays. The average

envelope of these multipath components is Gaussian with a spread factor defined by 2σ where σ^2 is the variance of the Gaussian process or σ is the RMS width of the mean square envelope of the channel impulse response. This spread factor is sometimes normalised to the symbol period, T_s, resulting in the spread factor definition $\frac{2\sigma}{T_s}$ [18]. The multipath spread of typical channels are a function of link length and are in the region of $T_m \approx 6\sigma \approx 1\mu s$ for 300 km links [19, 20]. Typical symbol rates are in the order of 8 MBaud (QPSK modulation) resulting in the normalised spread values in the order of 1 to 3.

The time variations of troposcatter channels are divided into slow and fast variations with the timescale of the slow variations being in the order of days. The effects of the fast variations are more appropriate in the timescale of the symbol period and each path of the sampled channel impulse response is treated as Rayleigh fading with the changes having a Gaussian shaped Doppler power spectrum and Doppler spreads of up to 20 Hz.

HF Channels

High Frequency (HF) radio systems use the ionosphere to provide beyond the line-of-sight communication links of greater length than tropospheric scatter links. The ionosphere is caused by solar radiation (cosmic, X-rays and UV) which ionises the atmosphere and comprises of a number of layers which reflect and refract RF signals with carrier frequencies of 3 — 30 MHz. The D layer lies between 50 and 90 km above the earth's surface and only exists during the day with peak ionisation at midday during the summer months [21]. For communication purposes this layer is principally an attenuator. The E layer is at a height of 90 to 130 km and contains the highest ion density in the ionosphere. The layer is caused by the ionisation of atomic oxygen and is at a maximum level around noon, dropping off to residual levels at night. The F layers (F1 and F2) are above the E layer to a height of 450 km and exist during both day and night. Although the ionising radiation is greatest in the F layer, the ion density is not as great as the E layer as the atmosphere becomes increasingly rarified. The F layers and to some extent the E layer exist during the night (and absence of solar radiation) as the probability of electron / ion recombination is small for the lower density air.

Depending on the prevailing conditions and communication link length, the channel has predominantly one or two paths resulting from single and double hops (reflections between the ionosphere's layers and / or the earth's surface) and the constant motion of the ions results in a fading channel [22]. For a single path, the flat-fading channel has fade rates in the order of 0.2 Hz to 1 Hz. For the two path case, each path has the same mean power or the second path is up to 40 dB lower and the paths are separated in time by a delay of 0.5 ms to 3 ms. Symbol rates over HF links are in the order of 2.4 kBaud resulting in significant ISI. The paths fade independently with a Gaussian spectrum and have Doppler spreads in the order of 0.1 Hz to 10 Hz in the case of flutter fading (caused by aircraft) [14,

23, 24].

Atmospheric noise is, in practice, impulsive in nature with large, rapid variations but relatively constant average value [25]. However the noise is often modelled as being Gaussian for convenience.

2.3 Radio Communication System and Channel Modelling

The majority of the results presented in this thesis have been obtained through the use of computer simulations, in particular Monte Carlo analysis to approximate the symbol error rates. The radio communication systems have been simplified, where appropriate, to reduce the computational overhead and standard baseband representations of the RF carrier waveforms and communication channels have been used throughout [2]. The SPW software package [26] has been used extensively to model and simulate the characteristics and performance of both channels and radio systems.

From the previous sections in this chapter, it can be seen that there are a large number of factors affecting the overall performance of a radio communication system, from source coding, channel coding, modulation format and radio channel. Thus, for comparative purposes, many of the system and channel models have been kept constant or generalised. In particular no source coding, channel coding or interleaving has been applied and the modulation format has been restricted to QPSK / QAM or multi-level PAM. The emphasis has been on comparing the performance of various equalisation strategies on more or less similar systems rather than examining the quantative effects of other system parameters on the performance of the equalisers.

All radio channels have been simulated rather than using measurements from a real channel as the simulation methods can be easily applied to different radio channels. In addition the relative performances between various systems operating on a simulated channel provide a better indication of the relative merits of operational systems rather than absolute performance measurements using a snapshot of a real channel [27]. In the latter case, the system is often tailored to the sample of the channel which may not be representative of typical channels in the longer term.

The SPW package simulates the communication systems on a sample-by-sample basis with each sample passing through the whole system prior to the next sample being simulated. The sample rate is not restricted to the symbol rate and thus the continuous time domain portions of the system can be better approximated by using more than one sample per symbol. This allows the effects of transmit / receive filters and symbol clock synchronisation to be studied and to provide more accurate modelling of the continuous nature of the channel.

For simulation purposes the continuous time CIR is sampled in time and implemented

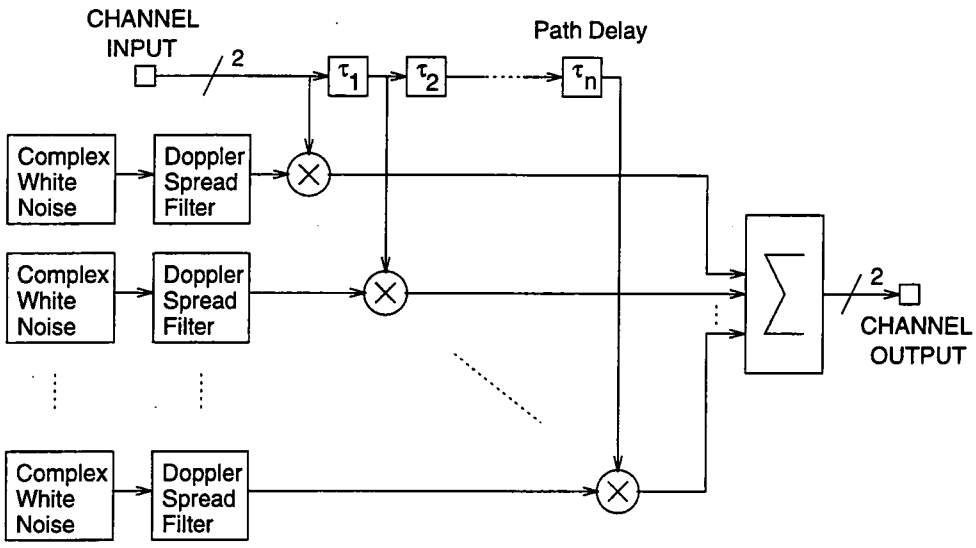

Figure 2.6: Generic fading channel model.

as an FIR filter with either real or complex and time-varying or constant taps depending on the radio channel being simulated. The propagation delay of the channel is ignored resulting in the first path / tap with finite attenuation being assigned zero propagation delay. The majority of the simulations examine the effects of time-varying and complex channel taps (as a result of the baseband representation). For a Rayleigh fading channel, these taps are generated by passing complex white Gaussian noise through a filter whose frequency response is matched to the Doppler spread characteristics of the channel being modelled (Fig. 2.6) [28, 29]. The delay elements (τ_1 etc.) in the channel FIR filter correspond to the relative delay between the paths of the channel in question and thus the channel should be simulated at a rate dependent on an integer multiple of the symbol rate. For general time-varying channel models the delays have been set equal to the transmitted symbol period for reasons of increased simulation speeds. These types of channels are easier to equalise and thus the method allows suitable equalisation techniques to be identified prior to more realistic simulations.

As the Doppler spread of most channels is much lower than the symbol rate, simulation efficiency can be improved by generating the channel filter taps at a slow rate and interpolating up to the symbol rate using a polyphase filter. The time domain Doppler filter is obtained by sampling the frequency response of the channel's Doppler characteristics, taking the inverse DFT and windowing the resulting time-domain filter taps. Note that this technique broadens the 'true' frequency response but, since this response is only an approximation to the general behaviour of the channel, the effect is not considered problematic as long as the level crossing rate (the rate at which the power attenuation of the path rises above a certain level) is not distorted significantly. In a real situation it is quite possible that the maximum Doppler shift

is greater than that due to the relative velocity between transmitter and receiver as there may well be moving reflectors (vehicles, for example).

2.4 Summary

The fundamental components of a digital radio communications system have been described together with some of the factors affecting system configuration and simulation. The characteristics of general and specific radio channels have been discussed, in particular the delay spread and time-varying properties, and the effects of the channel on the transmitted signal have been described. The multipath nature of many radio channels causes ISI if the delay spread of the channel is a significant fraction of the transmitted symbol period or greater. In the receiver an equaliser is employed in the demodulator to remove the ISI effects introduced by the channel in order to facilitate the symbol classification of the received signal samples. In the next chapter, the process of equalisation and the structure of equaliser implementations, both conventional and novel, are described.

Chapter 3

Equalisation

This chapter details the functions of the equaliser and describes both conventional structures and more recently developed structures based on neural networks. Adaptive equalisation schemes are discussed and adaptation algorithms such as LMS and RLS are examined in the context of equaliser parameter adjustment. The application of adaptive equalisers for time-varying / fading communication channels is summarised — detailed explanations of specific structures relevant to particular applications are given in later chapters.

3.1 Principles of Equalisation

The function of an equaliser is to process the received signals in order to aid their subsequent classification by a decision device / detector in the receiver demodulator (Fig. 3.1). The mechanism of classifying the processed received signals is highly dependent to the equalisation process or *vica versa* and so the combination of equaliser and detector structures are considered together. The equaliser is primarily designed to counter the adverse effects of nonideal channel characteristics — intersymbol interference (ISI), but may be designed to correct for distortions generated by transmitter and receiver circuitry in addition. Typical communication systems employ different compensators to minimise the effects of differing distortion mechanisms. This thesis concentrates upon channel equalisation as the channel is often the most significant source of distortion.

The basic structure of an equaliser is shown in Figure 3.2. Equalisers / detectors may be categorised by whether they operate on a symbol-by-symbol basis or classify a sequence of symbols [30]. For symbol-by-symbol based equalisers, the received signal classification process estimates the transmitted symbol one at a time rather than a sequence of transmitted symbols *en masse*. Both forms use a sequence of sampled received signals in order to produce their estimate. Some equalisers attempt to remove the ISI introduced by the channel by

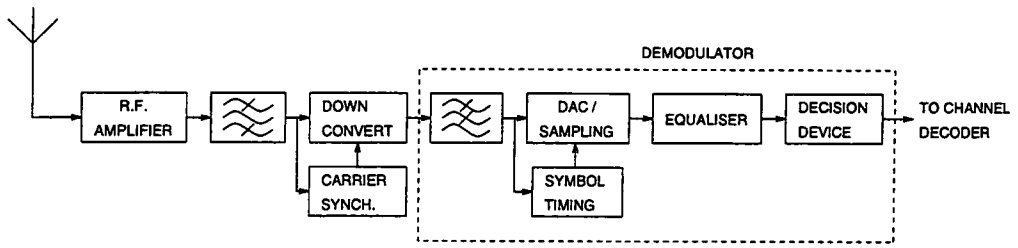

Figure 3.1: Schematic of a general receiver structure.

producing an approximation of the transmitted waveform from the received signal which is then fed to the detector. These equalisers are independent of the modulation scheme used and the design is based on the consideration of the channel in isolation from the modulator. However, the form of the detector is dependent on the modulator / transmitted waveforms and may involve correlating the processed waveform with replicas of the possible transmitted waveforms. Other equalisers include a correlating / comparison function within their structure and output a metric for each member of the transmitted symbol alphabet which is then fed to the detector to select the best metric. These forms of equalisers are dependent on the modulation scheme employed and may base their metric formulation on probability theory.

The received signal is commonly sampled prior to the equaliser either at the symbol rate or at an integer multiple of this rate. The received signal should be passed through a receive filter to limit out-of-band noise prior to sampling and in PAM systems the filter is ideally a whitened matched filter — the filter being matched to the combined filter obtained from the convolution of the transmit pulse shaping filter and the channel model filter. Whitening ensures that noise is decorrelated between successive samples (assuming symbol rate sampling). However, if the channel is not known *a priori* then the receive filter is typically a root raised cosine filter (as is the transmit, pulse shaping filter) to limit out-of-band noise and ensure no ISI in an ideal channel situation — root raised cosine filters do result in ISI at the symbol rate sampling instants as the frequency response is not skew-symmetric about the cut-off frequncy but the convolution of two such filters does satisfy Nyquist's criterion [3].

For symbol-rate sampling, the combined baseband transfer function of the transmit, channel and receive filters is given by:

$$h(\tau, k) = h_i(k) \qquad \tau = iT_s \tag{3.1}$$

and the symbol-rate received signal samples are given by:

$$r(k) = \sum_{i=0}^{L} h_i(k)x(k-i) + n(k) \tag{3.2}$$

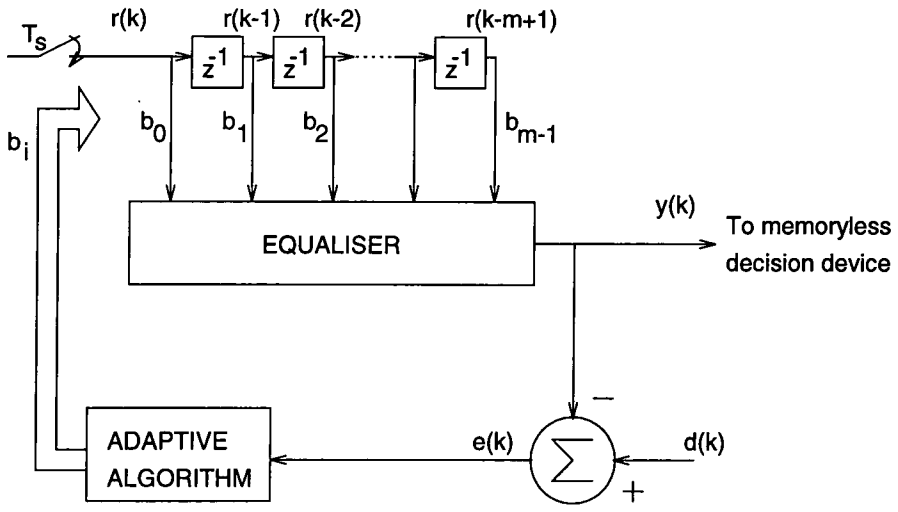

Figure 3.2: Schematic of a general equaliser structure.

where τ is the delay variable, T_s is the symbol period and LT_s is the length of the combined impulse response giving a symbol-rate sampled impulse response of $(L+1)$ components. $h_i(k)$ are the components of the sampled impulse response, $x(k)$ are the transmitted symbols and $n(k)$ is the additive noise term. In general, all these terms are complex valued.

The ISI introduced by the channel causes the received signal samples to differ from the corresponding sampled transmitted signal but, in the absence of noise, samples of the received signal take on distinct values or states. For a system employing an M-ary PAM scheme, the equaliser attempts to transform the received signals into one of the M bands for classification by the detector, typically an M-level quantiser. The equaliser and detector / decision device form sets of decision boundaries in which all states lying within the bounds of a particular set of boundaries are classified as being caused by one symbol. Noise in the received signal will corrupt the states and may move them into erroneous decision regions. The dimension of the decision boundaries depends upon the number of received states and previously classified symbols used in their calculation. In general, increasing the dimension of the decision boundary increases the minimum distance between the boundaries and received signal states, therefore reducing the probability of incorrect classification. The decision boundary may be linear or nonlinear depending on the structure of the equaliser.

Table 3.1 shows an example of the received signal states $(r'(k), r'(k-1))$ for a channel with a transfer function (sampled at the symbol rate) $H(z) = 0.348 - 0.870z^{-1} + 0.348z^{-2}$, a transmitter using a bipolar PAM scheme and no additive receiver / channel noise. Figure 3.3 shows them plotted with a decision delay of one sample period and a typical, linear decision boundary separating the two sets of states. As the channel may be represented as an FIR filter, the received signal states occur in a defined order due to the sequential nature of

$x(k)$	$x(k-1)$	$x(k-2)$	$x(k-3)$	$r'(k)$	$r'(k-1)$
1	1	1	1	1.566	1.566
1	1	1	-1	1.566	0.870
1	1	-1	1	0.870	-0.174
1	1	-1	-1	0.870	-0.870
1	-1	1	1	-0.174	0.870
1	-1	1	-1	-0.174	0.174
1	-1	-1	1	-0.870	-0.870
1	-1	-1	-1	-0.870	-1.566
-1	1	1	1	0.870	1.566
-1	1	1	-1	0.870	0.870
-1	1	-1	1	0.174	-0.174
-1	1	-1	-1	0.174	-0.870
-1	-1	1	1	-0.870	0.870
-1	-1	1	-1	-0.870	0.174
-1	-1	-1	1	-1.566	-0.870
-1	-1	-1	-1	-1.566	-1.566

Table 3.1: Received signal states for a channel with transfer function $H(z) = 0.348 - 0.870z^{-1} + 0.348z^{-2}$.

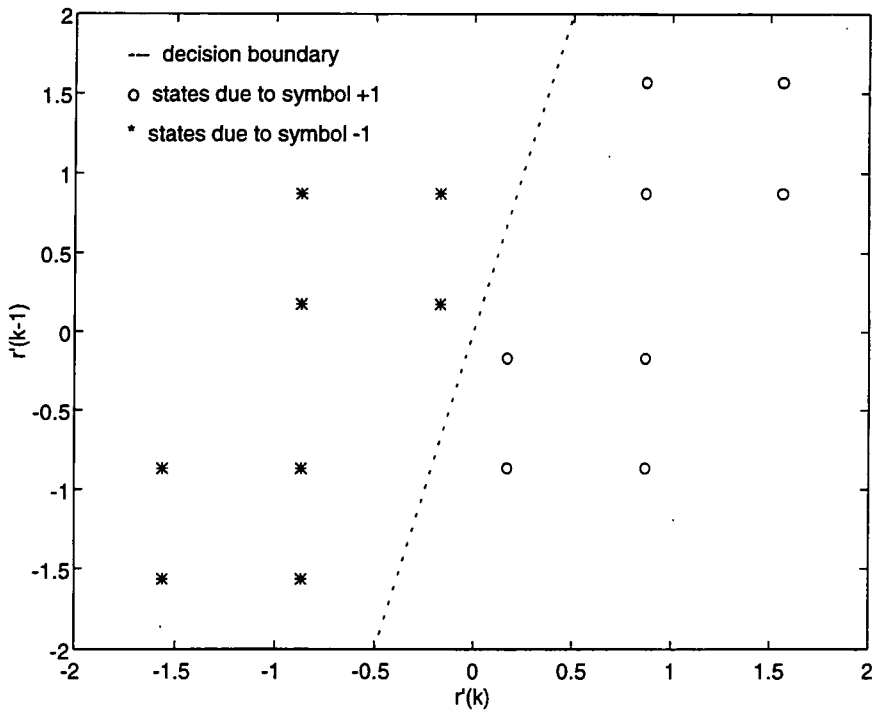

Figure 3.3: Plot of received signal states and linear decision boundary.

the delay line. The combination of m sequential received signal samples form m-dimensional received signal state vectors. This example demonstrates that more than one received sample is often required to separate the received states with a linear decision boundary. The decision delay can also help separate the states by altering the mapping of states to symbols. In the example, the received signal state, $r'(k)$, with no additive noise is given by:

$$r'(k) = \sum_{i=0}^{2} h_i(k)x(k-i) \tag{3.3}$$

The received signal state maps onto the symbol $x(k-1)$ rather than $x(k)$ — which would correspond to a decision delay = 0.

3.2 Adaptation Algorithms

In many communication systems, the channel characteristics change over time and thus the equaliser must be able to adapt to these changes for optimal results. This is particularly so in mobile communication systems where channels vary rapidly but also occurs in HF radio systems, systems employing frequency hopping and multi-link telephone systems. In order to adapt to changing channels, the equaliser must have some *a priori* knowledge of the signals being transmitted. Adaptation of the equaliser occurs by minimising an error function formed from characteristics of the received signals and the *a priori* knowledge of the transmitted signals.

Two broad classes of adaptive algotithms exist: algorithms that require a defined training signal to be transmitted so that the error function may be formed by comparing the received signal with the known transmitted signal, and algorithms that form an error function by comparing the statistics of the received signal with those of the transmitted signal. This latter technique is known as blind equalisation but is not considered in this thesis due to the lengthy training times and reliability of the measurements of the statistics. However, for fading communication channels where both the channel and transmitted symbols are unknown to varying degrees, the problem is essentially one of blind equalisation and some techniques discussed in later chapters have similarities with blind equalisation techniques [31]. The conventional adaptive algorithms may be applied to many of the different structures of equalisers or associated processing elements.

Another technique, called decision-directed adaptation, may be applied to equalisers employing the conventional training signal method of adaptation. After the training signal has been sent, the equaliser parameters are updated using the error between the equaliser and decision device outputs and attempts to track variations in the channel. This method makes the assumption that the classified symbols are correct and updates the parameters to more optimum values accordingly. The technique is suitable for relatively slow channel variations

compared to the symbol rate but begins to fail as the changes become more rapid. This is due to the increasingly incorrect assumption that the symbols have been classified correctly.

Common adaptive algorithms are the Least Mean Square (LMS) and Recursive Least Squares (RLS) algorithms which are designed to adjust the parameters of linear filters (e.g. FIR filters or linear equalisers). The basis of both these algorithms is to adjust the filter parameters in such a way to minimise an error function. The definition of different error functions leads to variants of a theme. Appendix B gives a mathematical overview of the complex signal versions of these algorithms — the real versions being easily obtained from the complex versions.

3.3 Conventional Symbol-by-symbol Equalisers

This section describes the two most common conventional symbol-by-symbol equalisers — the linear transversal equaliser (LTE) and the decision feedback equaliser (DFE), together with the application of common adaptation algorithms to adapt the structures to unknown or time-varying channels.

3.3.1 Linear Transversal Equaliser

The linear transversal equaliser (LTE) is the simplest equaliser structure and consists of an FIR filter, a decision device and adaptation algorithm (Fig. 3.4). Due to the sampled nature of this structure the analogue received signal, after passing through a whitened matched filter, is sampled prior to the FIR filter. The sampling may occur at either the symbol rate or at an integer multiple of the symbol rate. The latter, known as a fractionally spaced equaliser (FSE), must include a symbol-rate sampler between the FIR filter and decision device. The analogue version in which the filter is a tapped analogue delay line with sampling occuring between filter and decision device is called a zero-forcing equaliser.

The output of the symbol-rate sampled version is given by:

$$y(k) = \sum_{i=0}^{m-1} b_i(k) r(k-i) \tag{3.4}$$

where $b_i(k)$ are the filter coefficients and m is the order of the filter.

The LTE may be viewed as the inverse filter of the combined channel, transmit and receive filters. Usually this combination of filters is modelled as an FIR filter and thus the ideal inverse filter structure is an IIR. An FIR filter of finite length can only approximate the impulse response of this ideal inverse filter. The zero-forcing equaliser attempts to produce a combined impulse response which is zero at all but one of the sampling times and effectively

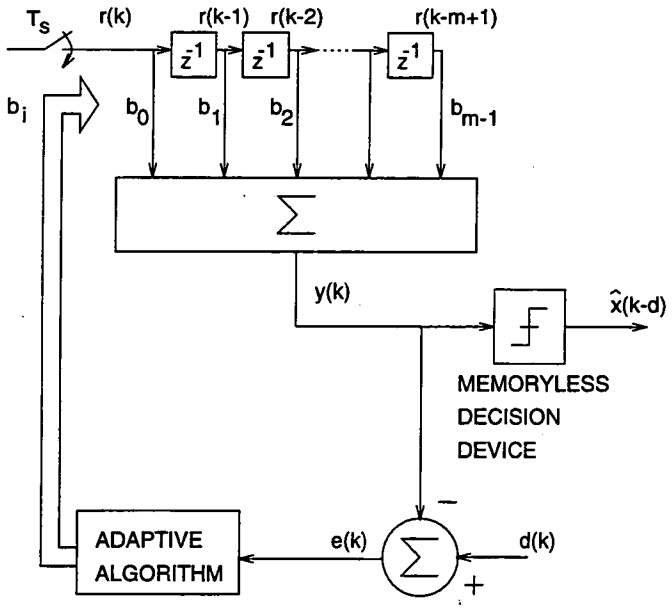

Figure 3.4: Linear transversal equaliser. The filter coefficients are shown as b_i but are, in general, time varying.

operates as an inverse filter only at these times.

The performance of the LTE is degraded when the channel has deep spectral nulls. The equaliser must provide a high gain at these frequencies but this increases the noise power and hence the probability of a symbol being incorrectly classified. If the channel is nonminimum phase (i.e. its z-transform has one or more zeros outside the unit circle) then the inverse causal filter is not stable. However a truncated form of the stable, noncausal inverse filter may be used by allowing a decision delay between receiving a symbol and classifying it. The delay permits a causal filter to be created from the noncausal inverse filter.

The FSE achieves a higher performance especially when operating in sytems utilising forms of quadrature amplitude modulation (QAM). Due to nonsymmetric channel frequency responses, the baseband response is, in general, nonsymmetric about the half symbol-sampling frequency and this may lead to large changes in the spectrum around the band edges (integer multiples of the symbol sampling frequency). The FSE can adjust the frequency response independently at the two band edges creating a flatter frequency response prior to symbol-rate sampling.

An alternative view of the mechanism by which the LTE aids the classification of symbols in the presence of noise is obtained by considering how the decision boundaries are formed with respect to the received signal states (i.e. received signal samples in the absence of noise). The LTE calculates a linear combination of a finite number of the received samples and classifies a symbol by comparing the magnitude of this scalar combination with a set of thresholds. For

a bipolar PAM scheme the decision device is simply a binary quantiser with a zero input-level threshold. The decision boundary for the case of no additive noise is given by:

$$b_0 r'(k) + b_1 r'(k-1) + \cdots + b_{m-1} r'(k-m+1) = 0 \tag{3.5}$$

where $r'(k-i)$ are the received states,

$$r'(k) = \sum_{i=0}^{L} h_i(k) x(k-i) \tag{3.6}$$

The decision boundary divides the received signal states into two subsets — each subset corresponding to the set of signal states caused by the transmission of a particular symbol. The tap weights are chosen to maximise the minimum distance between the two subsets of nominal received signal states and the decision boundary. Symbols are classified incorrectly when receiver / channel noise cause the received states to cross the decision boundary.

The adaptation algorithm usually used is either an LMS type or an RLS type used in conjunction with a training signal. The RLS algorithm is generally faster and produces a lower mean square error but is more computationally expensive. The LMS algorithm performs gradient descent of the mean square error surface,

$$J = E[|d(k) - y(k)|^2] \tag{3.7}$$

where $d(k) = x(k-d)$ — the transmitted symbol with a decision delay of d symbol periods, by using the instantaneous square error as an approximation to the mean. The algorithm does not perform a true gradient descent but converges in the mean to the minimum value of the error surface. The filter coefficients, $b_i(k)$, are updated after receiving each state of the training sequence according to the equation:

$$b_i(k+1) = b_i(k) + \mu r(k-i)^* e(k) \qquad i = 0, \ldots, m-1 \tag{3.8}$$

where μ is the stepsize, $r^*(k-i)$ are the conjugate of the received signal states and the error signal, $e(k)$, is given by:

$$\begin{aligned} e(k) &= d(k) - y(k) && (3.9) \\ &= x(k-d) - y(k). && (3.10) \end{aligned}$$

See Appendix B for further details on the adaptation process.

3.3.2 Decision Feedback Equaliser

The decision feedback equaliser (DFE) uses previously classified symbols to modify the decision boundaries formed by the equaliser. The DFE structure is shown in Figure 3.5 and consists of feedforward filtering of the sampled received signal, feedback filtering of classified symbols and a memoryless decision device. Variants using a fractionally spaced feedforward filter exist but the feedback filter always has a symbol-period spaced delay line. The combination of decision device and feedback filter forms a nonlinear IIR filter and creates a nonlinear equaliser structure. The IIR-type structure provides a more suitable inverse filter for the equalisation of channels modelled as FIR filters.

The output of the DFE (symbol-rate feedforward delay line variant) is given by:

$$y(k) = \sum_{i=0}^{m-1} b_i(k) r(k-i) + \sum_{j=1}^{n} a_j(k) \hat{x}(k-d-j) \tag{3.11}$$

where m is the order of the filter, $b_i(k)$ are the feedforward filter coefficients, n is the order of the feedback filter, $a_j(k)$ are the feedback filter coefficients and $\hat{x}(k-d-j)$ are the previously classified symbols with decision delay, d.

The method by which the DFE structure aids classification may be highlighted by considering how the equaliser achieves the inverse filtering requirement. In the absence of a feedforward filter, (i.e. $m = 1$), the feedback filter is required to cancel out all terms of the sampled channel impulse response other than the 0^{th} term (assuming zero channel propagation delay). Thus the ideal coefficients, $a_i(k)$, of the feedback filter are related to the symbol-rate sampled channel impulse response, $h_i(k)$, by:

$$a_i(k) = -h_i(k) \qquad i = 1, \ldots, n \tag{3.12}$$

For a sampled channel impulse response of length $(L+1)$, then $n = L$ for optimum cancellation. Under the condition of correct symbol classification and perfect knowledge of the CIR,

$$\begin{aligned} y(k) &= r(k) - \sum_{i=1}^{n} a_i(k) \hat{x}(k-i) && (3.13) \\ &= \sum_{i=0}^{L} h_i(k) x(k-i) - \sum_{i=1}^{L} h_i(k) x(k-i) + n(k) && (3.14) \\ &= h_0(k) x(k) + n(k) && (3.15) \end{aligned}$$

The addition of a feedforward filter allows increased flexibility in the choice of filter coefficients as the feedback filter must now cancel trailing terms of the convolved channel and feedforward filter impulse responses. The duration of the combined impulse response

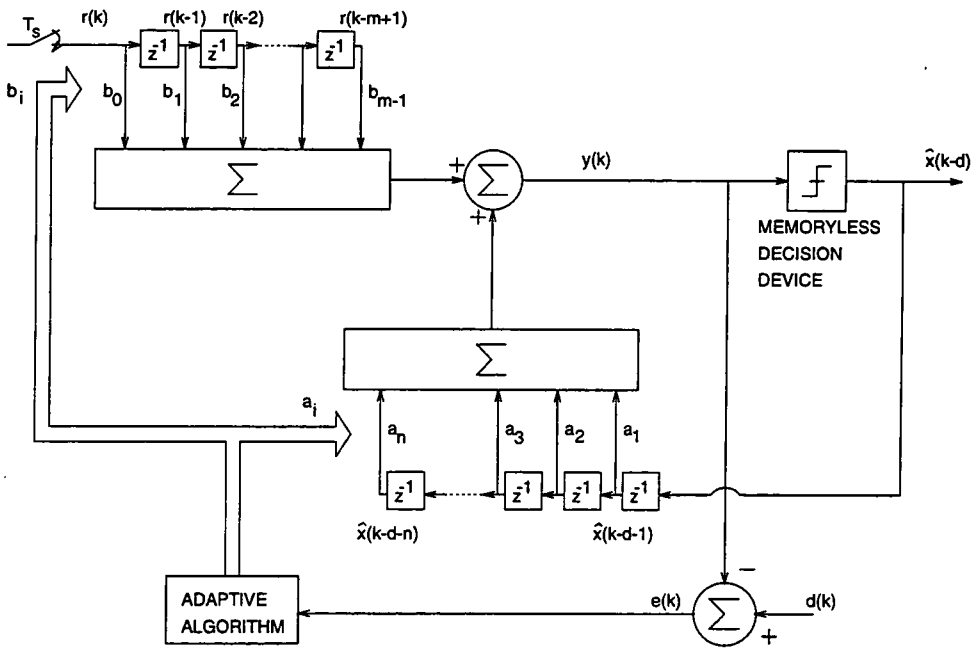

Figure 3.5: Decision feedback equaliser.

increases to $(L+m)$. Thus $n = (L+m-1)$ for cancellation of all ISI terms in the m received signal samples. By permitting a decision delay, the feedforward filter effectively cancels the leading terms of the sampled channel impulse response and the feedback filter cancels the trailing terms. A decision delay of d symbol periods reduces the order of the feedback filter to $n = (L+m-1-d)$.

Viewing the equalisation process as classification of the received signal samples, the feedback filter reduces the number of received signal states which the decision boundaries must separate. For a given set of n classified symbols in the feedback delay line (the feedback vector), the number of possible signal states that the next signal state can be is reduced by a factor, M^n, to

$$\frac{M^{L+m}}{M^n} = M^{d+1} \qquad (\text{ as } \; n = L+m-1-d \;) \tag{3.16}$$

for M-ary modulation schemes. This simplifies the decision boundaries and can increase the distance between the boundaries and the reduced set of nominal received signal states considerably. For any given feedback vector the decision boundaries are linear as they are formed by the linear feedforward filter coefficients. The nonlinearity of the equaliser is due to the variation of the position of the boundaries as the feedback vector changes state.

Adaptation of the DFE to an unknown channel uses standard gradient descent techniques as outlined in Section 3.2 and Appendix B. The feedforward filter coefficients b_i are adjusted

using the equation (LMS version):

$$b_i(k+1) = b_i(k) + \mu r^*(k-i)e(k) \qquad i = 0, \ldots, m-1. \tag{3.17}$$

The feedback filter coefficients, a_i, are adjusted by:

$$a_i(k+1) = a_i(k) + \mu x^*(k-d-i)e(k) \qquad i = 1, \ldots, n \tag{3.18}$$

where μ is the stepsize, $r(k-j)$ are the received signal samples, $x(k-d-i)$ are the training (transmitted) symbols, m is the order of the feedforward filter, n the order of the feedback filter and the error, $e(k)$, is given by

$$\begin{aligned} e(k) &= d(k) - y(k) && (3.19) \\ &= x(k-d) - y(k) && (3.20) \end{aligned}$$

where d is the decision delay. N.B. During training the actual transmitted symbols should be fed into the feedback delay line instead of the estimated symbols to aid training.

Decision-directed adaptation used to track time-varying channels replaces the training symbols (equal to the transmitted symbols) by the output of the decision device. This assumes that the detector is correctly classifying the received signal and thus the approximation

$$\widehat{x}(k-d) \approx x(k-d) \tag{3.21}$$

is valid.

3.4 Conventional Symbol-sequence Equalisers

This section explains the operation of symbol-sequence equalisers — primarily the optimum Maximum Likelihood Sequence Estimator, and variants. A hybrid version is also discussed which uses Bayes' theory to compute the most probable symbol or symbol sequence.

3.4.1 Maximum Likelihood Sequence Estimator

The maximum likelihood sequence estimator (MLSE) technique differs from the other conventional equalisation techniques by forming the best estimate of a sequence of symbols rather than on a symbol-by-symbol basis. Instead of performing an inverse filtering function, the MLSE calculates the parameters for a model of the channel and uses them to estimate the received signal states. These received signal states are required by a Viterbi detector in order to predict the most probable sequence of transmitted symbols given a sequence of received

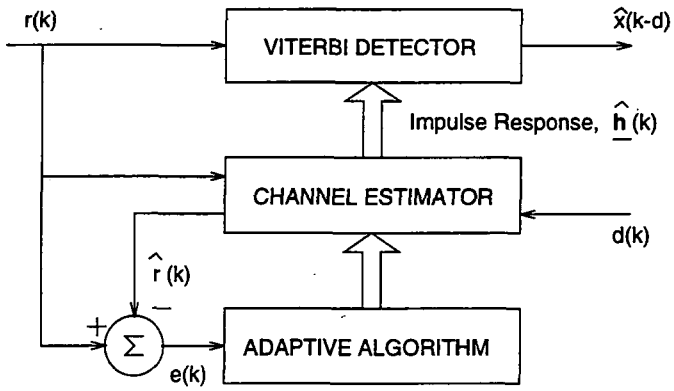

Figure 3.6: Maximum likelihood sequence estimator.

signal samples. The Viterbi detector is a recursive processor which calculates the maximum likelihood estimate efficiently [32].

The general structure of the MLSE is shown in Figure 3.6. The whitened matched filter required *prior* to symbol rate sampling is not shown. Strictly this matched filter should be altered to account for channel variations but practical results have shown that failing to do so results in only a slight degradation in equaliser performance.

The channel estimator typically consists of an adaptive FIR filter, sampled at the symbol rate, and an associated adaptation algorithm. Adaptation algorithms requiring a conventional training sequence are the most common and typically involve an LMS gradient descent of the error surface. The training signal, $d(k)$ is fed through the channel estimator to produce an estimate of the received signal, $\widehat{r}(k)$. The error signal, $e(k)$, is formed from the estimated and actual received signal

$$e(k) = r(k) - \widehat{r}(k) \tag{3.22}$$

and is used to update the estimate of the sampled channel impulse response (CIR). The training signal $d(k)$ is equal to the transmitted symbols, $x(k)$, during the training period or, in the case of decision-directed tracking, the output of the Viterbi detector $\widehat{x}(k-d)$ may be used to track the channel. This requires some slight modifications which are detailed in Chapter 5.

Once channel estimation has taken place, the estimated CIR, $\underline{\widehat{\mathbf{h}}}(k)$, is transferred to the Viterbi detector which calculates estimates of the received signal states,

$$\widehat{r'}_i(k) \qquad i = 0, \ldots, (M^{L+1} - 1). \tag{3.23}$$

The number of received signal states is M^{L+1} for a channel with a symbol-period sampled CIR of $(L+1)$ components and an M-ary modulation scheme — corresponding to all the

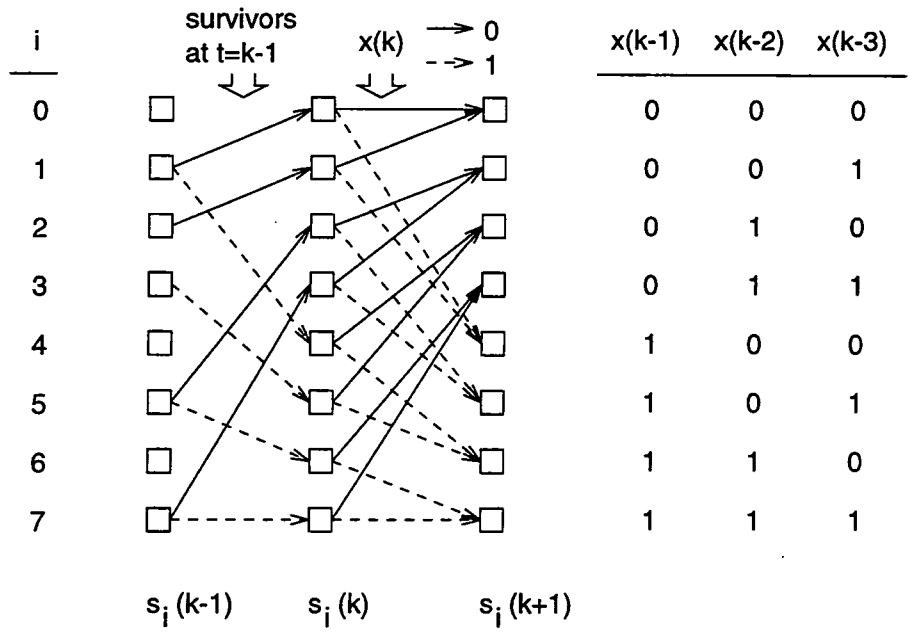

Figure 3.7: Viterbi trellis example. The modulation scheme is binary, represented by the symbols {0,1}.

possible combinations of the transmitted symbol sequence,

$$[x(k), x(k-1), \ldots, x(k-L)]$$

which forms the received signal state (i.e. the received signal in the absence of noise). The Viterbi algorithm can be viewed as a trellis search algorithm where the states in the trellis, $s_i(k)$, correspond to distinct combinations of the transmitted symbol subsequence,

$$[x(k-1), \ldots, x(k-L)]$$

and the transition between states is governed by the transmitted symbol, $x(k)$. Clearly, there are M transitions to or from each state in the trellis, one for each of the M values $x(k)$ can take (Fig. 3.7). The mapping from state transitions to symbol sequences is one-to-one [33]:

$$s_i(k) \rightarrow [x_i(k-1), \ldots, x_i(k-L)]$$

Whilst operating on the received signal samples, the Viterbi detector calculates the value of the log-likelihood ratio for each valid transition between the trellis states using a knowledge of the probability density function of the additive noise process, $p_n(\cdot)$, the probability distribution function of the transmitted symbols, $P(\cdot)$, the received signal, $r(k)$ and the estimated received signal states, $\widehat{r'}_i(k)$. The log-likelihood ratio, λ, is known as the transition metric

and is given by:

$$\lambda(\xi_{ij}(k)) = -\ln(\ P(s_j(k+1)|s_i(k))\ ) - \ln(\ p_n(r(k)|\xi_{ij}(k))\ ) \tag{3.24}$$

where $\xi_{ij}(k)$ is the transition from state $s_i(k)$ to state $s_j(k+1)$ at time $t = k$. For a trellis of M^L states there are a total of M^{L+1} transitions between one stage of the trellis and the next.

If the noise process is Gaussian then

$$\ln(\ p_n(r(k)|\xi_{ij}(k))\ ) \propto |r(k) - \widehat{r'}_i(k)|^2 \tag{3.25}$$

The probability distribution of the transmitted symbols, $P(\cdot)$, is usually constant and may be omitted in order to simplify computational load.

Each trellis state, $s_j(k+1)$, may be reached from M previous states and so, for each trellis state, the Viterbi detector selects the transition to the state with the minimum total path length. These transitions are known as the survivors. For each transition the total path length is formed by accumulating the values of the log-likelihood ratios of all prior transitions leading to the previous survivor and the value for the particular transition (the transition metric). The total path length is related to the probability of receiving that particular symbol sequence or path.

Strictly, the symbol sequence should start and end with known subsequences in order to initialise and terminate the algorithm — the transmitted sequence starts and ends at known trellis states. However, this is often impractical in many communication applications where there is not a frame structure to the transmitted data as the overhead of transmitting 'control' symbol sequences is unacceptable. An alternative is to use a fixed decision delay, d, whereby, at each sample period, the algorithm selects the survivor with the minimum total path length and traces the survivor's path back an appropriate, fixed number of states and outputs the corresponding symbol estimate, $\widehat{x}(k-d)$. The magnitude of this decision delay affects the performance of the equaliser and is examined for the case of fading channels in Chapter 5.

Even without known start and end symbol subsequences the equaliser performance is better than the LTE or DFE structures with similar decision delays. The variation in survivor path lengths is such that survivors starting from erroneous states are soon eliminated given sufficient decision delay. As the SNR decreases the variations in path metrics become smaller and an erroneous sequence estimate becomes increasingly likely. The computational complexity of the algorithm is significantly greater than the LTE or DFE.

Given a data sequence starting and ending with known symbol subsequences then the MLSE is an optimum sequence detector. However, whilst operating on nonstationary or fading channels, the tracking capabilities of the channel estimator become a limiting factor.

These tracking errors maybe reduced by using decision-directed adaptation of the channel estimator. The estimated symbol sequence used as the training signal can either be taken from the main output of the Viterbi detector or from a lower decision delay output. This latter case can improve the symbol error rate for some time-varying channels where the advantage in delaying the sequence decision to improve the decision accuracy is offset by poor channel estimation which results in the Viterbi detector using erroneous values of the received signal states, $r_i'(k)$, in the sequence estimation.

3.4.2 Reduced State Sequence Equaliser

When the memory of the channel, L, increases the number of states in each stage of the Viterbi trellis becomes large and the MLSE equaliser has a huge computational burden. For high symbol rate communication systems, the practical implementation is not feasable for economic or processor capability reasons. Reduced state techniques reduce the number of states in the trellis at the expense of bit error rate performance [34, 35].

The Delayed Decision-Feedback Sequence Estimator (DDFSE) reduces the number of states in each stage of the trellis by only providing a separate, reduced state for each combination of a subsequence of the transmitted symbol sequence. The conventional MLSE's states, $s_i(k)$ correspond to a distinct combination of the transmitted symbol sequence,

$$[x(k-1), \ldots, x(k-L)]$$

whereas the DDFSE's reduced states, $\tilde{s}_i(k)$, correspond to distinct combinations of

$$[x(k-1), \ldots, x(k-u)]$$

where $u \leq L$. Thus the number of reduced states is M^u for an M-ary modulation scheme compared to M^L for the conventional MLSE.

In order to provide information on the full symbol sequence, each reduced state has an associated 'past decision' sequence, $\underline{\widehat{x}}_i(k)$,

$$\underline{\widehat{x}}_i(k) = [\widehat{x}_i(k-u-1), \ldots, \widehat{x}_i(k-L)]. \tag{3.26}$$

Note that each reduced state can have completely different 'past decision' sequences — indicated by the subscript i ($i = 0, \ldots, M^u - 1$). Figure 3.8 shows an example of a reduced state trellis.

The DDFSE calculates the received signal states and the transition metrics for each of the M transitions from each reduced state, $\tilde{s}_i(k)$, using the relevant reduced state's symbol subsequence and 'past decsion' sequence, $\underline{\widehat{x}}_i(k)$ and the channel estimate, $\underline{\widehat{\mathbf{h}}}(k)$. As in the

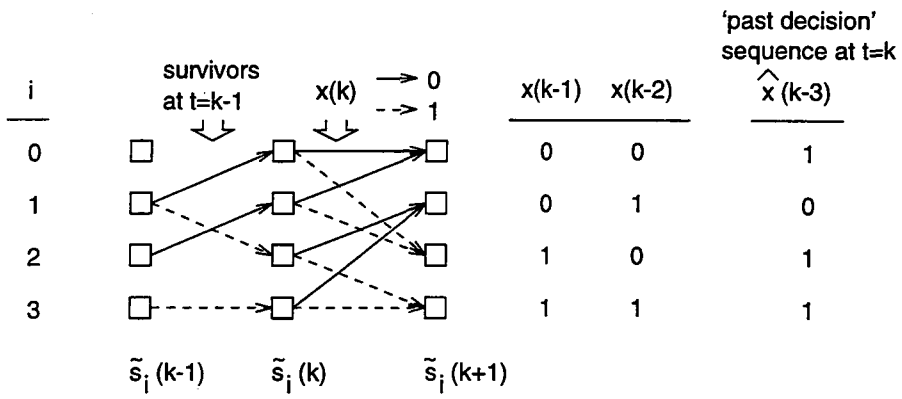

Figure 3.8: DDFSE trellis example. Here $M = 2$, $L = 3$ and $u = 2$ giving a trellis of 4 states and a single element 'past decision' sequence.

conventional MLSE, the total path lengths of the M transitions arriving at each reduced state, $\tilde{s}_j(k+1)$, are compared and the most probable is selected as the survivor to that destination reduced state.

The destination reduced state's 'past decision' sequence, $\underline{\widehat{x}}_j(k+1)$, is formed from the 'past decision' sequence, $\underline{\widehat{x}}_i(k)$, and symbol sequence of the survivor's departing reduced state, $\tilde{s}_i(k)$. The 'past decision' sequence acts like a shift register of previous decisions and the oldest symbol in the 'past decision' sequence is shifted out and the oldest symbol in the sequence corresponding to the departing state becomes the newest symbol in the destination state's 'past decision' sequence. If, at time t=k, departing state $\tilde{s}_i(k)$ has a symbol sequence and 'past decision' sequence:

$$[x_i(k-1),\ldots,x_i(k-u)] \quad , \quad [\widehat{x}_i(k-u-1),\ldots,\widehat{x}_i(k-L)]$$

then the destination state, $\tilde{s}_j(k+1)$, will have the corresponding sequences:

$$[x_j(k),x_j(k-1),\ldots,x_j(k-u+1)] \quad , \quad [x_i(k-u),\widehat{x}_i(k-u-1),\ldots,\widehat{x}_i(k-L+1)]$$

N.B. The symbols $\{x_j(k-1),\ldots,x_j(k-u+1)\}$ are exactly the same as the symbols $\{x_i(k-1),\ldots,x_i(k-u+1)\}$.

For example, if a departing state's symbol sequence is [010] and 'past decision' sequence [1011], then a transition to a destination state [x01] would result in state [x01]'s 'past decision' sequence to be [0101] at the next stage of the trellis.

The remainder of the algorithm is the same as the MLSE algorithm in regard to producing a symbol estimate, $\widehat{x}(k-d)$, with a fixed decision delay and the use of these estimates to aid channel tracking.

3.4.3 Block Decision Feedback Equaliser

The block decision feedback equaliser (Block DFE) can act as either a symbol-by-symbol decision equaliser or as a symbol sequence estimator. Indeed the structure may be expanded to give a performance equal to that of the MLSE [36]. The structure uses a nonlinear decision device that implements a maximum *a posteriori* (MAP) detector [37] instead of an M-level quantiser. This relatively new hybrid equaliser is discussed as it has significant similarities with the Radial Basis Function (RBF) equaliser which also uses Bayes' theory to develop a MAP detecor and employs decision feedback to aid classification (see Section 3.5.2 for details of the RBF / Bayesian equaliser).

Figure 3.9 shows the structure of the Block DFE which uses two received signal samples to estimate a sequence of two transmitted symbols in this example. As the number of received signal samples and symbol decision sequence length increases, the Block DFE performance approaches that of the MLSE. The structure does not employ a front-end FIR filter but does require a received sample delay line. The decision device gives p symbol estimates which are derived from m received signal samples. These m samples should contain energy from at least one of the p symbols to be estimated. However, in addition, they contain ISI terms from other symbols and these may be removed by utilising previous symbol estimates (assuming correct decisions) and a knowledge of the sampled CIR as per the DFE. Each received signal sample is affected by $(L+1)$ symbols with the received signal sample given by:

$$r(k) = \sum_{i=0}^{L} h_i x(k-i) + n(k) \tag{3.27}$$

where L is the channel memory and h_i are the coefficients of the sampled combined CIR. The 'cancelled' received signal samples, $y^j(k)$ are given by:

$$y^j(k) = \sum_{i=0}^{p-1-j} h_i x(k-j-i) + n(k-j) \tag{3.28}$$

where $j = 0, \ldots, (m-1)$ and $p \leq L$ and assuming that correct past symbol decisions have been made. Thus the inputs to the decision device, $y^j(k)$, are the received signal samples with all known past symbol interference effects removed from them. This simplifies the MAP decision procedure.

For the symbol-decision case, a single symbol decision, $\widehat{x}(k)$, is made and is fed back into all delay lines which are modified slightly to provide correct cancellation of previous symbol interference effects.

The MAP detector is derived using a knowledge of the probability density function of the additive receiver / channel noise. For each possible symbol or symbol sequence, the detector calculates the conditional probability of receiving that sequence given the set of received signal

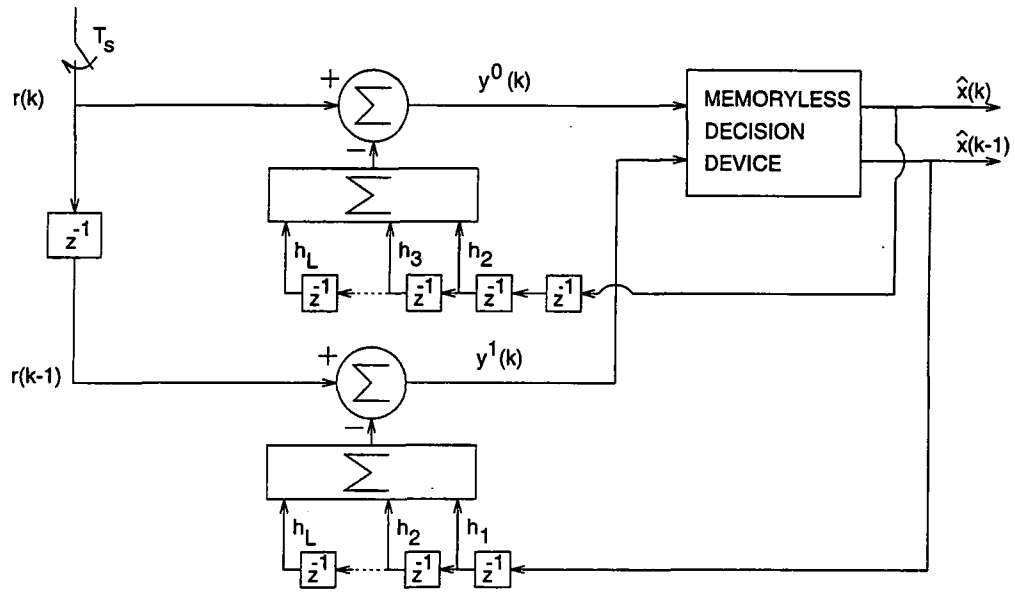

Figure 3.9: Block decision feedback equaliser example. Here $m = 2$, $p = 2$ and the detector operates without decision delay.

states and a set of correct past decisions. The symbol/sequence with the highest conditional probability is selected for the symbol / symbol sequence decision. A decision delay, d, may be introduced into the MAP detector but the coefficients of the feedback delay lines must be modified to take this into account.

The description given assumes a time-invariant channel but the coefficients of the feedback filter may easily be adjusted to take into account time variations.

Although [36] does not consider an adaptive version of the Block DFE, the structure is suitable for adaptation using standard gradient descent techniques. However, if the structure employed is large, the number of feedback filter coefficients that need to be adapted may lead to large processing requirements. An alternative is to use a channel estimator as in the MLSE and use these coefficients in the feedback filters whose coefficients are directly related to the sampled channel impulse response.

3.5 Neural Network Equalisers

The DFE and MLSE improve the estimate of a symbol or sequence of symbols by employing various nonlinearities in the classification process; the DFE is a nonlinear IIR filter and the MLSE algorithm is inherently nonlinear in its metric formulation and decision process. Neural networks have been applied to equalisation problems in an attempt to enhance classification performance by utilising the nonlinearities inherent in the networks. In addition,

the performance gap between DFE and MLSE techniques is significant but the MLSE has a much larger computational burden. Neural networks may provide a compromise between performance and computational complexity.

At any given decision time both the LTE and DFE form linear decision boundaries to separate the received signal state space and all states appearing in a given region are classified as a particular symbol. However, the division of the state space by linear boundaries is, in general, not optimal particularly when the boundaries separate many received signal states. The optimal decision boundaries for such symbol-decision equalisers are nonlinear and may be derived by the application of Bayes' decision theory which gives the maximum *a posteriori* (MAP) detector [37]. The nonlinearities of the neural networks can generate nonlinear decision boundaries and be used to equalise ISI and other channel / receiver nonlinearities.

Neural networks have proven classification capabilities and have been used successfully in pattern recognition problems [38]. Many different network structures exist with variations in the topology of connections between the elemental nodes of the network and the nonlinearities associated with each node. This section considers 3 structures of networks which have been used to implement symbol-by-symbol decision equaliser functions: the multilayer perceptron (MLP) network, the radial basis function (RBF) network and the recurrent neural network (RNN).

3.5.1 Multilayer Perceptron Equaliser

A multilayer perceptron network consists of a number of nodes arranged in layers with the output of each node in a layer being connected to the inputs of all nodes in the following layer [39]. There are no connections between nodes in the same layer or to nodes in previous layers and thus this is a strictly feedforward network (Fig. 3.10). Each node calculates the sum of its weighted inputs (the activation) together with a constant value (the threshold) and outputs a nonlinear function of the activation. The nonlinear function is usually a sigmoid, hyperbolic tangent function or hard-limiter. The node output, $y_i(k)$ may be expressed as:

$$y_i(k) = f(\sum_{j=0}^{m-1} w_{ij}(k)y_j(k) + t_i) \tag{3.29}$$

where $f(\cdot)$ is the nonlinear function, $y_j(k)$ are the node inputs, $w_{ij}(k)$ are the input weights and t_i is the threshold. The summation term represents the node activation.

The inputs to the equalise form the input layer and consist of delayed samples of the received signal and, in the case of a decision feedback scheme, previously classified symbols [40, 41]. The received signal is nominally sampled at the symbol rate, $1/T_s$. Each received sample of the input layer is connected to all nodes in the next layer, termed the hidden layer. The input layer is anomalous in that it only provides connections to the network and does

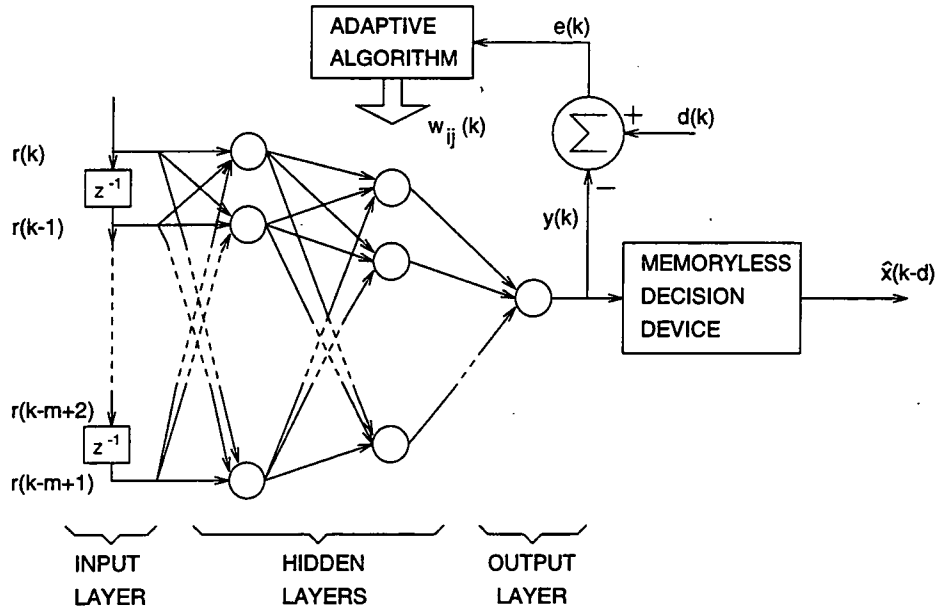

Figure 3.10: Structure of the feedforward Multilayer Perceptron equaliser.

not incorporate any of the nonlinear nodes. The layered network is sequential rather than synchronous in the sense that, for a given input, all first layer nodes calculate their outputs which the second layer nodes use as inputs, i.e. the nodes do not incorporate symbol-rate delay elements. The outputs of the final layer nodes are classified by a decision device which outputs one of the M symbols for an M-ary modulation scheme.

The MLP equalises the channel by classification rather than inverse filtering. The received signal state space is divided into regions by decision boundaries with each region being associated with a transmitted symbol. The MLP forms the decision boundaries and transforms the received signal samples into values which can be classified by the memoryless decision device.

In order to form curved decision boundaries, the network must have at least one hidden layer. As the input layer is intrinsically linear, the output layer will only form nonlinear decision boundaries from the inputs if they have been processed by an intermediate layer [42]. Without a hidden layer and in the case of symbol feedback, the decision boundaries will be a nonlinear function of previous received signal samples but still form linear decision boundaries to separate the current received signal samples. This situation is analogous to the nonlinear effects on the decision boundaries produced by the conventional DFE. As the number of hidden layers and nodes increase, complex decision boundaries can be formed including boundaries isolating distinct regions of the state space.

Adaptation (or training) of the network is achieved by adjusting the weights, $w_{ij}(k)$, of the network using the back-propagation algorithm [39]. The threshold levels are normally

adjusted by implementing the threshold as the output of a node with a single, constant input and incorporating these additional nodes in the back-propagation algorithm. The algorithm requires a conventional training signal to be transmitted and an error function is formed from the nodes in the output layer of the equaliser and the desired output(s). The technique performs a gradient descent of the instantaneous square error surface as in the LMS algorithm. The major difference between the two algorithms lies in the inclusion of additional calculations in order to form error gradients for the hidden layer nodes for which no explicit desired output exists. The back-propagation algorithm requires that the node transfer function be differentiable and thus cannot be applied to networks with hard-limiting / quantising node nonlinearities.

In general, the MLP suffers from the problem of a multimodal error surface due to the node nonlinearities. This can cause the back-propagation algorithm to converge to local minima of the error surface producing a suboptimal equaliser. A number of techniques for reducing the probability of the algorithm becoming trapped in local minima exist including 'momentum' terms in the weight update equations and the use of adaptive learning rates. However, these methods do not guarantee convergence to the global minimum and their effectiveness is application dependent.

As the channel to be equalised becomes more complex, the decision boundaries required become increasingly nonlinear and require larger numbers of nodes and layers to achieve sufficient performance. The adaptation of the network takes considerably longer due to the increasing complexity of the error surface and the computational overhead of the trained network increases substantially. The relatively long training times tend to make MLP equalisers unsuitable for channels with rapid fading.

3.5.2 Radial Basis Function Equaliser

The RBF network has a simpler structure than the MLP network, consisting of a single hidden layer of nodes and a final layer implementing a set of M linear combiners (Fig. 3.11) [39]. As with the MLP, the network is a classifier rather than an inverse filter.

The input layer is similar to that of the MLP, and forms an input vector from a sequence of delayed samples of the received signal (the received signal vector) and, in the case of a decision feedback structure, previously classified symbols. The received signal is nominally sampled at the symbol rate, $1/T_s$. There are no connections between nodes of the hidden layer resulting in a feedforward network. The nodes operate in a distinctly different manner than the nodes in an MLP network. Each node has an associated 'centre' — a vector equal to the received signal state vector that the node 'recognises', and the output of the nodes are Gaussian functions of the Euclidean distance between the current received signal vector and

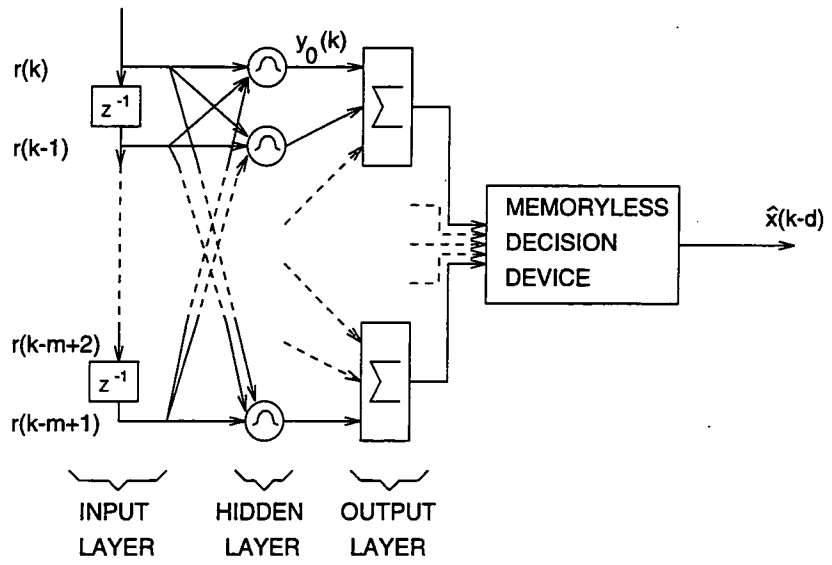

Figure 3.11: Structure of the Radial Basis Function equaliser.

the node centre. Thus a node output, $y_j(k)$ is given by:

$$y_j(k) = \alpha \exp\left(\frac{-\|\underline{\mathbf{r}}(k) - \underline{\mathbf{r}}'_j(k)\|^2}{\rho}\right) \tag{3.30}$$

where

$$\underline{\mathbf{r}}(k) = [r(k), r(k-1), \ldots, r(k-m+1)]$$

is the current received signal vector of dimension m,

$$\underline{\mathbf{r}}'_j(k) = [r'_i(k), r'_i(k-1), \ldots, r'_i(k-m+1)]$$

is the node centre, ρ is the 'width' of the Gaussian function and α is a constant.

In the absence of decision feedback, each node has a centre of dimension equal to the number of received signal samples forming the received signal vector. The number of nodes is equal to the number of received signal state vectors — given by $M^{(L+m)}$ for an M-ary modulation scheme, where L is the channel memory.

The nodes may be divided into subsets whose corresponding received signal state vectors (centres) are classified as one of the M symbols. The output layer forms M separate sums for all nodes in each of the M subsets and passes these to a decision device which selects the symbol corresponding to the maximum sum as its output. Each of the M sums represent how probable a given received signal vector is to the set of received signal state vectors classified as a particular symbol.

In Chen et al [43] decision feedback is used not to form additional components of the input

vector but to allow the number of nodes in each subset to be reduced when calculating the Euclidean distances. For a sequence of n previous symbols that have been correctly classified, the sequential nature of the FIR channel model implies that the current received signal vector can only be one of a reduced number of received signal state vectors. The reduction factor, M^n, is the same as that acheived with a conventional DFE as given in Section 3.3.2.

The RBF network implements a symbol-by-symbol decision Bayesian equaliser for channels corrupted by Gaussian noise [43]. The Bayesian equaliser requires the calculation of the conditional probability of choosing a particular symbol given a sequence of received signal samples and assuming a correct sequence of previously classified symbols. If the M symbols are transmitted with equal probability then the expression for the conditional probability reduces to a function of the form of the noise probability density function. Given that the channel noise distribution is Gaussian, the RBF network calculates a value proportional to this conditional probability. The decision device in the symbol-decision Bayesian equaliser selects the symbol for which the conditional probability is largest. Hence this implementation of RBF equalisers are also termed Bayesian equalisers.

Adaptation of the RBF network to unknown channels follow variations of one of 2 methods, both of which require a training signal to be transmitted. The first method uses a channel estimator as in the MLSE equaliser. Estimates of the received signal states and vectors are calculated using the channel model and a node centre is set to each of these vectors. Nodes are assigned to the estimated received signal state vectors according to which symbol the vectors should be classified. All vectors corresponding to a particular transmitted symbol are assigned to nodes in the same symbol-decision subset which eliminates the need to train the output layer of linear combiners.

The second method uses a clustering algorithm, such as the κ-means algorithm, in parallel with an LMS or RLS algorithm to adapt the weights of the linear combiners. This method requires a training signal only for the LMS / RLS half of the algorithm. The unsupervised κ-means algorithm initialises the nodes' centres to small random values and, for each received signal vector, the node with the closest centre is moved a fraction of the distance closer to the received signal vector. This leads to the nodes becoming centred on the received signal state vectors. The LMS / RLS algorithm adjusts the weights of the linear combiners so that they sum the outputs of the nodes of the same symbol-decision subsets. The error signal is formed using the known training signal and the outputs of the M linear combiners. A combiner has its weights updated if it corresponds to the training symbol transmitted.

The first adaptation method is model based and thus limited to the type of channel model. For many channels the type and length of impulse response is known to an adequate degree and thus channel modelling does not pose a severe limitation. The second adaptation method is model free and thus can adapt to a larger number of channel types given a sufficient number of nodes. Note that the output layer structure for this method is slightly different in that

each linear combiner forms a weighted sum of the inputs from all nodes due to the fact that the symbol associated with each node is not known at the start of the clustering algorithm.

The 'width', ρ, of the nodes' Gaussian transfer function are set to an estimate of the square root of the receiver noise power. The network is relatively insensitive to this parameter and so errors in the noise power estimation do not affect the equaliser performance significantly.

Further details of the RBF / Bayesian equaliser may be found in Chapter 6 which examines the similarities between Bayesian and MLSE equalisation techniques.

3.5.3 Recurrent Neural Network Equaliser

The RNN is the most general form of neural network with each node being connected to all other nodes in the network. Figure 3.12 shows the general structure of a discrete time RNN. Each node outputs a nonlinear function of the sum of its weighted inputs, as in the MLP network — the nonlinear function typically being a sigmoid, hyperbolic tangent or hard-limiter. The fully interconnected structure of the network means that, in general, there are no layers apart from the input layer. A layered structure may be obtained if some of the feedback paths have zero weighting. The inputs to the network are delayed samples of the received signal; the signal being sampled at the symbol rate, $1/T_s$. The network passes one of the nodes' output to a decision device which forms an estimate of the transmitted symbol, $\widehat{x}(k-d)$, with a decision delay, $d \geq 0$.

The feedback in the network creates a nonlinear dynamical system and the network may exhibit various oscillatory or convergent behaviours. The dynamics of a node in the *continuous time* network may be described by [39]:

$$\frac{dy_i(t)}{dt} = -y_i(t) + f(s_i(t)) \tag{3.31}$$

where $y_i(t)$ is the node output, $s_i(t)$ is the node activation (the sum of the weighted node inputs) and $f(\cdot)$ is the nonlinear function. For simulation purposes time must be discretised requiring an approximation of the derivative. This leads to discrete time node dynamics defined by:

$$y_i(k\tau) = f(s_i(k\tau)) \qquad k = 0, 1, \ldots \tag{3.32}$$

where τ is the time step. The RNNs dealt with in this thesis are only concerned with the discrete time case and τ is set to the symbol period, T_s, unless stated otherwise. With this assumption, τ may be removed from the notation.

The discretisation of the continuous time RNN nodes introduces a single delay element in the feedback paths of the discrete time RNN as shown in Figure 3.12. Some definitions of the node dynamics introduce a delay element to all node inputs so that the node outputs

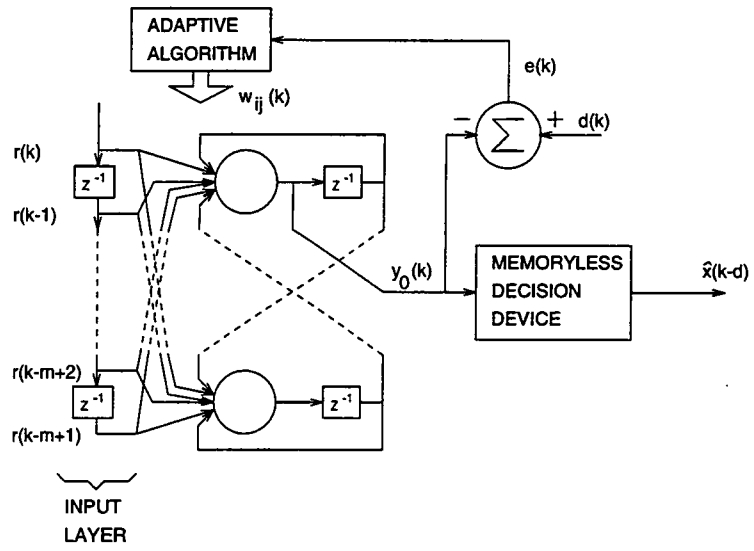

Figure 3.12: Structure of the Recurrent Neural Network equaliser.

$y_i(k)$ for an activation of $s_i(k-1)$. The dynamics defined by Eqn. 3.32 are used so that the nodes have zero delay. Thus the activation of node i is given by:

$$s_i(k) = \sum_{j=0}^{m-1} w_{ij}(k) r(k-j) + \sum_{j=0}^{n-1} w_{i,j+m}(k) y_j(k-1) \qquad i = 0, \ldots, (n-1) \tag{3.33}$$

where m is the number of input received signal samples, n is the number of nodes and $w_{ij}(k)$ are the node weights.

One method of adapting the RNN weights is obtained by the real-time recurrent learning (RTRL) algorithm [44] which requires a conventional training sequence. The RTRL performs a gradient descent of the instantaneous square error surface — the error signal being formed between the output of the equaliser, $y^0(k)$, and the desired output, $d(k)$. As with the MLP network the nodes must have a differentiable nonlinearity for gradient descent adaptation methods. Similarily local minima may exist due to the network nonlinearities and the algorithm cannot be guaranteed to converge to the global minimum.

The RTRL algorithm updates the weights, $w_{ij}(k)$, at each sample instant rather than applying a set of m recieved signal samples on the network inputs and adjusting the weights until a minimum of the error surface is reached before presenting a new set of received signal samples to the network input. This second method is known as the back-propagation-through-time algorithm and is an extension of the MLP back-propagation algorithm [39].

Kechriotis et al. [45] have shown that an RNN can perform equalisation of an FIR channel model with small memory using a small number of nodes, typically two or three. This is significantly smaller than the number of nodes used in a feedforward MLP structure

[40] which results in far lower computational requirements. The methods by which RNNs achieve equalisation are not fully understood and the precise design criteria for the number and type of nodes and the weight settings in relation to the channel model parameters were not examined.

3.6 Summary

This chapter has given an overview of the mechanisms by which equalisers compensate for intersymbol interference introduced by the channel. A number of conventional equaliser structures have been described, both symbol-by-symbol and symbol sequence based.

These conventional techniques have a wide variation in performance and computational complexity with the LTE having low performance and complexity and the MLSE having high performance and complexity. Whilst operating on fading channels, MLSE techniques perform less than ideally (see Chapter 5) and neural network based equalisers may prove to be more robust than the MLSE on fading channels or provide a compromise between the performance of the MLSE and the DFE or DDFSE.

The following chapter examines RNN equalisers in detail with both a theoretical treatment of the mechanisms used to achieve equalisation and simulation of their relative performance compared to other equalisation techniques.

Chapter 4

Recurrent Neural Network Equalisers

The work reported on recurrent neural network (RNN) equalisers in [45] considered only the performance aspects of the structures and compared them against the conventional LTE. This chapter tests the validity of these results, compares the performance against the DFE and RBF structures and examines the mechanisms through which RNNs achieve equalisation.

4.1 Initial Simulations

Initial simulations were performed to confirm some of the results obtained in [45] and, in addition, to compare the RNN equaliser's symbol error rate (SER) against LTE, DFE and RBF equaliser structures. The channel model

$$H_1(z) = 1 + 0.7z^{-1}$$

was used to assess the performance of an RNN equaliser against an LTE and a 2^{nd} model,

$$H_2(z) = -0.2052 - 0.5131z^{-1} + 0.7183z^{-2} + 0.3695z^{-3} + 0.2052z^{-4},$$

for comparison with the RBF equaliser and DFE. Both channels are time invariant / non-fading but do introduce ISI and thus are aimed at testing the equalising capabilities of the RNN under less severe conditions.

The radio system simulated had no channel coding / decoding and the modulator converted random data bits to binary PAM symbols, $x(k) \in \{-1, +1\}$, without any pulse shaping (Fig. 4.1). The radio channel was simulated with one sample/symbol resulting in relative path delays equal to an integer number of symbol periods. All signals are real valued for simplicity

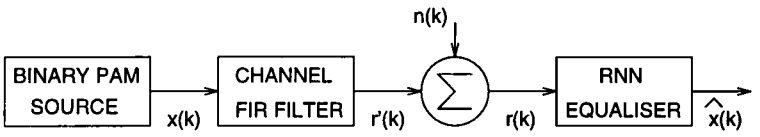

Figure 4.1: Radio system model for initial RNN simulations.

of the initial simulations.

The first channel's power gain is unnormalised (i.e. the power gain $\neq 1$) so that direct comparisons with results obtained by Kechriotis *et al* [45] may be made. The signal power is increased by such channels and so the receiver additive noise power must be increased to a obtain a consistent SNR. The SNR is defined as:

$$\text{SNR} = \frac{G_c \sigma_s^2}{\sigma_n^2} \tag{4.1}$$

where G_c is the power gain of the channel, σ_s^2 is the transmitted signal power and σ_n^2 is the noise power.

The RNN used to equalise channel $H_1(z)$ consisted of a single input node (i.e. a single received signal sample delay line), two nodes with hyperbolic tangent nonlinearities and a zero-crossing threshold detector operating with zero decision delay. This is the same structure as used by Kechriotis *et al* [45]. The LTE was a 5^{th} order filter and with a zero decision delay detector. Symbol error rates and mean square error performance were calculated using an ensemble of 100 or 1000 trials depending upon the error rate.

Each trial was composed of a training sequence of 1000 symbols followed by 1000 symbols for which the number of classified symbols in error was counted. The error count is accumulated over the ensembles to reduce statistical variations especially at low error rates. The SER is defined as the number of symbol errors divided by the total number of symbols transmitted during the measurement period. The RTRL training algorithm [44] was used to train the RNN and an LMS algorithm for the LTE during the initial training sequence of each trial. Whilst data was being transmitted, the equalisers' coefficients were held constant — i.e. no decision directed adaptation was used. The training rates / stepsizes were adjusted so that both algorithms adapted the equalisers reliably and a comparitively large number of training symbols ensured that the equalisers adapted to a 'best' solution prior to error rate calculation. An ensemble of trials was used in order to average out variations in equaliser coefficients between trials. This method was adopted as the RNN equaliser had been observed to train unreliably during some training sequences.

Figure 4.2 shows the ensembled-averaged (mean) square error for the two node RNN and 5^{th} order LTE for channel $H_1(z)$ with an SNR = 15dB and ensemble of 100 trials. The

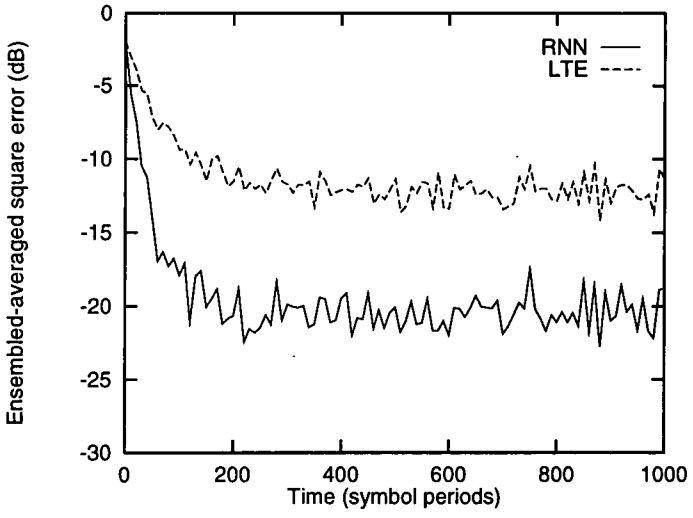

Figure 4.2: Ensembled averaged square error for the channel $H_1(z)$.

error is formed from the desired response and input to the decision device in both cases. This treats both equalisers as inverse filters rather than classifiers. The RNN MSE is lower than that of the LTE MSE due the partial 'quantising' nature of the RNN nonlinearity. The results are comparable to those obtained in [45] — the 5^{th} order LTE simulated gave a final MSE similar to that of the 20^{th} order LTE reported. As the LTE filter impulse response is the truncated inverse CIR filter impulse response (IIR), the extra taps have small values if the inverse filter response decays rapidly and thus has little affect on equaliser performance. The RNN MSE curve reaches approximately the same final levels as reported but with a stepsize, $\mu = 0.15$. The rate of convergence is faster than the rates obtained in [45] which uses relatively large stepsizes for gradient descent techniques. Figure 4.3 shows the symbol error rate of the RNN and LTE which shows that the RNN has better performance at larger SNRs but has a comparable performance as the noise power increases for this particular channel. These results are consistent with [45] which show the relative improvement of the RNN over the LTE.

The second channel, $H_2(z)$, was taken from Chen *et al* [43] in order to compare RBF, DFE and RNN equalisers. The symbol error rates obtained by simulations of a DFE and RBF / Bayesian equaliser may be compared with those in [43]. The structure of the RNN used for this 5^{th} order channel was four input nodes, two nonlinear nodes and a detector with a decision delay of 2. This structure was obtained by trial and error, as the one that trained most reliably. The DFE structure used $m = 5$ taps in the feedforward filter, $n = 4$ symbol estimates fed back and a decision delay of $d = 4$ as in [43]. Similarly, the RBF structure used $m = 5$, $n = 4$ and $d = 4$. Figure 4.4 shows the SER performance of the three equalisers. The DFE and RBF results obtained were identical to those obtained in [43]. The RNN structure considered performs considerably worse than the DFE or RBF structures.

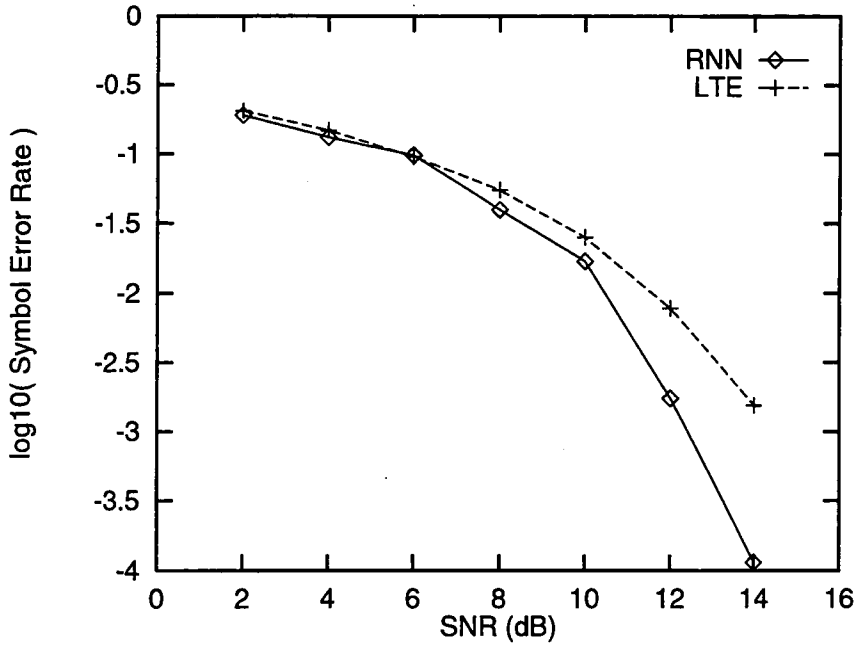

Figure 4.3: SER performance of RNN and LTE for the channel $H_1(z)$.

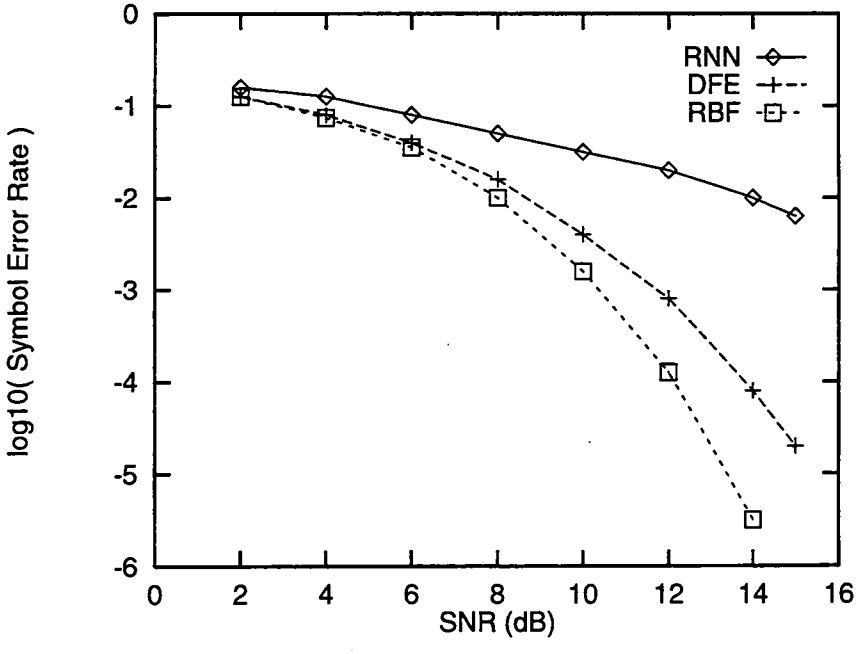

Figure 4.4: SER performance of RNN, DFE and RBF for the channel $H_2(z)$.

4.2 Analysis of Single Node RNN Dynamics

For two path channel models, $H(z) = h_0 + h_1 z^{-1}$, the RTRL adaptation algorithm creates an RNN structure that uses a single node (i.e. all weights, other than those connecting the input layer to node 0 and the feedback from node 0 to itself, are very small). This is the simplest RNN structure and a convenient starting point for the analysis of the dynamics of RNNs and the mechanisms by which they achieve equalisation. Figure 4.5 shows the structure of a single node RNN equaliser. The weights w_a, w_b may be adjusted to produce a variety of dynamics.

A simple graphical representation of the dynamics may be obtained by plotting the node's transfer function, $y(k) = \tanh(s(k))$, and lines corresponding to the current input conditions. The equations of these RNN 'state' lines are obtained by rearranging the equation for the activation of the node:

$$s(k) = w_a y(k-1) + w_b r(k) \tag{4.2}$$

$$\Rightarrow y(k-1) = \frac{s(k)}{w_a} - \frac{w_b r(k)}{w_a} \tag{4.3}$$

where $y(k-1)$ is the node output state at time $t = (k-1)$, $r(k)$ is the received signal sample and $s(k)$ is the node activation.

A family of RNN state lines is created by the multitude of possible received signal samples due to receiver noise. In the absence of noise the number of state lines is equal to the number of received signal states — equal to $M^{(L+1)}$ for a channel with $(L+1)$ paths. The current RNN output state is obtained graphically by selecting the state line corresponding to the current received signal sample and, with a knowledge of the old output state, using it to give the new activation level and corresponding output state (Fig. 4.6). The intersection of a state line with the node transfer function indicates an equilibrium point under the particular input conditions. Note that only a subset of the received signal states can exist as stable received signal states — corresponding to the repeated transmission of a single symbol. These stable signal states and corresponding RNN state lines govern the step response of the RNN.

From Eqn. 4.3, the gradient of the state line is affected by the feedback weight, w_a, and the intersection with the $s = 0$ line is affected by weights w_a, w_b and the received signal sample, $r(k)$. The transient nature of the node's step response is primarily dependent on the weight w_a.

1. **Transient response for** $0 < w_a < 1$
   Under this condition the node has a single equilibrium point and the output converges to this point independently of the initial output state. The rate of convergence is not constant and is dependent upon the transfer function gradient and the gradient of the state line. As weight w_a decreases, the rate of convergence increases. Note that this rate

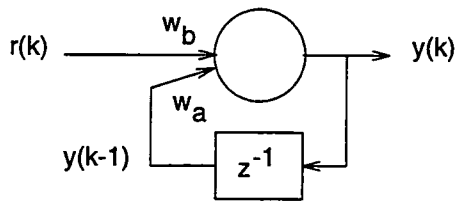

Figure 4.5: Structure of a single node RNN.

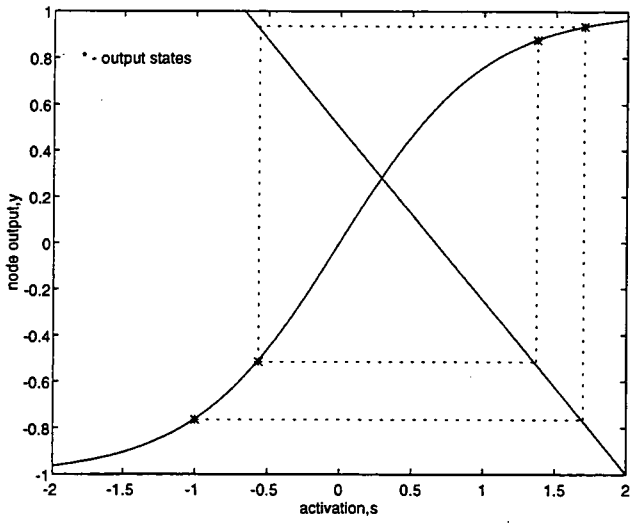

Figure 4.6: Example of the dynamics of a single node RNN.

varies over the period of transient response due to the varying transfer function gradient with the convergence rate decreasing as the output state approaches the equilibrium point. The equilibrium point is affected by w_a, w_b and $r(k)$ with an increase in either w_b or $r(k)$ increasing the equilibrium point. Receiver noise causes the received signal samples to deviate from the received signal states and thus gives rise to a variation in the node output 'equilibrium' point. Since the noise is constantly varying, these 'equilibrium' points are hypothetical as the output will not converge to any one point. The variation in output state may be reduced by designing the equilibrium point to be in the flatter regions of the node's transfer function.

2. **Transient response for $-1 < w_a < 0$**
   The node has a single equilibrium point as in the case above but the transient response is now oscillatory. The convergence rates and equilibrium points are subjected to similar effects by altering w_a, w_b or $r(k)$.

3. **Transient response for $w_a \geq 1$**
   The gradient of the state line will now be less than or equal to the gradient of the

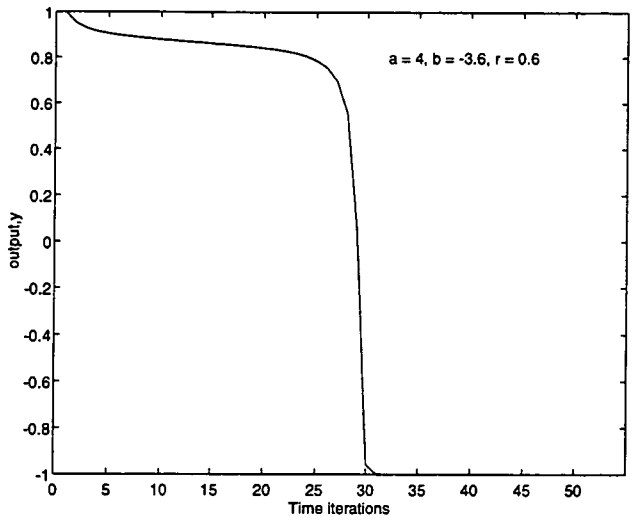

Figure 4.7: Step response for $w_a \geq 1$.

transfer function at some levels of activation. In a system with a linear transfer function this would lead to a single, unstable equilibrium point at $y = 0$ and otherwise have an unbounded output. In a practical implementation of such a system the transfer function would only be linear up to a saturation point. For small values of w_b and $r(k)$ the nonlinear nature of the transfer function causes the node to have two stable equilibrium points (one positive and one negative) and one unstable equilibrium point. Output states greater than the unstable point converge to the positive stable point and those less than the unstable point to the negative stable point. As w_b or $r(k)$ is increased the stable equilibrium points increase and the unstable point moves towards the negative stable point. Note that equilibrium points with small magnitudes are highly sensitive to changes in w_a, w_b and $r(k)$. A critical point is reached when the unstable equilibrium point merges with either the positive or negative stable points. If w_b or $r(k)$ is changed causing this critical point to be passed then the node has a single stable equilibrium point. The step response of a node with weights set just beyond the critical point is shown in Figure 4.7. The critical point occurs when the gradient of the RNN state line is equal to the gradient of the transfer function at the merged equilibrium point.

4. **Transient response for $w_a \leq -1$**

   The node has an oscillatory transient response with two stable 'equilibrium' points (one positive and one negative) and one unstable equilibrium point. The node will converge to oscillate between the two stable points rather than remaining at a single stable point. In effect the node is operating in a limit cycle. As w_b or $r(k)$ is increased both stable equilibrium points and the unstable point increase. When w_b or $r(k)$ is increased such that the gradient of the transfer function at the negative stable point is greater than

the modulus of the gradient of the state line, then the positive stable point starts to decrease whilst the other points continue to increase. Further increases in w_b or $r(k)$ cause all three points to converge towards each other resulting in a stable positive equilibrium point. Under these conditions the node ceases to operate in a limit cycle and the output converges to a single point. These conditions are met when the gradient of the transfer function at the unstable point equals the modulus of the gradient of the state line. The stable point increases with further increases in w_b or $r(k)$. Negative values of w_b or $r(k)$ result in similar characteristics except that the equilibrium points move to a single negative, stable point.

4.3 Design of a Single Node RNN Equaliser

In order to act as an equaliser the RNN must have a single stable equilibrium point corresponding to each received signal state generated by the repeated transmission of one of the alphabet of M symbols. Thus w_a and w_b must be set so that the state lines of the relevant received signal states intersect the transfer function at the required M RNN output states. If the gradient of the state lines is set such that limit cycles or two distinct equilibrium points exist then the probability of erroneous decisions is increased. The size of the limit cycles is sensitive to the position of the intersect point and the gradient of the state line. In addition to the state lines creating stable output states, there must exist another set of state lines which result in transitions between the output states. Ideally these should be set so that the output changes to the desired state in a single iteration of the RNN. A minimum condition is that a nonideal transition should not cause classification errors.

In order to highlight the limitations of the single node RNN and to give a lower bound on the probability of symbol error under noisy conditions, an example of a simple equalisation problem is studied.

Consider a bipolar PAM scheme with symbols transmitted through a time invariant channel of the form:

$$H(z) = h_0 + h_1 z^{-1}$$

where h_0 and h_1 are the (real) gains of the two paths and $-\frac{h_1}{h_0}$ is the zero of the transform. The delays are equal to the symbol period, T_s. The channel gain, $G_c = 1$, so that $(h_0^2 + h_1^2 = 1)$. The transmitted symbols, $x(k) \in \{+1, -1\}$. The received signal states, $r'(k)$, (received samples with no additive noise) are shown in Table 4.1. The first two entries give the received signal states for which the state lines should give a transition to an output state, $y(k) = +y_{eq}$ — the first from $y(k-1) = +y_{eq}$ and the second from $y(k-1) = -y_{eq}$. The $y(k) = +y_{eq}$ state is classified as symbol $x(k) = +1$ and the $y(k) = -y_{eq}$ state as $x(k) = -1$ by the decision

$x(k)$	$x(k-1)$	$r'(k)$
1	1	$+h_0+h_1$
1	-1	$+h_0-h_1$
-1	1	$-h_0+h_1$
-1	-1	$-h_0-h_1$

Table 4.1: Symbols and received signal states for the channel $H(z) = h_0 + h_1 z^{-1}$.

device. Equation 4.2 gives a pair of simultaneous equations:

$$\begin{aligned} s(k) &= +w_a y_{eq} + w_b(h_0 + h1) \\ s(k) &= -w_a y_{eq} + w_b(h_0 - h1) \end{aligned} \tag{4.4}$$

Solving for w_a and w_b:

$$w_a = \frac{-h_1 s}{h_0 y_{eq}} \tag{4.5}$$

$$w_b = \frac{s}{h_0} \tag{4.6}$$

where $s = \tanh^{-1}(y_{eq})$ — a constant.

Solving for w_a and w_b using the second pair of received signal states for transitions to the $y(k) = -y_{eq}$ state gives the same results. These equations define the weights for a zero decision delay equaliser.

The weight w_a is proportional to the position of the channel zero and the state lines have positive gradients when $\frac{h_1}{h_0} < 0$. For a minimum phase channel ($\left|\frac{h_1}{h_0}\right| < 1$) the equaliser will have a single equilibrium point. For a single equilibrium point:

$$\left|\frac{1}{w_a}\right| > \frac{d}{ds}\tanh(s). \tag{4.7}$$

Substituting for w_a with Eqn. 4.5 and using the fact that $y_{eq} = \tanh(s)$,

$$\begin{aligned} \left|\frac{h_0 \tanh(s)}{-h_1 s}\right| &> \frac{d}{ds}\tanh(s) \\ \Rightarrow \left|\frac{h_0 \tanh(s)}{-h_1 s}\right| &> \left|\frac{4e^{2s}}{(e^{2s}+1)^2}\right| \end{aligned} \tag{4.8}$$

as $\frac{d}{ds}\tanh(s) \geq 0$. Therefore

$$\begin{aligned} \left|\frac{e^{2s}-1}{e^{2s}+1}\right|\left|\frac{h_0}{h_1 s}\right| &> \left|\frac{4e^{2s}}{(e^{2s}+1)^2}\right| \\ \Rightarrow \qquad \left|\frac{h_1}{h_0}\right| &< \left|\frac{e^{4s}-1}{4se^{2s}}\right| \end{aligned} \tag{4.9}$$

Thus the limit on $\left|\frac{h_1}{h_0}\right|$ is always greater than 1 and increases as the equilibrium point is made larger. For output states of $y_{eq} = \pm 0.8$ then $\left|\frac{h_1}{h_0}\right| < 2.02$. The RNN operates with multiple equilibrium points if the magnitude of the channel zero increases beyond this point. Either case causes an increased probability of symbol classification error under noisy conditions.

4.3.1 Effects of Noise on RNN Equaliser Performance

A lower bound on the probability of symbol error may be obtained by considering the primary error mechanism of the single node RNN. Consider a sequence of symbols $x(k) = +1$ transmitted through a channel corrupted by zero-mean noise either in the channel or receiver. The noise causes variations in the intersections of the state lines with the $s(k) = 0$ line and classification errors occur when this results in a shift in output state to one with opposite sign. The RNN output is nominally at the positive equilibrium point but oscillates about this point due to the noise. The intersection of state lines with the $s(k) = 0$ line is given by substituting Eqns. 4.5 and 4.6 into Eqn. 4.3 and setting $s(k) = 0$ as required:

$$\begin{aligned} y(k-1) &= \frac{-w_b r(k)}{w_a} \\ &= \frac{+y_{eq}}{h_1} r(k) \end{aligned} \tag{4.10}$$

where $r(k) = r'(k) + n(k) = h_0 + h_1 + n(k)$.

For state lines with negative gradients an error will result if the noise sample, $n(k)$, causes the intersection to fall below the current output state. The classification decision boundary is formed by equating the intersection point with the current output state — equivalent to setting $s(k) = 0$ in Eqn. 4.2 and substituting for w_a and w_b. If the output is at the positive equilibrium point, $y(k-1) = +y_{eq}$ (i.e. the stable point corresponding to $r'(k) = h_0 + h_1$), the magnitude of the additive noise sample causing an error is given by:

$$\frac{-w_b}{w_a}(r'(k) + n(k)) < +y_{eq}. \tag{4.11}$$

Substituting for w_{a0}, w_b and $r(k)$:

$$\begin{aligned} \frac{y_{eq}}{h_1}(h_0 + h_1 + n(k)) &< +y_{eq} \\ \Rightarrow \qquad n(k) &< -h_0. \end{aligned} \tag{4.12}$$

For a zero-mean white Gaussian noise process the probability of error is given by:

$$P_e(n(k) < -h_0) = \int_{-\infty}^{-h_0} \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{\frac{-n(k)^2}{2\sigma_n^2}} dn(k) \tag{4.13}$$

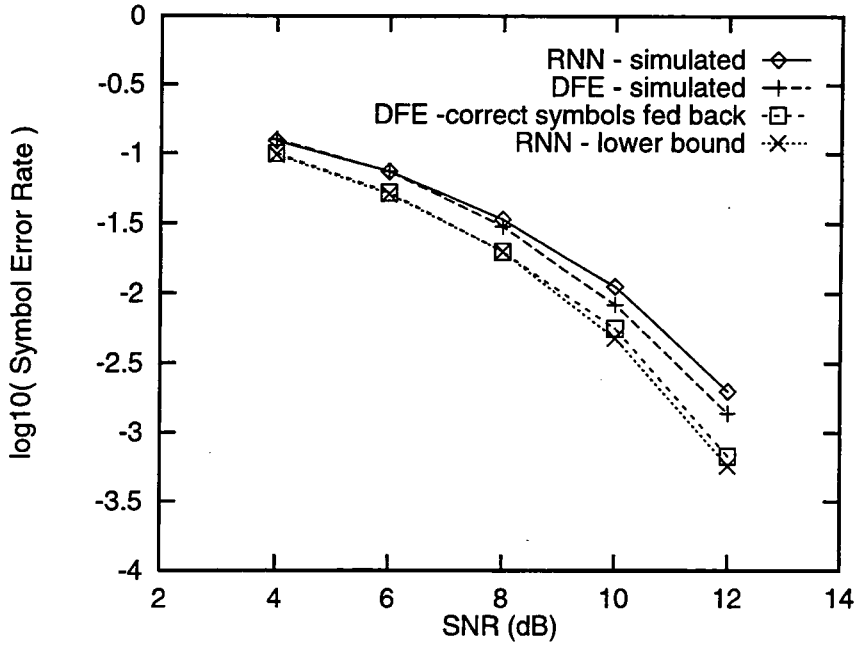

Figure 4.8: Symbol error rate performance of a single node RNN.

where σ_n^2 is the variance of the noise power. Expressions for the magnitude of noise required for the remaining 3 state lines to give errors result in $|n(k)| > h_0$ and thus each has the same probability of error.

$$P_e = \frac{1}{2}\text{erfc}\left(\frac{h_0}{\sqrt{2\sigma_n^2}}\right) \tag{4.14}$$

For a given output state the equaliser is less immune to noise as $\left|\frac{h_1}{h_0}\right|$ increases.

Actual probabilities of error will be higher if the output state is not equal to the nominal output state when the particular noise sample occurs. This is especially true when the equaliser is changing between classification states. Therefore the effects of noise on one symbol affect the performance of the equaliser on subsequent symbols even though noise on the first sample may not have resulted in an error. Figure 4.8 shows the plot of SER vs. SNR for the analytical lower bound and results from simulations. The channel model is $H(z) = 0.8192 + 0.5734z^{-1}$ and the RNN output nodes were designed to have stable equilibrium points, $p = \pm 0.8$, which result in designed weights of $w_a = -0.961$ and $w_b = 1.341$. This channel is the power normalized version of channel $H_1(z)$ in Section 4.1. No training was used as the weights have been determined theoretically. As expected the simulated results give higher symbol error rates than the analytic lower bound. It was also noted that the relevant weight settings obtained for the two node RNN trained using the RTRL algorithm on channel $H_1(z)$ (Section 4.1) were very similar to the designed settings (taking into account the effect of the unnormalized channel and variations inherent in the adaptation algorithm). The 3^{rd} and 4^{th} curves are explained in Section 4.3.4.

4.3.2 Effects of Decision Delay on RNN Equaliser Performance

By delaying the classification decision by one symbol period (i.e. the detector ouputs the symbol estimate, $\widehat{x}(k-1)$, after receiving the sample $r(k)$), nonminimum phase channels having a zero with larger magnitude than that given by Eqn. 4.9 may be equalised. From Table 4.1 the equaliser must now have a positive equilibrium point when $r'(k) = +h_0 + h_1$ or $r'(k) = -h_0 + h_1$. These new conditions result in different equations for w_a and w_b:

$$w_b = \frac{s}{h_1} \tag{4.15}$$

$$w_a = \frac{-h_0 s}{h_1 y_{eq}} \tag{4.16}$$

where $s = \tanh^{-1}(y_{eq})$. By a similar analysis to that in Section 4.3, the limit on $\left|\frac{h_1}{h_0}\right|$ for a single equilibrium point becomes:

$$\left|\frac{h_1}{h_0}\right| > \left|\frac{4se^{2s}}{e^{4s}-1}\right| \tag{4.17}$$

For output states of $y_{eq} = \pm 0.8$, $\left|\frac{h_1}{h_0}\right| > 2.02$ and thus the equaliser cannot operate on minimum phase channels without multiple equilibrium points for each state line. The magnitude of noise samples causing errors by the mechanism described in Section 4.3.1 are given by:

$$|n(k)| > h_1 \tag{4.18}$$

and for additive white Gaussian noise the probability of error is given by:

$$P_e = \frac{1}{2}\text{erfc}\left(\frac{h_1}{\sqrt{2\sigma_n^2}}\right) \tag{4.19}$$

The improvement in noise immunity as $\left|\frac{h_1}{h_0}\right|$ increases is smaller than the improvement for a corresponding decrease in a zero decision delay equaliser.

4.3.3 Equalisation of Higher Order Channel Models

When the order of the channel model increases, the design of the equaliser must allow for the increased number of received signal states. An n^{th}-order model has M^n received signal states for an M-ary PAM scheme. For binary PAM and a 3^{rd} order model, the equaliser has 8 state lines of which two should result in a transition from the $+p$ state to the $+p$ state. The received signal states corresponding to these two state lines are not, in general, equal to each other and, as weights a and b are fixed, the equaliser cannot be designed to give the same output, $y(k) = +y_{eq}$, for both cases. A sub-optimum solution may be designed with the

two output states close to the desired output state. The difference between them depends upon the coefficients of the FIR channel model and the desired output state. Clearly if this desired output state is in the flatter region of the transfer function then the difference in actual output states is much smaller.

4.3.4 Comparisons with the Decision Feedback Equaliser

Many of the limitations of the single node RNN are removed if the node's transfer function is replaced by a binary quantiser (in the case of a binary PAM equaliser). This allows a greater degree of freedom in choosing the weights w_a, w_b as there are no distinct values of activation which give the desired output states, $y(k) = \pm y_{eq}$.

Such a node exhibits a transient response which converges to the required output state in a single time iteration as long as receiver noise does not cause a classification error. This property still holds in the presence of sufficiently small magnitudes of receiver noise unlike a node with a sigmoid type transfer function. Since the gradient of the quantiser transfer function is zero at all points other than at $s(k) = 0$, limit cycles or multiple equilibrium points do not exist and the bounds on $\left|\frac{h_1}{h_0}\right|$ may be removed.

Variations of the state lines due to noise do not affect the output states as long as the $s(k) = 0$ decision boundary is not crossed. The distinct advantage of this is that noise which does not cause an error on one symbol has no further effect on the classification of future symbols. In effect, nodes with sigmoid type transfer functions feed a filtered version of the noise back to the node inputs which increases the effective SNR and degrades the equaliser performance. The filtering is due to the nonlinear nature of the transfer function. Thus the symbol error rate will be lower than a node with a sigmoid type transfer function for a given SNR.

This alternative single node RNN is precisely the same as a decision feedback equaliser (DFE) without a feedforward filter and having a single feedback tap. Figure 4.8 shows the increased performance of this DFE structure. The fourth curve shows the results of feeding the correct symbols, $x(k)$, back rather than the classified symbols, $\widehat{x}(k)$. Theoretically, this curve should be the same as the curve for the single node RNN lower bound. The degradation in performance due to erroneous classifications is apparent.

The DFE is a conventional nonlinear equaliser structure that uses feedback to increase the minimum distance between the received signal states and the decision boundaries. This process may be viewed as using past symbol decisions to reduce the number of received signal states which the decision boundaries must separate. The feedback varies the position of the decision boundaries depending on previously classified symbols. The decision boundaries that are formed are a linear function of the input received signal samples at any given classification time. The nonlinearity lies in the fact that the position of the linear decision boundaries are

not fixed.

By using more than one feedback tap, the DFE can reduce a larger set of received signal states when equalising higher order channel models. The minimum distance between the reduced set of states and the decision boundaries may be increased by applying a feedforward FIR filter on the input of the equaliser to form higher dimensional boundaries. The FIR filter forms linear decision boundaries of dimension equal to the order of the filter. This filtering may be incorporated into the general RNN structure by increasing the number of input nodes. These nodes form a tapped delay line for the received signal samples. The weighted sum of the outputs of the delay line formed by the recurrent node while calculating the activation is clearly an FIR filtering process. Thus the advantages of pre-filtering the received signal samples may be implemented with ease into the RNN structure. However, at any given classification time, the decision boundaries are still linear functions of the current received signal samples and the ability of neural networks to form nonlinear boundaries is not exploited.

4.3.5 Single node RNN Training Considerations

An RNN node with a quantising / hard limiting transfer function produces a system that is extremely difficult to train by means of gradient descent of an error function. In the RTRL algorithm, the error function is formed by taking the square of the difference between the desired and actual output of the node. Differentiating the error function requires the derivative of the node output with respect to the weights — which is zero at all but an isolated set of points. Thus the use of the instantaneous square error as a error function is unsuitable. A short term average may be formed to produce larger variations in the error function.

The adaptive DFE overcomes this problem by forming the error function from the desired output and the signal immediately prior to the quantiser. This method provides a relatively smoother function for classical gradient descent techniques, subject to the effects of incorrect symbols being fed back during adaptation.

4.4 Multinode RNN Equaliser Design

Having noted the similarities in the mechanisms of the single node RNN and the DFE, the design of multinode RNNs based on ideas from DFEs is examined. This approach has been adopted due to the following observations:

1. The structure of the RNN is very similar to that of the DFE and single node RNNs, adapted using the RTRL algorithm, mimic the DFE.

2. The dynamics of the single node RNN are varied and those of multinode RNNs are even more so. Thus design of multinode RNNs with hyperbolic tangent nonlinearities is not straightforward especially as there is no precise output specification for nodes other than the output node whose signal is fed to the classification device.

3. The adaptation of multinode RNNs with hyperbolic tangent nonlinearities is difficult and the RTRL algorithm does not guarantee convergence to the global minimum of the multimodal error surface.

The DFE uses feedback to cancel the postcursors of the sampled CIR convolved with the DFE's feedforward filter impulse response [46]. For a given decision delay, d symbol periods, the feedback cancels out the symbol energy occuring in the received samples for all symbols prior to the d^{th} symbol. Feedback creates an error propagating mechanism due to incorrectly classified symbols erroneously cancelling symbol energy and increasing intersymbol interference. This additional interference may cause further classification errors. Thus it is possible for the effect of an incorrect decision to exist after the erroneous symbol has been shifted out of the feedback delay line. For many channels error propagation is not catastrophic and the equaliser is capable of recovering.

Once the DFE has classified a particular received sample there is no mechanism whereby the decision may be updated as more energy from the original transmitted symbol is received, i.e. there is no 'soft'-decision capability. If the decision delay is such that classification takes place after all, or nearly all, the transmitted symbol energy has been received, then there will be no subsequent information available to facilitate the decision in a symbol-by-symbol decision equaliser. However, if system constraints are such that a decision must be made before all the symbol energy has been received, then a mechanism by which decisions may be updated could potentially increase equaliser performance. This requirement is beneficial in systems where symbol estimates are used to aid tracking of a time variant channel — decision directed adaptation (see Chapter 5 for examples).

The fully recurrent RNN structure may be modified in such a manner so as to provide a soft-decision facility. The design requires that the output of each node to be constrained to perform a specific function. In the RNN structures reported previously [45], only one node output is specified — the remainder form an arbitrary nonlinear function determined by the adaptation algorithm via the network weights so as to minimise an error metric. In the following sections, two RNN structures are developed — the second structure is more fully recurrent and is closer in structure to the RNN equalisers reported previously [45]. An adaptation algorithm for networks with quantising nodes is formulated using the LMS algorithm as a basis.

4.4.1 RNN with Symbol Feedback — RNN1

A bank of DFEs operating in parallel and each with a differing decision delay may be viewed as a multinode RNN with a partial set of feedback connections. This structure is shown in Figure 4.9. Each node forms the weighted sum of its inputs and passes the result (the node activation, $s_i(k)$) through a slicer decision device to give the node output, $y_i(k) = \widehat{x}(k-i)$. Instead of employing symbol feedback delay lines, each node passes its output to nodes requiring that symbol in order to cancel intersymbol interference. The top node has zero decision delay and each of the lower nodes has an increasing decision delay of one symbol period per node.

The choice of input connections and feedback connections is explained by considering a sampled channel impulse response of the form:

$$H(z) = h_0 + h_1 z^{-1} + \cdots + h_L z^{-L} \tag{4.20}$$

The received sample, $r(k)$, is given by the convolution of the transmitted symbol sequence and the channel impulse response:

$$r(k) = h_0 x(k) + h_1 x(k-1) + \cdots + h_L x(k-L) + n(k) \tag{4.21}$$

where $x(k)$ are the transmitted symbols and $n(k)$ is the additive noise term.

For the node with zero decision delay (i.e. the node estimating $x(k)$) the only received sample containing energy from $x(k)$ is $r(k)$. Likewise, for the node with decision delay $d = 1$ the received samples containing energy from $x(k-1)$ are $r(k)$ and $r(k-1)$. Thus the nodes only have a partial set of input connections from the set of received samples, $\{r(k), \ldots, r(k-n+1)\}$. In general node i has a set of input weights, a_{ij}, from the input $r(k-j)$ to node i :

$$a_{ij} \quad \begin{cases} i = 0, \ldots, n-1 \\ j \leq i \end{cases} \tag{4.22}$$

where n is the number of nodes in the network.

Feedback between nodes is restricted so that nodes only receive inputs fed back from nodes with greater or equal decision delay than their own. This policy is adopted so that only the 'tail' of the combined sampled impulse response is cancelled. For example, the node with a single decision delay (node 1) forms, from the received samples, the sum

$$\begin{aligned} a_{10}r(k) + a_{11}r(k-1) \;=\; & a_{10}(h_0 x(k) + h_1 x(k-1) + \cdots + h_L x(k-L)) \\ & + a_{11}(h_0 x(k-1) + h_1 x(k-2) + \cdots + h_L x(k-L-1)) \\ & + a_{10}n(k) + a_{11}n(k-1) \end{aligned} \tag{4.23}$$

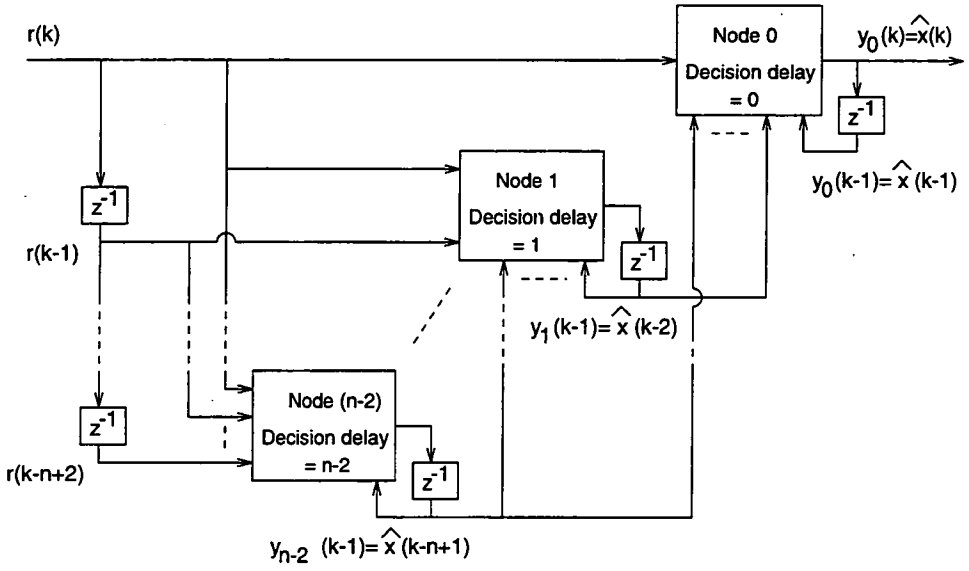

Figure 4.9: Schematic of multinode RNN equaliser - RNN1.

The feedback inputs cancel all terms involving $x(k-2), x(k-3), \ldots, x(k-L-1)$ which removes the intersymbol interference generated by these symbols. The feedback weights, b_{ij}, connecting the ouput of node j to node i, are given by:

$$\begin{aligned}
b_{11} &= -(a_{10}h_2 + a_{11}h_1) && \text{— cancelling terms in } x(k-2) \\
b_{12} &= -(a_{10}h_3 + a_{11}h_2) && \text{— cancelling terms in } x(k-3) \\
&\vdots && \\
b_{1,n-2} &= -(a_{10}h_L + a_{11}h_{L-1}) && \text{— cancelling terms in } x(k-L) \\
b_{1,n-1} &= -(a_{11}h_L) && \text{— cancelling terms in } x(k-L-1)
\end{aligned} \tag{4.24}$$

In general the feedback weights are related to the input weights, a_{ij}, and the channel coefficients, h_i, by:

$$b_{ij} = \begin{cases} 0 & i > j \\ -\sum_{p=0}^{i} a_{ip} h_{(j+1-p)} & i \leq j \end{cases} \tag{4.25}$$

where $i, j = 0, \ldots, n-1$. It is assumed that $h_i \mid_{i>L} = 0$ as per Eqn. 4.20.

A design issue arises in the selection of the number of nodes in the network. For the node

with zero decision delay a total of L feedback connections are required to fully cancel the intersymbol interference. This requires a further $(L-1)$ nodes as 1 feedback connection is provided from the node itself — giving a total network size of L nodes and with decision delays of 0 to $(L-1)$. However for the node with 1 sample decision delay, the input samples $r(k)$ and $r(k-1)$ involve terms of $\{x(k),\ldots,x(k-L-1)\}$ and so an extra node with decision delay of L is required for complete cancellation of the interfering symbol terms. Thus the network will grow without bound if each node accepts all received samples containing energy from the transmitted symbol it is estimating. To circumvent this problem either the requirement of the complete cancellation of interfering terms must be relaxed or else a node must generate the cancelling terms by passing its output through a delay line — as in the DFE.

Adopting a feedback delay line to provide symbol estimates not generated explicitly by nodes will reduce the level of intersymbol interference given that the symbol estimates are correct. To increase the probability of this assumption, the output of the node with the highest decision delay should form the input to the delay line. This node's symbol estimate will generally be the best as it has been formed using more received samples containing the transmitted symbol energy. If the end samples of the channel impulse response are very small the extra energy available to improve the symbol estimate may be swamped by the energy from symbols transmitted after the symbol being estimated. Under these circumstances the adaption algorithm should ideally adjust the weights from such received samples to small values so that these samples are, in effect, not used in the estimation process.

A feedback delay line of length $(L-1)$ provides estimates of all symbols with terms appearing in the set of received samples not directly estimated by nodes. For a network of L nodes, all nodes other than that with zero decision delay will use a partial set of these feedback symbols to cancel intersymbol interference. The set of weights, c_{ij}, connecting the output of the j^{th} element of the delay line to node i are given by:

$$c_{ij} = \begin{cases} 0 & i \leq j \\ -\sum_{p=0}^{i} a_{ip} h_{(j+n+1-p)} & i > j \end{cases} \tag{4.26}$$

where $i = 0,\ldots,n-1$ and $j = 0,\ldots,(L-2)$. It is assumed that $h_i\,|_{i>L} = 0$ as per Eqn. 4.20.

In a network of L nodes, the last node uses the received samples $\{r(k),\ldots,r(k-L+1)\}$ together with feedback connections to form its estimate of $x(k-L+1)$. This estimate is made without using all the samples containing energy from the transmitted symbol $x(k-L+1)$ — the remaining sample containing energy from this symbol, $r(k+1)$, has not yet been received. In many practical channels the symbol energy contained in this final sample is small and thus will not influence the decision significantly.

4.4.2 RNN with Symbol Feedback and Symbol Feedforward — RNN2

The equaliser structure described in Section 4.4.1 may be configured so that it has a larger set of feedback connections between the nodes and therefore begins to resemble the fully recurrent structure as described in Kechriotis *et al* [45]. By permitting nodes with low decision delay to feed their decisions forward to nodes with higher decision delay, intersymbol interference occuring due to symbols transmitted after the symbol being classified may be cancelled. This is equivalent to cancelling part of the precursor of the combined sampled impulse response.

Figure 4.10 shows the structure of the equaliser with the additional internode connections. The network is not fully interconnected with no symbol feedforward occuring between a node and the node with one extra symbol period decision delay. The unit delay inherent in the symbol feedback/forward process means that the symbol being fed back/forward at time $t = k$ is $\widehat{x}(k-d-1)$. This is the symbol that the node with decision delay $(d-1)$ is estimating and therefore the estimated symbol $\widehat{x}(k-d-1)$ is of no use in cancelling intersymbol interference.

The input connections to each node remain the same as the received samples containing a specific symbol's energy has not changed. The input weights, a_{ij} are given by Eqn. 4.22.

The feedback weights, b_{ij}, are given by:

$$b_{ij} = \begin{cases} 0 & i = j+1 \\ -\sum_{p=0}^{i} a_{ip} h_{(j+1-p)} & i \neq j+1 \end{cases} \tag{4.27}$$

where n is the number of nodes in the network and $i = 0, \ldots, n-1$. It is assumed that $h_i \mid_{i>L} = 0$ as per Eqn. 4.20.

The number of nodes in the network must be limited as explained in Section 4.4.1. By applying such a limit, the requirement of an additional delay line to cancel interfering symbol terms still exists. This delay line has the same length, $L-1$, as in the structure RNN1. The weights connecting the delay line to the nodes, c_{ij} are given by Eqn. 4.26.

If the number of nodes in the network, $n \leq 2$, then RNN structure 2 will be the same as RNN structure 1. Thus, by adopting a network of L nodes, RNN structure 2 may only be applied to channels with sampled impulse responses with $L = 3$ (4 paths) or greater.

4.4.3 LMS Adaptation of the Network Weights

The networks described are very similar to the DFE and so adaptation algorithms suitable for the DFE are applicable to the RNN networks given a few modifications. The LMS algorithm operating in conjunction with a known training signal is selected for simulations

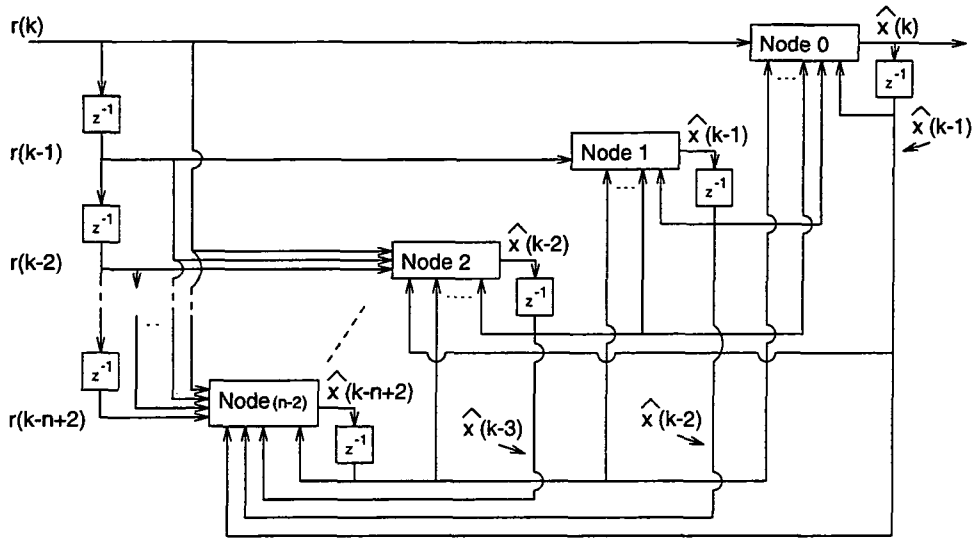

Figure 4.10: Schematic of multinode RNN equaliser - structure 2.

reported in this chapter due to its simplicity. The speed of adaptation and operation on time varying /fading channels is not an issue at this juncture.

The LMS algorithm attempts to minimise the instantaneous square error between a desired signal and an output signal by adjusting the parameters controlling the output signal. In the case of the RNN networks the node output cannot be used to form the error signal due to the node's nonlinearity (the quantiser decision device). Thus the node activation, $s_i(k)$, (the signal prior to the quantiser) is used as the output signal as in the DFE.

The desired response of each node in the network is specified allowing separate error signals to be formed for each node. These error signals may be combined to give a network performance measure rather than individual node performance measures. By assuming each node has correct symbol feedback at all times, each node may be treated as a separate entity and allows the weights on all the node's inputs to be adjusted using the local node error signal. This removes the requirement for a network performance measure. If this assumption is not made then a node's output is dependent on all other node outputs and thus weights rather than just the node's own input weights. The LMS adaptation algorithm is then required to operate on a nonlinear system. Under correct operation of the network the assumption of correct symbol feedback is valid and is the same assumption as used in the application of the LMS algorithm to the DFE [2].

The LMS algorithm for the RNN structures reduces to a separate adaptation algorithm for each node. At time $t = k$ the error at node i and is given by:

$$e_i(k) \quad = \quad d_i(k) - s_i(k)$$

$$= x(k-i) - s_i(k) \tag{4.28}$$

where $d_i(k)$ is the desired response for the node. The instantaneous square error $J_i(k)$ is use as an approximation to the mean square error and the weights a_{ij}, b_{ij} and c_{ij} are adjusted by gradient descent to minimise this error metric.

The node error metric is given by

$$J_i(k) = \frac{1}{2}(e_i(k))^2 \tag{4.29}$$

The error gradient and weight update equation for a_{ij} is

$$\begin{aligned} \frac{\partial J_i(k)}{\partial a_{ij}(k)} &= e_i(k)\frac{\partial e_i(k)}{\partial a_{ij}(k)} \\ &= -e_i(k)r(k-j) \qquad\qquad j \leq i \end{aligned} \tag{4.30}$$

$$\begin{aligned} a_{ij}(k+1) &= a_{ij}(k) - \mu\frac{\partial J_i(k)}{\partial a_{ij}(k)} \\ &= a_{ij}(k) + \mu e_i(k)r(k-j) \end{aligned} \tag{4.31}$$

The error gradient and weight update equation for b_{ij} is

$$\begin{aligned} \frac{\partial J_i(k)}{\partial b_{ij}(k)} &= e_i(k)\frac{\partial e_i(k)}{\partial b_{ij}(k)} \\ &= -e_i(k)y_j(k-1) \qquad\qquad j \geq i \qquad \text{— RNN1} \\ &\qquad\qquad\qquad\qquad\quad j \neq (i-1) \text{ — RNN2} \end{aligned} \tag{4.32}$$

$$\begin{aligned} b_{ij}(k+1) &= b_{ij}(k) - \mu\frac{\partial J_i(k)}{\partial b_{ij}(k)} \\ &= b_{ij}(k) + \mu e_i(k)y_j(k-1) \end{aligned} \tag{4.33}$$

The error gradient and weight update equation for c_{ij} is

$$\begin{aligned} \frac{\partial J_i(k)}{\partial c_{ij}(k)} &= e_i(k)\frac{\partial e_i(k)}{\partial c_{ij}(k)} \\ &= -e_i(k)q_j(k) \qquad\qquad j < i \end{aligned} \tag{4.34}$$

$$\begin{aligned} c_{ij}(k+1) &= c_{ij}(k) - \mu\frac{\partial J_i(k)}{\partial c_{ij}(k)} \\ &= c_{ij}(k) + \mu e_i(k)q_j(k) \end{aligned} \tag{4.35}$$

where μ is the adaptation step size and q_j are the contents of the feedback delay line.

Whilst the training sequence is being received and the network weights are being updated

the output from each node is replaced by the desired output after each adaptation step. This ensures that correct symbol feedback to the nodes is maintained during the adaptation process in accordance with the adaptation algorithm assumptions. Failure to do so may induce poor equaliser performance.

4.4.4 Performance Evaluation

The performance of the multinode RNN equaliser structures was evaluated by simulating the SER of the equalisers as a function of SNR whilst operating on the normalised power channels:

$$\begin{aligned} H_3(z) &= 0.4082 + 0.8164z^{-1} + 0.4082z^{-2} \\ H_4(z) &= 0.2041 + 0.4082z^{-1} + 0.6124z^{-2} + 0.6124z^{-3} + 0.2041z^{-4} \end{aligned}$$

and comparing the results with the SER of a conventional DFE and RBF /Bayesian equalisers. In these channels the delays are equal to the symbol period, T_s. The receiver noise was AWGN. All simulations used a bipolar PAM scheme for the transmitted symbols without pulse shaping as the channel was simulated using one sample/symbol. Prior to SER measurement, the equalisers were trained for 2000 symbols using an LMS algorithm with a common stepsize of $\mu = 0.05$ in all adaptation algorithms. A smaller step size will result in a lower MSE and therefore produce a lower SER as the weights may be adjusted more accurately and are subject to smaller variations once in the vicinity of the MSE surface minimum [47]. Since a comparative study is being performed, the absolute values of SER are not as significant as the relative performance between the various equaliser structures.

After each adaptation process was completed, 10^4 symbols were transmitted and the number of errors accumulated. This process was repeated until 10^5 or 10^6 nontraining symbols had been transmitted and an ensemble averaged SER calculated. This policy was adopted so that the variations in the weight settings inherent in the LMS adaptation algorithm are averaged.

Under the assumption that each symbol error is binomially distributed and independent of other symbol errors, the SER measurement will be a Gaussian distributed statistic. For 90% confidence that the true SER lies within $\pm 5\%$ of the measured value, the SER must be > 0.011 for 10^5 transmitted symbols and > 0.0011 for 10^6 transmitted symbols. These limits are given by (see [48]):

$$SER_{min} = \left(\frac{1.645}{0.05 \cdot \sqrt{N}} \right)^2$$

where N is the number of symbols transmitted.

The assumption of independent error events is not totally valid (see Section 4.4) but is used in order to obtain an approximation to the required number of symbols to be transmitted

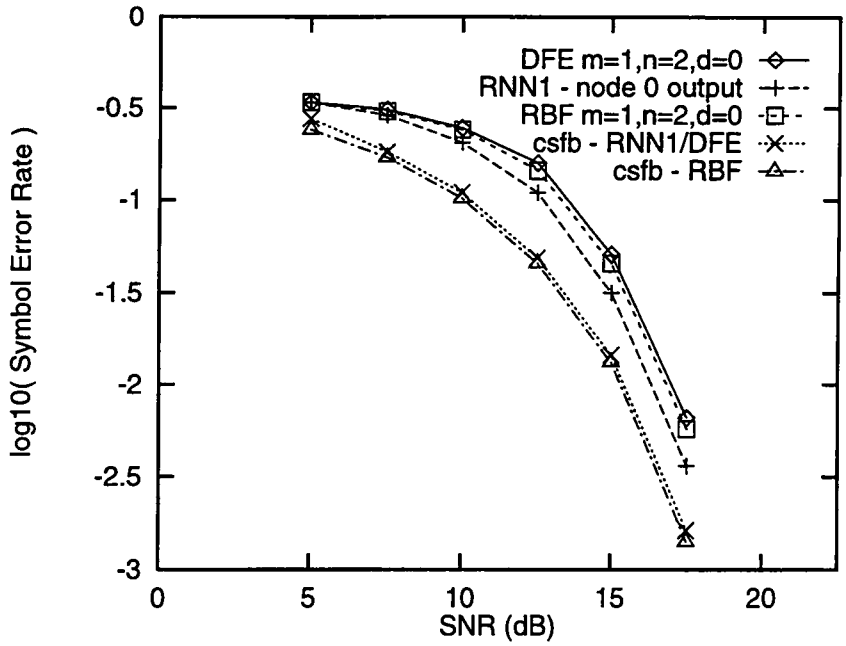

Figure 4.11: SER performance of RNN1 and DFE.

in order to achieve valid SER measurements.

The SER measurement was obtained by operating the equalisers on the same data and noise sequences for each curve in a given figure. This allows the relative performance of the equalisers to be compared. Repeating the measurements with different noise sequences showed that the variance of the SER estimates was $> \pm 10\%$, often being $> \pm 7\%$.

The RBF node centres were postitioned using a channel estimator of the same order as the CIR and adapted using the LMS algorithm and the same stepsizes and training sequences as the RNNs and DFE.

Channel $H_3(z)$

The three symbol period spaced path channel, $H_3(z)$, is a partial response channel having a single deep spectral null. RNN2 is not applicable to this channel and so the simulations compare the performance of RNN1 against a DFE and RBF equaliser.

The RNN structure consisted of two nodes using up to two received samples. The last node feeds its previous decisions through a delay line of length 1 in order to generate the symbol estimate, $\widehat{x}(k-3)$ at time $t = k$. Both the DFE and RBF had a received sample delay line, $m = 1$, a decision delay, $d = 0$ and used $n = 2$ past decisions to cancel ISI or select a subset of the network nodes in the case of RBF.

Figure 4.11 shows the SER against SNR for the three equalisers together with the SERs

obtained using correct symbol feedback. The difference between the SER curves for estimated and correct symbol feedback show that RNN1 suffers the least from incorrect symbol feedback. The DFE and RBF structures used in this simulation are appropriate only for zero decision delay and, as soon as a delay (one or two symbol periods) is used, both equalisers outperform RNN1.

Channel $H_4(z)$

The five sample channel, $H_4(z)$, is a nonminimum phase channel with most of the energy concentrated around the 3^{rd} and 4^{th} samples. The SER performance of both RNN structures may be compared against a DFE and RBF equalisers.

The RNN structures had four nodes and inputs of up to four received signal samples. A feedback delay line of length 3 fed by node 4 was used to generate the symbol estimates $\{\widehat{x}(k-6),\ldots,\widehat{x}(k-8)\}$ at time $t=k$. The four comparable DFE and RBF structures had $m=(d+1)$ received sample inputs where the decision delay, $d=0,\ldots,3$ and each used $n=4$ past decisions to either cancel ISI or select a subset of the RBF network nodes. The different DFE and RBF structures were used to compare the performance of the output of each of the four nodes in the RNN structures and in addition, the correct symbol feedback curves are shown to indicate the best theoretical performance of the equalisers.

Figure 4.12 shows the performance of the equalisers operating with $d=0$ — the 2 RNN structures using node 0 ($d=0$) as the equaliser output. The lower two curves show the performance of the three equaliser structures operating with correct symbol feedback / feedforward. This curve is the same for both RNN and DFE equalisers as the input configuration of node 0 of the RNNs and of the DFE is identical. The performance improvement of the RNN equalisers over the DFE and the RBF equaliser can be seen. The differences between the 2 RNN structures is noticable indicating that node 0 output is affected by symbol feedforward to lower nodes and the subsequent symbol feedback to node 0.

Figure 4.13 compares the performance with $d=1$ — the RNN structures using node 1 ($d=1$) as their output. RNN1 and the DFE have similar SERs indicates that the soft decision capability of RNN1 does not improve performance in this instance. RNN2 suffers from the affects of symbol feedforward even though the node 1 does not use feedforward cancellation directly. The difference between RNN2's actual and correct symbol feedback curves indicates that the structure is sensitive to symbol feedforward. The RBF equaliser outperforms the other equalisers, both theoretically and practically.

Figure 4.14 indicates that for $d=2$ the DFE and RNN1 have very similar performance and again RNN2 has a larger SER. This plot shows two correct symbol feedback curves for the RNN structures — the first for RNN1 and second for RNN2. Node 2 of RNN2 uses the output of node 0 as a symbol feedforward input and so cancels a greater portion of the ISI

occuring in its received sample inputs. Under correct symbol feedback (and feedforward) conditions this makes RNN2 more able to classify the recieved samples correctly. It is noted that the performance gap between ideal and actual is much greater for RNN2 than RNN1 and the DFE.

Finally, Figure 4.15 shows the performance of the structures with $d = 3$. The RNN1 and the DFE have the same structure and so their performance is identical. RNN2 has a consistently poorer SER compared to RNN1, the DFE and RBF equalizers. Once more the deviation between theoretical and actual performance is greater for RNN2. The RBF equaliser consistently outperforms the DFE and RNNs. The symbol feedforward of RNN2 does improve over the RBF in the theoretical case as both the pre- and postcursors of the CIR are cancelled. However, the poor symbol estimates used to cancel precursors degrades the RNN2 performance severely.

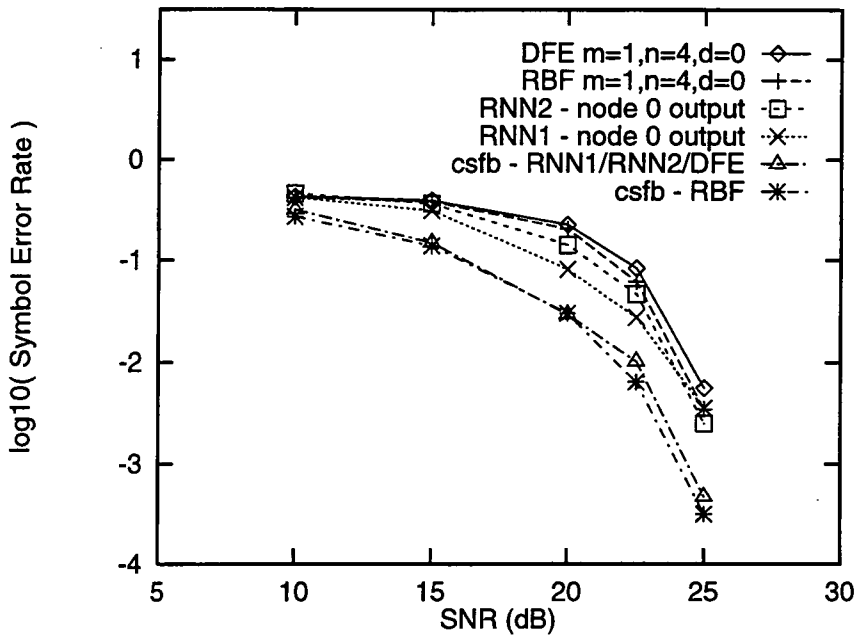

Figure 4.12: SER performance of RNN1, RNN2, DFE and RBF with $d = 0$.

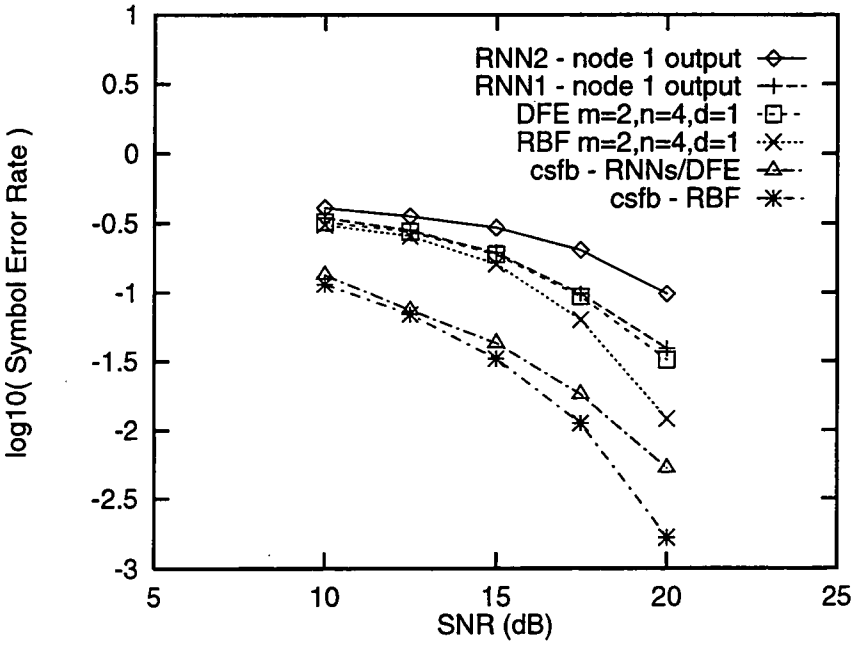

Figure 4.13: SER performance of RNN1, RNN2, DFE and RBF with $d = 1$.

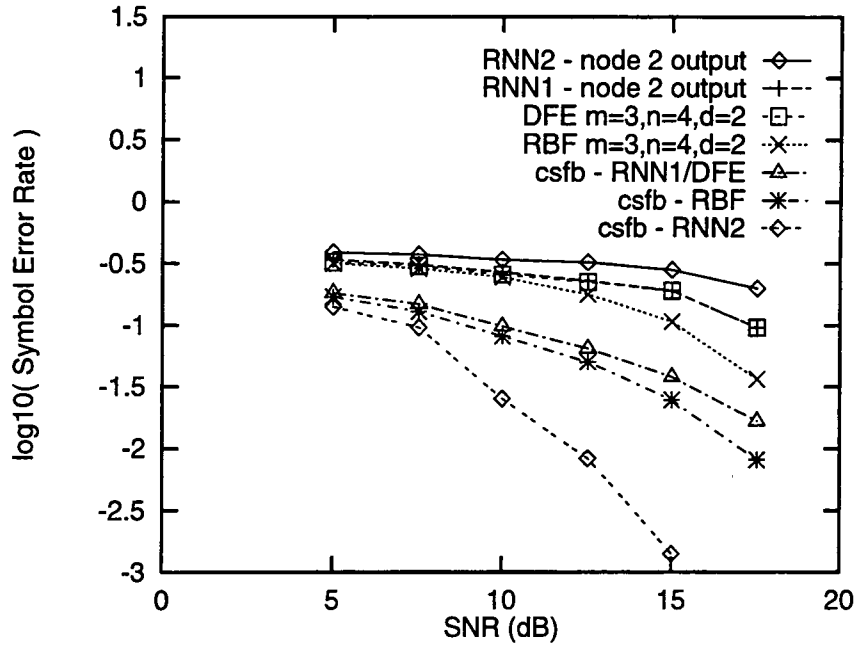

Figure 4.14: SER performance of RNN1, RNN2, DFE and RBF with $d = 2$.

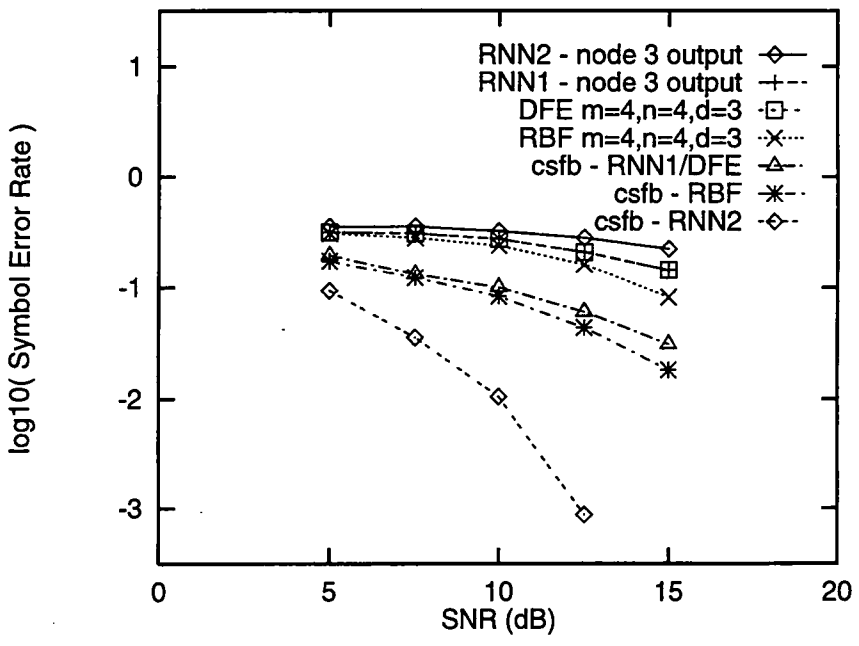

Figure 4.15: SER performance of RNN1, RNN2, DFE and RBF with $d = 3$.

4.5 Summary

The initial simulations have confirmed some of the results in Kechriotis *et al* [45]. For many simple channels the RNN equaliser has been observed to be composed of a single node. The mechanisms by which this single node acheives equalisation have been examined and compared to that of the conventional DFE. The replacement of the hyperbolic tangent nonlinearity by a quantiser is argued in order to improve equaliser performance in the presence of noise and to remove limitations on the types of channels capable of being equalised by the RNN.

The DFE structure created with the quantising nonlinearity has prompted the design of multinode RNNs which use ideas from the DFE in order to cancel precursor and postcursor cancellation of the CIR. The output of each node is specified in order to implement a form of soft decision feedback where by symbol estimates may be updated rather than making hard decisions as in the DFE. These new RNN structures have been compared with DFE and RBF / Bayesian equalisers and are seen to improve over the DFE and DE only when the decision delay is minimal. However, as soon as the decision delay increases the RBF equaliser outperforms the RNNs. The theoretical performance of the 2^{nd} RNN structure is shown to outperform the RBF but practically its performance is degraded due to erroneous decisions.

Given the general poor performance of RNNs over conventional techniques and the more promising RBF equaliser, the RNN structures developed are not considered to be suitable candidates for application to time varying channels. In the next chapter the performance of MLSE equalisers operating on such channels are studied to investigate the mechanisms causing poor detection. Chapter 6 investigates the properties of Bayesian / RBF equalisers and compares them with the MLSE.

Chapter 5

MLSE Techniques for Fading Channel Equalisation

Fading digital communication channels prove to be a harder class of channels to equalise than their time-invariant counterparts. This is particularly so when the channels fade more rapidly as the ISI mechanism must be tracked with time. For land mobile radio, the rate of fading increases with carrier frequency and mobile velocity (see Section 2.2.1) and thus applications such as high speed train communications experience more rapid fading [49]. Such channels are still classified as slow fading channels if the coherence time of the channel is less than the transmitted symbol period [4, 5].

The characteristics of maximum likelihood sequence estimation (MLSE) equalisation techniques are examined in detail in this chapter and the performance limitations and their causes are identified with particular emphasis on the tracking of the time-varying channels. The results obtained provide performance benchmarks with which to compare alternative equalisation techniques and indicate areas in which design improvements are likely to reduce the symbol error rate of the equaliser.

DFE equalisation techniques can also be applied to time-varying channels. They are computationally simpler by the fact that they do not require a channel estimator but their performance is limited by the adaptation algorithm's ability to track the inverse of the channel impulse response. The DFE performance is generally lower than that of the conventional MLSE, especially for more rapidly fading channels and for this reason the DFE has not been chosen as the benchmark algorithm. However, the DFE is still an important equaliser for slowly fading channels due to its simplicity and is used in commerical mobile radio systems. Variants using a channel estimator to provide the DFE coefficients have been investigated in [50, 51] which show an improved performance compared to the direct adaptation methods for the system configurations considered.

5.1 MLSE Equalisation and Channel Estimation

The MLSE technique, the principles of which are given in Chapter 3, is known to give optimal performance in resolving ISI assuming a perfect knowledge of the channel being equalised [2, 5, 33]. For communications via a fading channel, the channel is unknown and time-varying and so a channel estimator is employed to provide an estimate of the channel impulse response which is then used to calculate the transition metrics in the Viterbi algorithm. The optimality of the MLSE technique for such channels is questionable as its performance using erroneous channel estimates is less well known.

The fading nature of the channel requires that regular retraining and/or tracking of the channel estimator occurs in order to prevent a significant loss in performance [52]. The transmitted data is split into frames consisting of a training sequence and a data burst (Fig. 5.1) — a mechanism well suited to TDMA systems which have an inherent frame structure. For instance, regular retraining occurs in the GSM system with 26 symbols out of a normal burst of 148 symbols being dedicated to channel estimation [53]. For slow fade rates (relative to the transmitted data burst length) tracking may not be required as the channel variations over the duration of the data burst are small. If channel tracking is to be used then the MLSE technique is implemented using the Viterbi algorithm with constant decision delay in order to provide symbol estimates for the channel estimator [54]. Otherwise the Viterbi algorithm may be operated 'off-line' on the whole data burst and is not forced to make premature symbol decisions as can happen with a fixed decision delay implementation. In such circumstances the training sequence may be inserted midway through the data burst to reduce the error in the channel estimate at the beginning and end of the data burst. The algorithm's performance may be improved by providing 'head' / preamble and 'tail' / postamble symbols in order to initiate and terminate the Viterbi algorithm in known states. The overhead associated with these additional symbols is generally small for land mobile radio channels - being in the order of three or four symbols per pre-/ postamble for channels with typical multipath delay spreads and system symbol rates.

There are two main techniques of channel estimation: correlation methods and minimum mean square error (MMSE) / least square error methods. Both algorithms generate CIR estimates with T_s-spaced taps representing the energy associated with individual symbols rather than estimating the multipath delays with a higher resolution. The Viterbi algorithm requires a symbol-period spaced CIR to produce estimates of the received signal states during transition metric calculation.

Correlation methods [5] are generally used to obtain an estimate of the CIR which is assumed to be constant for the duration of the transmit data burst. The training sequence is transmitted mid-burst in order to reduce the estimation error at either end of the transmit burst. The length of the transmit burst affects the equaliser performance as the CIR estimate

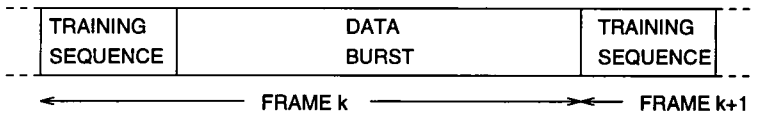

Figure 5.1: Frame structure of transmitted data.

degrades the further from the training sequence that the estimate is used. In [55] a technique is proposed whereby channel estimators obtained during transmission of training sequences are used to generate a succession of CIRs by interpolating over the duration of the data burst. The correlation method involves transmitting a training sequence, $s_T(k)$, and correlating the received signal, $r(k)$, with a replica of the conjugate of the training sequence. The received signal is the convolution of the equivalent baseband CIR, transmit and receive filters and training sequence (corrupted by additive receiver noise and/or co-channel interference, $n(k)$):

$$\begin{aligned} r(k) &= ([s_T(k) \otimes f_T(k)] \otimes h(k) + n(k)) \otimes f_R(k) \\ &= s_T(k) \otimes [f_T(k) \otimes h(k) \otimes f_R(k)] + n(k) \otimes f_R(k) \end{aligned} \tag{5.1}$$

where $f_T(k)$ and $f_R(k)$ are the transmit and receive filter impulse responses and $\otimes$ denotes the convolution operator. The output of the receiver correlator contains the combined CIR convolved with the autocorrelation function of the training sequence and a noise term:

$$\begin{aligned} r(k) \otimes s_T^*(L_T - k) &= s_T(k) \otimes s_T^*(L_T - k) \otimes [f_T(k) \otimes h(k) \otimes f_R(k)] \\ &\quad + s_T^*(L_T - k) \otimes [n(k) \otimes f_R(k)] \end{aligned} \tag{5.2}$$

where L_T is the length of the training sequence. The combined CIR is the filter impulse response resulting from the cascade of the transmit, channel and receive filters. As the length of the training sequence increases, the noise / interference term is reduced under the assumption that it is uncorrelated with the training sequence. The training sequence is chosen to have an autocorrelation function as close to a unit impulse function as possible, resulting in the output of the correlator being approximately the combined CIR. The training sequence autocorrelation function is called the ambiguity function and sequences with ideal charcteristics are termed CAZAC (Constant Amplitude Zero AutoCorrelation) codes or Barker sequences [56].

Minimum MSE channel estimation techniques operate on gradient descent or recursive identification principles [57]. A channel model is assumed (a symbol-period spaced FIR model) and the parameters are updated under the operation of an adaptation algorithm such as the LMS or RLS algorithms (see Appendix B). These techniques have been investigated for a variety of channels [50, 58]. A training sequence is periodically inserted into the data

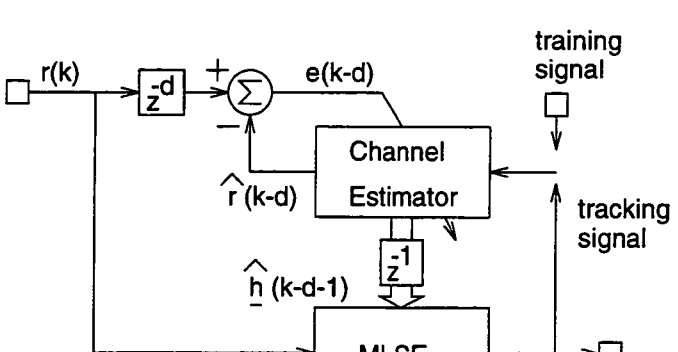

Figure 5.2: MLSE and channel estimator operating in a decision-directed tracking mode.

stream as per the correlation method and the channel estimate is updated by passing a replica of the training sequence through the channel model and using the error between the actual and estimated received signal samples to update the CIR estimate. These estimators may be operated in a decision-directed mode in between the training sequences to allow tracking of the channel. In this configuration, the output of the equaliser is fed back into the channel estimator instead of the transmitted data (Fig. 5.2) — the tracking algorithm normally assumes that these estimated symbols are correct. Under these conditions the channel estimator produces an approximation to an old channel rather than the current channel due to the decision delay of the equaliser. The received signal is delayed prior to being used to form the estimator's error signal due to the fact that, during tracking, delayed symbol estimates are fed into the channel estimator which uses an old combined CIR estimate to form an estimate of an old received signal. The delay of the channel estimate vector, $\underline{\hat{\mathbf{h}}}(k-d)$, by a period of T_s prior to being fed into the Viterbi processor is required during decision-directed tracking to ensure a non-zero delay feedback loop. The sequence of processing operations at time $t = k$ becomes:

1. The Viterbi processor uses $r(k)$ and $\underline{\hat{\mathbf{h}}}(k-d-1)$ to generate the symbol estimate $\hat{x}(k-d)$.

2. The channel estimator uses $\hat{x}(k-d)$ together with previous symbol estimates and $\underline{\hat{\mathbf{h}}}(k-d-1)$ to give an estimate of the delayed received signal, $\hat{r}(k-d)$.

3. The error signal between actual and estimated received signal is used to update the channel estimate, generating $\underline{\hat{\mathbf{h}}}(k-d)$ which is fed back to the Viterbi processor for use at the following sample period.

The use of delayed CIR estimates adversely affects the symbol error rate of the equaliser as shown experimentally later in the chapter. A scheme whereby old symbol estimates are

used to predict the current received signal and the current combined CIR could be envisaged but this is highly dependent on the channel's multipath delay spread, power profile and the equaliser decision delay. For time-invariant channels, the equaliser decision delay is in the order of three to five times the multipath delay spread and so the delayed symbol estimates have no impact on the current received signal and therefore cannot be used in a prediction of the received signal. For fading channels the decision delay is smaller, as will be shown later in the chapter, and the current received signal may contain energy from the old symbols. The quality of the prediction is then dependent on the amount of energy from the old symbols that is present in the current received signal. With a tapered power-delay profile, this energy will be small, on average, leading to poor predictions. This predictive scheme is not considered further due to these problems. Further details of alternative prediction techniques, which extrapolate a sequence of old channel estimates to give a current channel estimate, may be found in [27, 58].

The choice of LMS or RLS algorithms or one of their variants to adapt the channel estimator is dependent on the rate of convergence required and their error propagation characteristics [59, 60, 61]. Short training sequences are desirable to increase the system data rate which prompts the use of fast-convergening RLS algorithms [62]. However, RLS algorithms are not necessarily better than LMS algorithms for channel tracking purposes and have been reported to suffer when applied to decision-directed tracking where incorrect symbol feedback causes error propagation [2, 63]. Least square channel estimation techniques [47, 64] are similar to the MMSE technique being derived by minimising a suitable error function. However they do not employ recursive adaptation algorithms.

The relative advantages of each estimation technique depend upon the channel being equalised. For 'slowly' fading channels the correlation based method is sufficient given a suitably long training sequence whilst for faster fading channels the ability to operate the channel estimator in a decision-directed mode is often desirable. A combination of the techniques is possible but increases system complexity. The difference between slower and faster fading channels is measured as the average number of symbols transmitted between fades although this not universally adopted. This average is a function of the transmitted symbol rate and the fade rate / Doppler spread of the channel. However, the channels dealt with are all *slow* fading rather than *fast* fading as the coherence time of the channel is much less than the symbol period ($T_c \ll T_s$) — see Section 2.2.1 and [4, 5].

Variants on the standard MLSE technique include a version having two decision delays. A short decision delay is used to derive symbol estimates for decision-directed channel estimation and a second, longer, decision delay is used to give final symbol estimates for channel decoding. Other techniques use an adaptive DFE prior to the MLSE block in order to partially remove ISI [65]. This allows a reduction in the number of states in the Viterbi trellis and provides a

mechanism for symbol decisions in the DFE's feedback delay line to be updated.

5.2 System and Channel Models

The details of the system and channel models used to derive the simulation results characterising the MLSE equaliser are explained in this section. There are a large number of variables which may be adjusted to maximise the absolute performance of the equaliser and a number of them are held constant throughout in order to limit the simulation process. The choice of which variables to hold constant is made so that the results are not necessarily limited to any one application system. The relative effects of changes in parameters are considered to be of greater importance than the absolute performance as the latter is dependent on many system factors outside the scope of this thesis. For instance, coherent demodulation is assumed throughout and the effects of imperfect carrier and symbol timing synchronisation are not considered in detail. However, the sensitivity of the results to a number of other system variables on the equalisation process are discussed in a qualitative manner.

5.2.1 System model

The digital radio communication system as outlined in Figure 2.1 is simplified in the simulations presented in this chapter. No source or channel coding is assumed and instead, a sequence of statistically independent bits is passed into the modulator. Appropriately, no source or channel decoding is required in the radio receiver and the error rates are measured at the output of the equaliser and detector.

The modulator is one of the system variables that is held constant and all simulations in this and the remaining chapters (except for a few isolated cases) use a baseband QPSK / QAM modulator with a simulated symbol rate of 100kBaud. The mutually independent input bits are formed into complex symbols using Gray coding:

$a(2k)$	$a(2k-1)$	$x(k)$
0	0	+1+j
0	1	-1+j
1	1	-1-j
1	0	+1-j

where $a(k)$ are the input bits and $x(k)$ the modulated symbols. The most common error at the receiver detector output is between adjacent symbols and Gray coding ensures that adjacent symbols only differ by a single bit thereby reducing the bit error rate. After the modulation, which uses one sample per symbol, the symbols are passed through a root-raised cosine filter with a roll-off factor $\alpha = 0.5$, sampled at a higher rate, to band limit the modulated signal

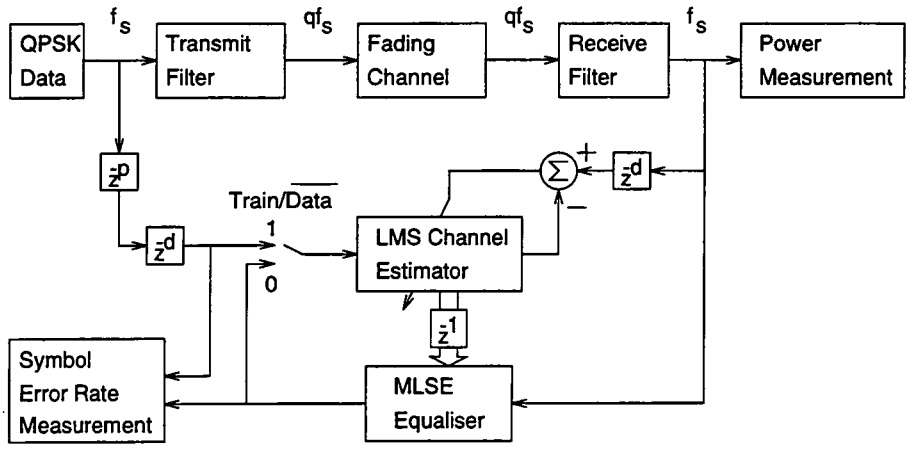

Figure 5.3: System simulation configuration.

prior to transmission through the channel (Fig. 5.3). The filter is implemented as an FIR filter with the impulse response truncated to 64 samples and separated in time by $\frac{T_s}{q}$, where q is the upsampling factor. The upsampling factor is used to simulate the continuous time domain and is necessary to allow channnels having path delays with non-integer multiples of the symbol period, T_s, to be simulated. The upsampling factor used in the simulations varies from either $q = 1$ or $q = 4$ depending upon the channel being simulated.

The receiver is modelled by adding complex white Gaussian noise to the received signal and filtering the resulting signal to remove out-of-band noise. Ideally this receive filter is matched to the cascade of the transmit filter and channel impulse response and the output downsampled to the symbol rate. The sampled signal should then pass through a noise whitening filter to remove the correlations in the additive noise caused by the receive filtering operation [2, 33]. However, as the channel is unknown and potentially rapidly time varying, the matched filter technique is not easy to implement, if at all, and the receive filter has been fixed as a root-raised cosine with a roll-off factor $\alpha = 0.5$, matching the transmit filter. In the absence of channel induced ISI the transmit and receive filter cascade form a raised cosine filter which satisfies Nyquist's criterion for zero ISI pulse shaping. After filtering the signal is downsampled to the symbol rate (assuming perfect sampler timing) and fed to the equaliser without passing through a noise whitening filter.

The channel estimator has been implemented as an FIR filter adapted using the LMS algorithm. Since the inputs of the channel estimator, $x(k-d)$ or $\widehat{x}(k-d)$, have constant magnitude, the standard LMS algorithm is equivalent to the normalised version (see Appendix B). The LMS algorithm was chosen as a fixed parameter due to its ease of implementation and performance qualities for decision-directed tracking. Since the simulations are concerned, in part, with faster fading channels, the correlation type estimation techniques have not been

considered as tracking is required in order to obtain reasonable bit error rates (see the simulation results for confirmation). The qualatative speed of fading is relative to the data burst duration rather than the symbol period. The estimator is operated in a decision-directed tracking mode as shown in Figure 5.2.

The frame structure of the data has been varied in the simulations to investigate the perfomance changes for increases in data burst length and for variations in the ratio of data symbols to training symbols. Typical data burst lengths are in the order of 100 symbols and training sequences in the order of 20 symbols giving a ratio of 5:1. Prior to symbol error rate measurement, the channel estimator is trained using 500 symbols in order to acquire a reasonable estimate of the channel. Subsequently the estimator is either operated in a decision-directed mode or uses the training sequence for CIR estimation. The channel estimator is not reset at the beginning of each training sequence but is adapted from the CIR estimate at the end of the previous data burst. This method allows the LMS algorithm to form a resonable CIR estimate by the end of the training sequence. It is noted that this scheme is at odds with a mobile radio application employing a TDMA format. The base station receiver will not receive signals from the same transmitter in consecutive frames and thus the CIR between frames will be significantly different due to the transmitters being located in different geopraphical positions. In the case of a mobile unit, the CIR will have changed radically from that during the previous transmit slot due to mobile motion. These factors require that the channel estimate acquires a completely new CIR estimate for each frame / time slot. However, the aim of the simulations is to investigate the characteristics of the MLSE equaliser rather than the acquisition properties of specific channel estimation techniques. Results confirming that the performance limitations observed are not due to acquisition problems during training sequence transmission are presented.

The training sequence used to adapt the channel estimator is not unique. Different sequences have been generated from the the output of the modulator rather than using CAZAC sequences. Thus it is possible for isolated frames to have training sequences that are not particularly suited to the LMS algorithm (i.e. sequences with a large eigenvalue spread [47]), but experimental results show that this does not have a significant effect. The pseudo-random symbol sequence from the output of the transmitter modulator is partioned into data and training sequences. The training sequences are fed to the channel estimator in the receiver during training phases after being delayed a period of time conditional on the propagation delay of the channel, transmit and receive filter delay and the decision delay of the equaliser.

A separate MLSE equaliser has been implemented using the Viterbi algorithm in response to limitations of the Viterbi block provided in the SPW library. The equaliser permits flexibility in system parameters such as the number of states in each stage of the trellis and the decision delay and number of stages in the truncated trellis. At each iteration of the algorithm the path metrics are limited in size by subtracting the minimum metric of the whole trellis

at the previous iteration from each survivor's transition metric. Paths with large metrics will gradually be replaced by those with smaller metrics and thereby path metric growth does not affect the simulations. This point is more appropriate to practical implementations with limited precision / word size but can be problematic in simulations running over large numbers of symbols. The Viterbi processor is left running during channel estimator training sequences and is not reset at the beginning of each frame. Pre / postamble symbols have not been incorporated into the frame structure and MLSE implementation (which would otherwise affect transition metric calculations) as they do not significantly affect the relative performance between the MLSE and variants presented. However, it is noted that their use can significantly reduce the absolute symbol error rates especially at lower error rates.

Throughout the simulations, symbol errors are measured rather than bit errors. Thus the benefits of Gray coding the bit to symbol mapping are not seen directly in the results. At low symbol error rates the errors can be assumed to be between adjacent symbols and thus the bit error rate will be half than that of the symbol error rate. The symbol error rate is defined as the total number of errors measured divided by the total number of data symbols transmitted. Error measurement begins at the start and finishes at the end of each data burst and so the error rate is averaged over a number of frames. The frame structure of the transmitted symbols requires that the equaliser, channel estimator and error rate measuring block to be synchronised. For equaliser decision delays, $d > 0$, commencing the error rate measurement at the beginning of the data burst means that the first d symbol estimates will (nominaly) be the end of the training sequence. Similarily at the end of the data burst, suspending error rate measurement at the end of the data burst means the last d data symbol estimates will 'not' be tested for veracity. However, the partioning between training and data symbols is only nominal due to the fact that the MLSE block does not use the training sequence as preamble and postamble symbols. The switch controlling the flow of training symbols or equaliser symbol estimates into the channel estimator for decision-directed tracking is operated at the nominal training / data boundary but as both are delayed in time by d symbol periods the effect is to shift the boundary back in time by dT_s. Thus varying the decision delay does not affect the error measurement timing or synchronisation of the Viterbi algorithm and channel estimator — the positioning of the boundary between training and data symbols is merely interpreted in a slightly different manner.

In order to be confident in the error rate measurements, the number of data symbols transmitted needs to be large. Precise confidence limits on the error rates is difficult as the error mechanism is not statistical independent of previous errors, especially during decision-directed tracking where erroneous symbol estimates can cause the channel estimate to diverge from the true CIR. Many of the error rates measured have been checked for variations by simulating identical systems and varying the seed in the psuedo-random number generator. Gross variations have not been observed when the number of symbols transmitted is in the

region of 150000. An additional consideration affecting the reliability of the results is the fade rate of the channel. For slowly varying channels more symbols need to be transmitted for similar confidence levels as it takes a longer period for the characterisitics of the channel to average out (e.g. the instaneous gain of the channel and maximum / minimum phase properties).

The SPW package does not guarantee that the same noise or channel tap sequences (generated from the 'master' psuedo-random source) are used in each simulation despite the same random seed. Due to the many time-coincident noise sources used, the sequence in which they are called cannot be dictated in advance and is affected by the number and type of blocks used in the simulation. Given *precisely* the same simulation model, the noise sequences are identical but changing elements in the receiver, for instance, where random sources are not used can affect the noise sequence and results obtained. Thus results obtained from different simulation runs must be compared with care. The majority of results presented on the same graphs have been obtained by simply varying parameters of a simulation which do not affect the sequencing of the noise sources and thus give reliable comparative results. Comparison of results between simulations are only significant if their confidence limits do not overlap but, as the definition and calculation of the limits are subject to interpretation, caution must be excercised when comparing results lying close to each other. To isolate this problem the transmitter, channel and receiver AWGN could be simulated in isolation and the results stored and used to drive the remaining receiver circuitry but this has a large memory / storage overhead which has prevented its use.

The signal-to-noise ratio (SNR) is defined as the average signal power divided by the average noise power measured immediately prior to the equaliser. The transmitted data has a power of 2 (real and complex paths each have unity power) and the transmit and receiver filters each have a power gain of $\left(\frac{1}{q}\right)$ where q is the upsampling / downsampling factor and the channel is designed to have an average power gain of $G_c = 1$. Due to variations in the channel power gain as the channel fades, the average received signal plus noise power, P_{S+N} is measured prior to the equaliser over the duration of the simulation. The SNR is derived from this measure given the known receiver AWGN noise power, $2\sigma_n^2$ and receive filter power gain:

$$\begin{aligned} P_{S+N} &= P_S + P_N \\ &= \frac{2G_c}{q^2} + \frac{2\sigma_n^2}{q} \end{aligned} \tag{5.3}$$

$$\begin{aligned} \Rightarrow \text{SNR} = \frac{P_S}{P_N} &= \frac{G_c}{q\sigma_n^2} \\ &= \left(\frac{qP_{S+N}}{2\sigma_n^2} - 1\right) \end{aligned} \tag{5.4}$$

5.2.2 Channel models

The channels are modelled with zero propagation delay for simplicity and with delays equal to an integer number of sample periods, $\frac{T_s}{q}$ — thus the upsampling factor q affects the resolution of the channel's multipath delays. The basic channel model is shown in Figure 2.6 with complex Gaussian white noise being passed through a Doppler spread filter producing the path's gain and phase shift in complex format. The Doppler spread filters have real impulse responses leading to symmetrical sidebands. An additional block is used in the simulations between the Doppler spread filters and the multipliers for simulation speed improvements. As the fade rate, fd, is significantly smaller than the sample rate of the channel — $f_d \approx$ 100Hz and $\frac{q}{T_s} \approx$ 400kHz, generating the taps at the channel sample rate requires a Doppler spread filter with an extremely narrow passband and results in huge numbers of filter taps. To avoid this problem, the noise block and Doppler spread filter block are sampled at 1kHz and upsampled by a factor of ≈ 400 to the channel sample rate by means of an interpolator. The interpolator is implemented as a polyphase filter with ≈ 400 branches, each having a lowpass FIR filter of 25 taps (efficiently implementing a 10000 tap filter) and with a cut off frequency of 500Hz [66]. The power of the noise block is set so that the sum of the noise powers over all paths is 0.5 (the Doppler spread filters have unity power gain). As the paths have complex taps, this gives a channel power gain of unity.

5.3 MLSE Characterisation

The effects of varying a number of the MLSE equaliser's parameters are shown in the simulations results presented in this section. The primary results show the effects of channel estimation errors on the symbol error rate (SER), especially when the equaliser is operated in a decision-directed tracking mode where an approximation to an old CIR is formed. The affects of incorrect symbol decisions on the tracking algorithm are shown together with the impact of varying the decision delay, the LMS algorithm's update paramter and the length and ratio of the training sequences and data bursts. A number of different channels are simulated with various fade rates, multipath delay spreads, power-delay profiles and Doppler spread characteristics.

5.3.1 Channel Estimation Effects

A three equipower path channel ($L = 2$) with path delays equal to the symbol period, T_s, has been simulated in order to observe the general nature of the MLSE equaliser whilst oparating with a number of channel estimator configurations. A T_S-spaced channel was chosen as it provides a relatively easy situation for the channel estimator which generates a T_s-spaced model. As perfect symbol-sampling synchronisation is assumed, the additional ISI caused by

the transmit and receive root-raised cosine filters at the sampling instant is deemed small and so the transmit / receive filters are not included in the simulation. This provides an idealised situation in which the additive noise is uncorrelated but enables the characteristics of the equalisation algorithm to be examined in favourable conditions and allows shorter simulation run times. The frame structure was chosen as 100:20 (data burst length : training sequence length). The channel paths had real and imaginary powers { 0.16667, 0.16667, 0.16667 } giving a channel power gain, $G_c = 1.0$ and had a classical Doppler spectrum with maximum Doppler spread of ± 200Hz. The channel estimator used a perfect estimate of the number of paths (i.e. a model of three T_s-spaced taps) and was updated by the LMS algorithm with a stepsize, $\mu = 0.1$ (see Section 5.3.4). The MLSE equaliser had a decision delay of five symbol periods. These parameters are summarised as follows:

Channel parameters		
L	Path delay	Doppler spread
2	T_s	Classical, $f_d = \pm 200$Hz

System parameters				
Tx./Rx. filters	Frame structure	Decision delay	Channel estimator taps	μ
none	100:20	5	3	0.1

The channel estimator configurations are:

1. No tracking. The channel estimate is held constant after the training phase for the whole duration of the subsequent data burst to observe the effects of not tracking the channel.

2. Decision-directed tracking. The channel is tracked during the data burst in the standard configuration.

3. True data tracking. The channel estimator uses a perfect knowledge of the transmitted data during tracking in the data burst. This simulates the effects of erroneous symbol decisions produced by the equaliser / detector on the tracking algorithm.

4. Undelayed true data tracking. Perfect undelayed symbol estimates are used during data burst tracking to observe the effects of generating out-of-date channel estimates.

5. Delayed true CIR. A delayed version of the true CIR is fed to the MLSE equaliser to demonstrate the effects of imperfect channel estimation.

6. True CIR. The true CIR is used by the equaliser to show the performance limit of the MLSE algorithm.

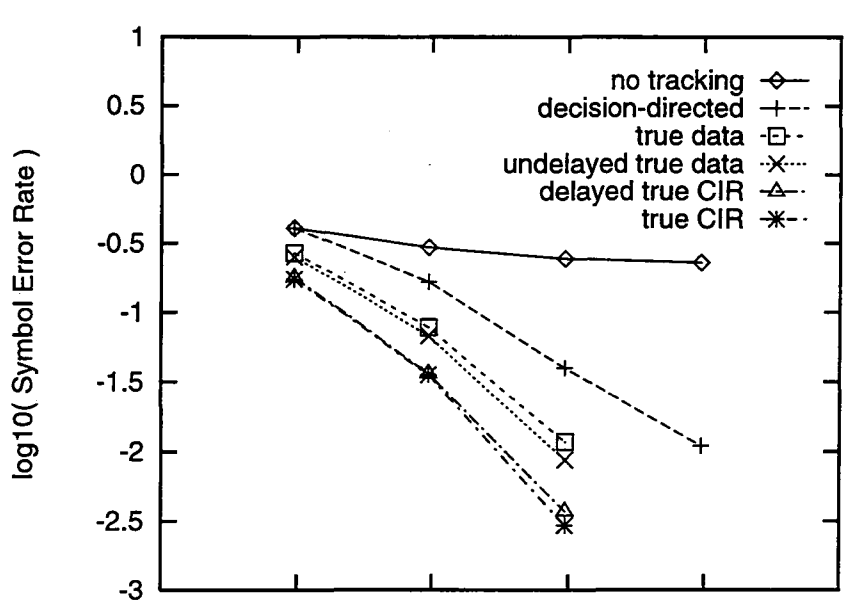

Figure 5.4: MLSE performance with varying channel estimator modes. $L = 2$, T_s-spaced equipower paths, $f_d = \pm 200$Hz.

Figure 5.4 shows the SER vs. SNR for these channel estimator configurations. The error rates were measured over 158760 transmitted data symbols. The benefits of tracking the channel through the data burst are clear and the SER degradation due to incorrect symbol estimates is significant. Relatively small performance improvements are obtained due to the channel estimator delay for the fade rate used but, as the Doppler spread increases, this delay becomes more significant (see Section 5.3.5).

Figure 5.5 shows the normalised mean tap error (MTE) increase during the data burst for the no tracking, decision-directed tracking and true data tracking configurations with an SNR=15dB. The normalised MTE is defined as:

$$\frac{\sum_{i=0}^{L} |h_i(k) - \widehat{h}_i(k)|^2}{\sum_{i=0}^{L} |h_i(k)|^2} \tag{5.5}$$

averaged over 1587 data bursts. This definition is similar to that in [67, 68]. The true data tracking curve shows that the MTE remains constant for the duration of the burst indicating that poor acquisition during the training sequence is not the cause of the higher symbol error rates in other channel estimator configurations. This is confirmed in Figure 5.6 which shows the mean number of errors as a function of symbol position within the data burst. As most errors occur midway onwards in the frame, the use of preamble and postamble sequences will not dramatically affect the error rates.

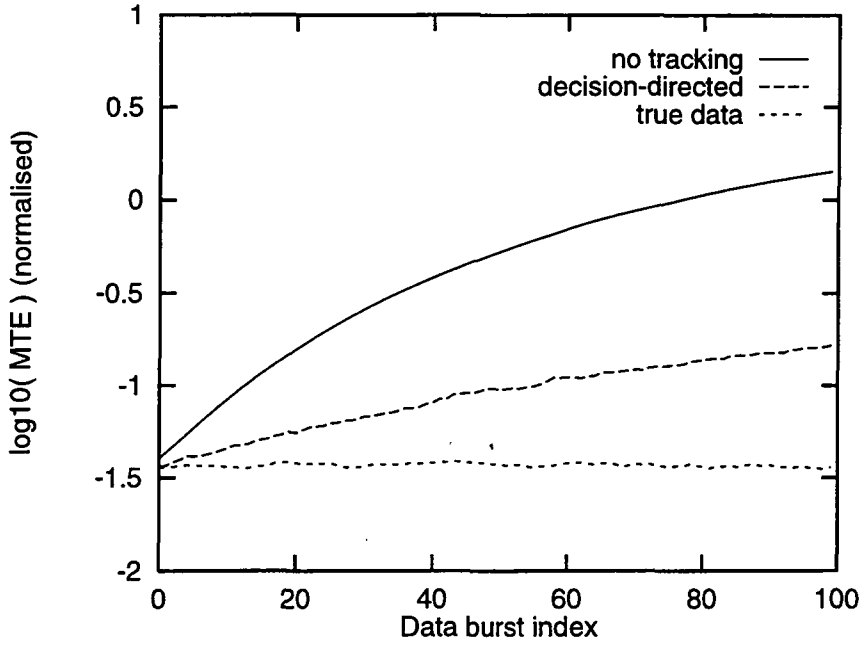

Figure 5.5: Normalised mean tap error vs. data burst index. $L = 2$, T_s-spaced equipower paths, $f_d = \pm 200$Hz.

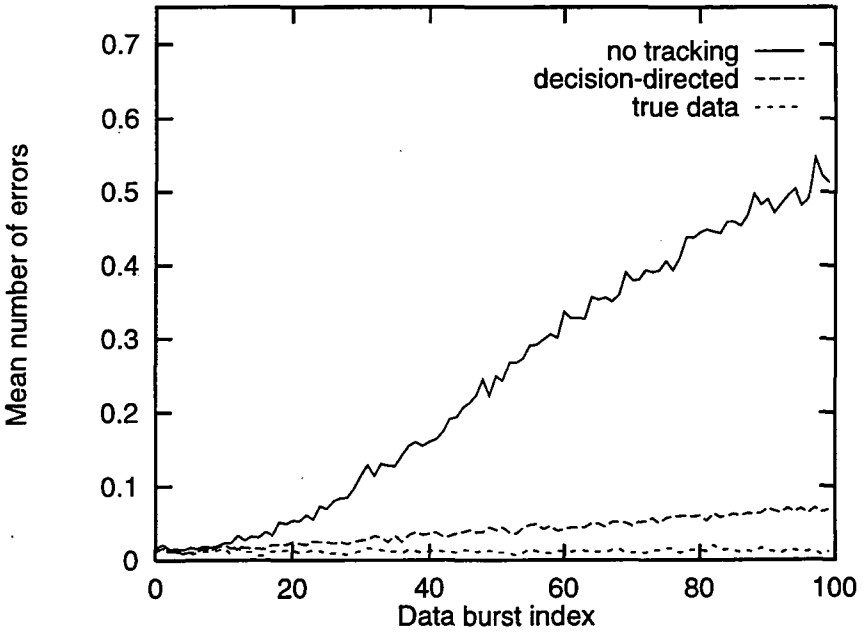

Figure 5.6: Mean number of errors vs. data burst index. $L = 2$, T_s-spaced equipower paths, $f_d = \pm 200$Hz.

5.3.2 Fractionally Spaced Multipath Delay Effects

The performance of the MLSE equaliser on a fractionally spaced channel are examined by simulating a 4 $\frac{T_s}{2}$-period spaced equipower path channel. Each path had real and imaginary path powers of 0.125 giving a channel power gain of, $G_c = 1.0$ and the paths faded with a classical Doppler spectrum and maximum Doppler spread of ± 200Hz. The channel was simulated at $q = 4$ times the symbol rate (100kBaud) and the transmit / receive filters were root-raised cosines (RRC) with $\alpha = 0.5$. A 3 T_s-spaced tap channel estimator was used with an LMS stepsize of $\mu = 0.1$ and the MLSE operated with a decision delay of 5 T_s and the simulation used a frame structure of 100:20 as in Section 5.3.1.

Channel parameters		
L	Path delay	Doppler spread
3	$T_s/2$	Classical, $f_d = \pm 200$Hz

System parameters				
Tx./Rx. filters	Frame structure	Decision delay	Channel estimator taps	μ
RRC $\alpha = 0.5$	100:20	3	3	0.1

Figure 5.7 shows the results obtained with the estimator operating in the various configurations as explained in Section 5.3.1. The calculation of the true T_s-spaced channel impulse response is derived in Appendix C. A straightforward convolution of the transmit, channel and receive filter impulse responses is not possible as the channel is time varying. The calculation generates the components of the received signal (prior to the addition of the receiver noise) in terms of the gain and phase shift (in complex format) of each transmitted symbol affecting the received signal. The sequence of coefficients was truncated to three adjacent terms corresponding to a 3 T_s-spaced tap channel estimate with the first term being the offset from the start of the sequence by a period equal to the sum of the transmit and receive filter delay in order to provide a CIR length consistent with the other simulation configurations. This method of truncation does not provide the optimum performance of the MLSE as the received signal is affected by energy from additional symbols — the effect of non symbol-spaced channels and receive filters is to spread the energy of the effective CIR, increasing the delay spread. However the simulation result is included as it provides an indication of the performance obtained with a better channel estimate.

The results show that the limitations in the standard configuration still exist in this more realistic system and channel model. The cause of the limitations is independent of the presence of transimit and receive filters.

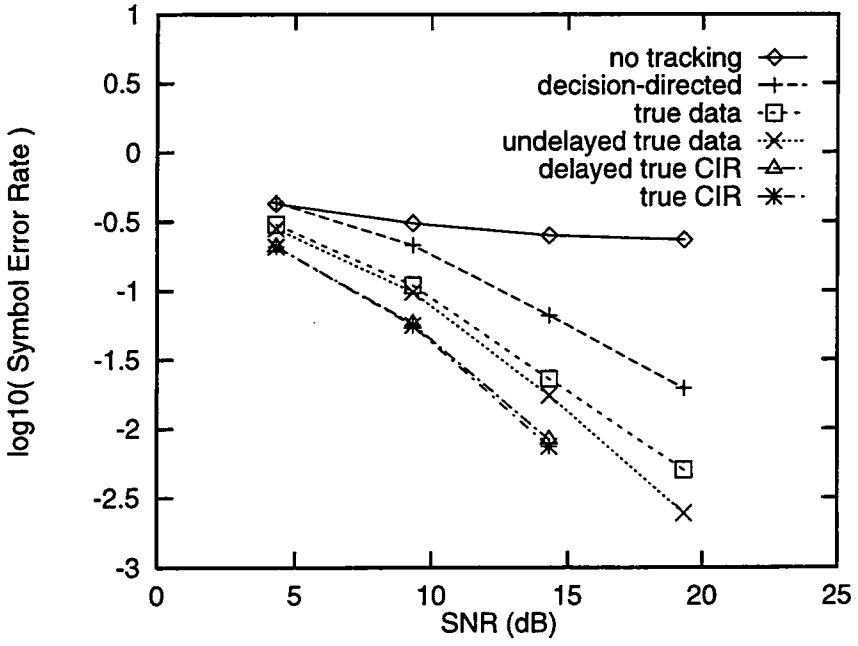

Figure 5.7: MLSE performance with varying channel estimator modes. $L = 3$, $T_s/2$-spaced equipower paths, $f_d = \pm 200$Hz.

5.3.3 Multipath Diversity Effects

The results from Sections 5.3.1 and 5.3.2 indicate that similar limitations in the MLSE equaliser due to channel estimation occur in both fractionally and symbol-period delayed paths. The effect of varying multipath delay spread is simulated by comparing results from 1, 2, 3, 4 and 5 T_s-spaced equipower path channels. The channel path powers were normalised to unity power gain.

Channel parameters		
L	Path delay	Doppler spread
0,1,2,3,4	T_s	Classical, $f_d = \pm 200$Hz

System parameters				
Tx./Rx. filters	Frame structure	Decision delay	Channel estimator taps	μ
none	100:20	0,2,5,5,5	1,2,3,4,5	0.1

The equaliser decision delays were chosen to give the best symbol error rates and the channel estimators were operated in a decison-directed tracking mode using correct estimates of the number of paths.

Figure 5.8 demonstrates the effects of multipath diversity with the SER for SNRs in the range 5—20dB improving and then decreasing with the number of paths. It is noted that these results were obtained from different simulation runs and so statistical variations must be taken into account when comparing the relative performance. The single path channel ($L = 0$) performs poorly due to the fact that when a fade occurs there are no other paths providing signal power. Given that the estimator can estimate the gain and phase shift of each path to a certain degree, the problems of a loss in signal power (i.e. the instantaneous SNR drops) outweigh any degradation due to ISI. This balance depends on how well the channel estimator can approximate the true CIR — see Section 5.3.5 for results with lower quality channel estimates. Fades in the received signal power become less probable as the number of paths increase but further inceases do not reduce this probability significantly and the degradations due to ISI and channel estimation predominate.

For SNRs above 20dB, the SERs of channels with more than one path are significantly smaller. The results have not been plotted as the SERs where too low to be have confidence in them.

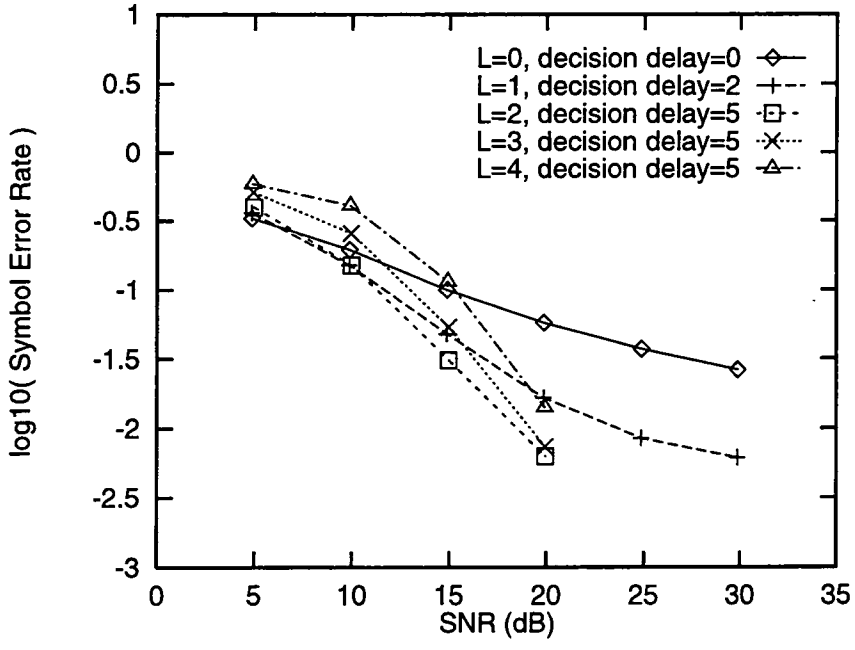

Figure 5.8: MLSE performance with multipath diversity. All channels have T_s-spaced equipower paths and $f_d = \pm 200$Hz.

5.3.4 LMS Stepsize Parameter Effects

The absolute performance of the equalisers has been shown to be dependent on the channel estimator tracking performance. The effects of the LMS stepsize parameter, μ, on the symbol error rate are marked and, in the simulations results given above, the value of μ giving the most consistent results was found prior to the start of the simulations. This results presented in this section (Fig. 5.9) demonstrate the relative effects of varying μ on the symbol error rate vs. SNR for a 2 T_s-spaced equipower path channel. The channel estimator was operated in the decision-directed mode.

Channel parameters		
L	Path delay	Doppler spread
1	T_s	Classical, $f_d = \pm 200$Hz

System parameters				
Tx./Rx. filters	Frame structure	Decision delay	Channel estimator taps	μ
none	100:20	2	2	variable

It is noted that a single value of μ does not give the best results for all SNR values — higher values of μ improve the symbol error rate in high SNR conditions but lower values of μ are better in the range of 10 — 15dB. The results are similar for channels with longer multipath delay spreads and tend to have an optimum value of μ — higher values lead to noisier channel estimates and lower values to channel estimator lag [47], both of which give rise to higher symbol error rates. Channels with faster fade rates generally perform better with slightly higher values of μ — see Section 5.3.5.

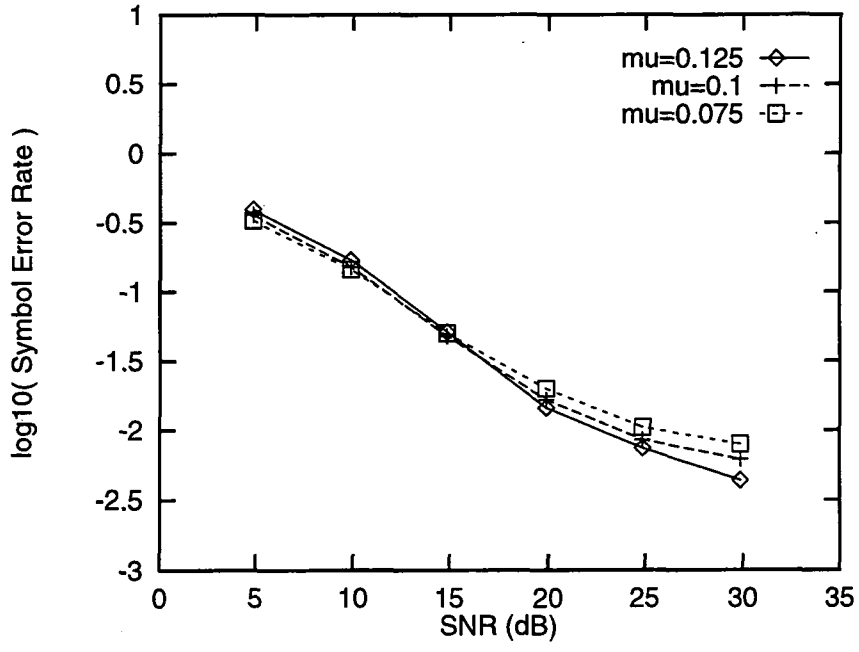

Figure 5.9: MLSE performance with varying LMS stepsize parameter, μ. $L = 1$, T_s-spaced equipower paths, $f_d = \pm 200$Hz.

5.3.5 Fade Rate Effects

The effect of fade rate on the symbol error rate is significant. The fade 'rate' is given here as the maximum Doppler shift, f_d of the classical Doppler spectrum. The level crossing rate, LCR, (i.e. the rate at which a path's amplitude drops below a specfic level [4]) could be used but, for fading paths with Rayleigh amplitude distributions, $f_d \propto$ LCR [4]. The simulation configurations are shown below for the equipower path channels tracked with estimators operating in the decision-directed mode.

Channel parameters		
L	Path delay	Doppler spread
2,3	T_s	Classical, $f_d = \pm 100$Hz $= \pm 200$Hz $= \pm 400$Hz

System parameters				
Tx./Rx. filters	Frame structure	Decision delay	Channel estimator taps	μ
none	100:20	see figures	3,4	0.1

Figures 5.10 and 5.11 show the results for the 3 and 4 equipower path channels respectively and indicate the decision delays used for each curve. These delays were found to produce better results — see Section 5.3.6. For both channels the effects of faster fading is to degrade the symbol error rates. This is due to deteriation in the tracking capabilities of the channel estimator as can be seen in Figure 5.12 which shows the symbol error rates for the various channel estimator configurations. This last result was obtained from the 3 T_s-spaced equipower path channel with maximum Doppler spread of $\pm$400Hz.

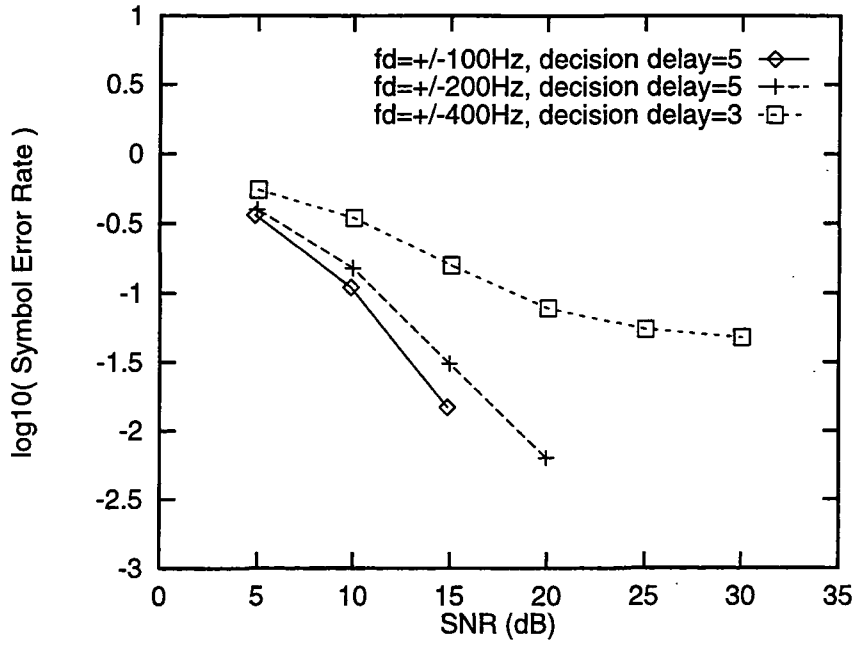

Figure 5.10: MLSE performance with varying Doppler spread, f_d. $L = 2$, T_s-spaced equipower paths, decision-directed tracking.

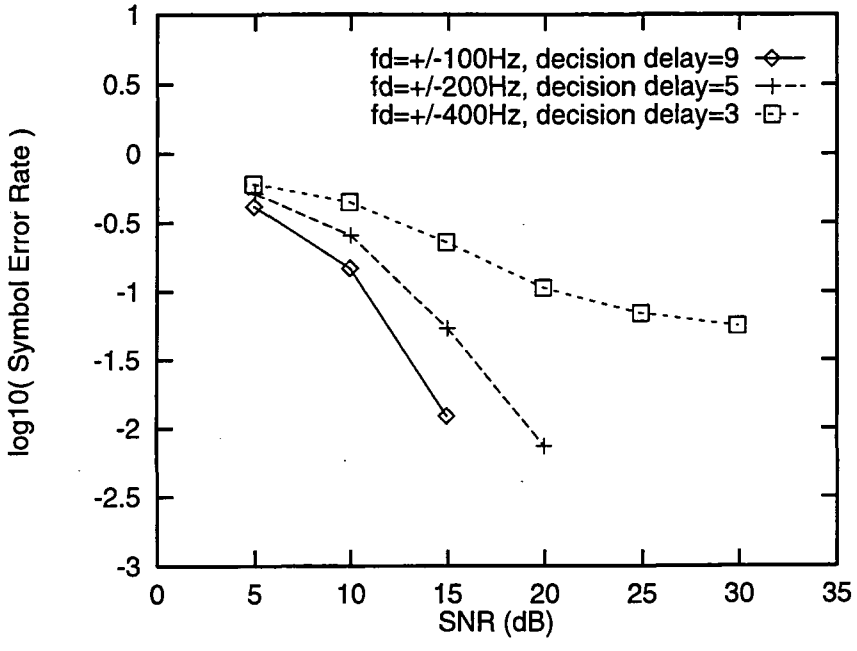

Figure 5.11: MLSE performance with varying Doppler spread, f_d. $L = 3$, T_s-spaced equipower paths, decision-directed tracking.

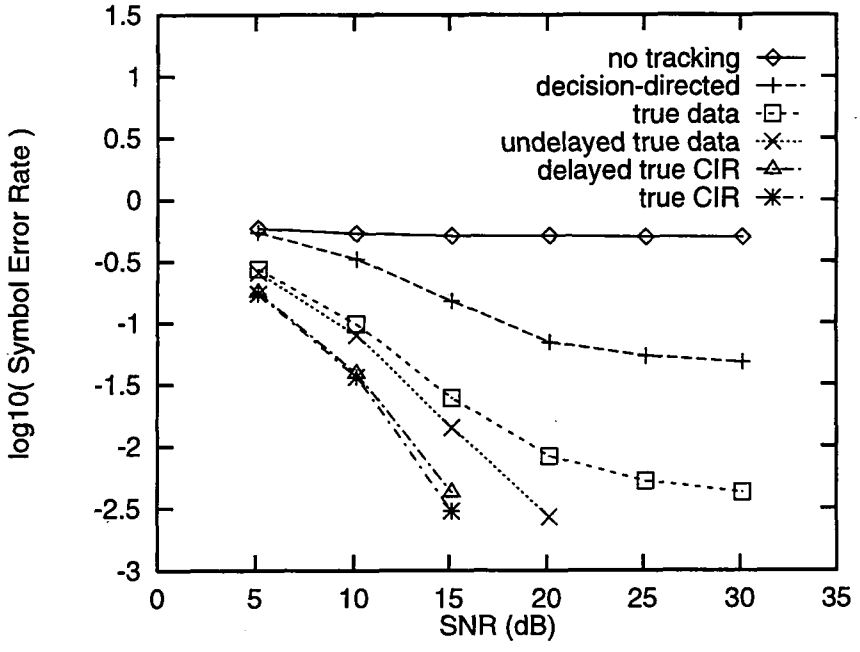

Figure 5.12: MLSE performance with varying channel estimator modes. $L = 2$, T_s-spaced equipower paths, $f_d = \pm 400$Hz.

5.3.6 Decision Delay Effects

Many simulations in the preceeding sections have used a specific value of decision delay in the Viterbi algorithm. This section examines the effect of varying this parameter on the symbol error rates for a number of equipower path channels and fade rates. Additionally, the performance of a dual decision delay MLSE (DDD-MLSE) is simulated. This MLSE variant has a shorter decison delay for the symbol estimates used for decision directed tracking and a longer delay for final symbol estimates which are used for symbol error rate calculation. The algorithm operates in a very similar manner to the standard MLSE except that two symbol estimates are generated from the survivor with the minimum path metric. The simulation configurations are given in the following table.

Channel parameters		
L	Path delay	Doppler spread
1,2,3	T_s	Classical, $f_d = \pm 100$Hz $= \pm 200$Hz $= \pm 400$Hz

System parameters				
Tx./Rx. filters	Frame structure	Decision delay	Channel estimator taps	μ
none	100:20	variable	2,3,4	0.1

Figures 5.13, 5.14 and 5.15 show results for $L = 1$, 2 and 3 respectively for an SNR=15dB. The optimum decision delay for the decision-directed tracking is seen to depend on the fade rate and the CIR length parameter, L. Figure 5.16 shows the effects of the DDD-MLSE algorithm on the $L = 2$ channel with $f_d = \pm 100$Hz, ± 200Hz and ± 400Hz. The channel estimator decision delays are taken from the previous results as those which gave the lowest symbol error rate (i.e. 5, 5 and 3 for the three fade rates respectively). The additional performance improvement is seen to be very small indeed.

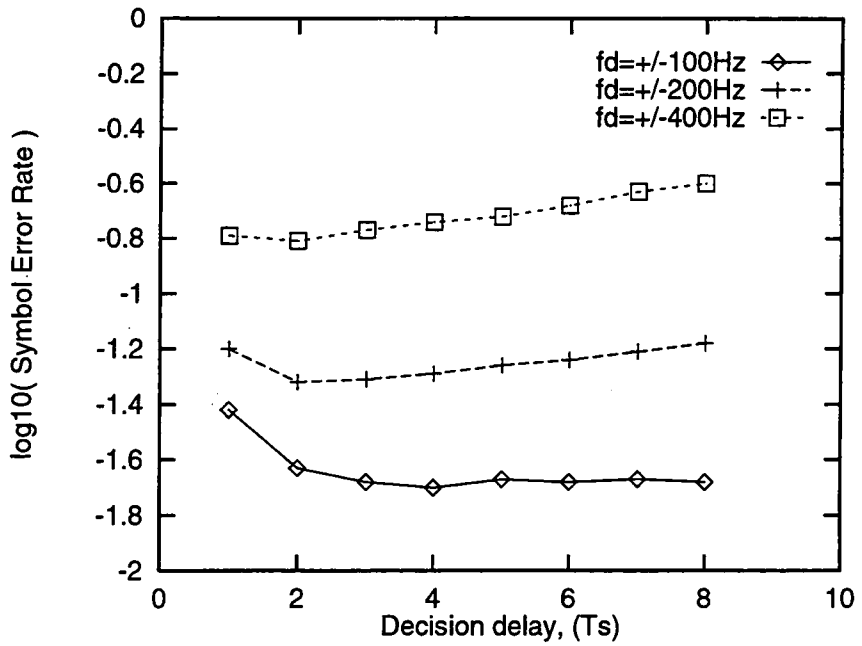

Figure 5.13: MLSE performance with varying decision delay. $L = 1$, T_s-spaced equipower paths, SNR=15dB.

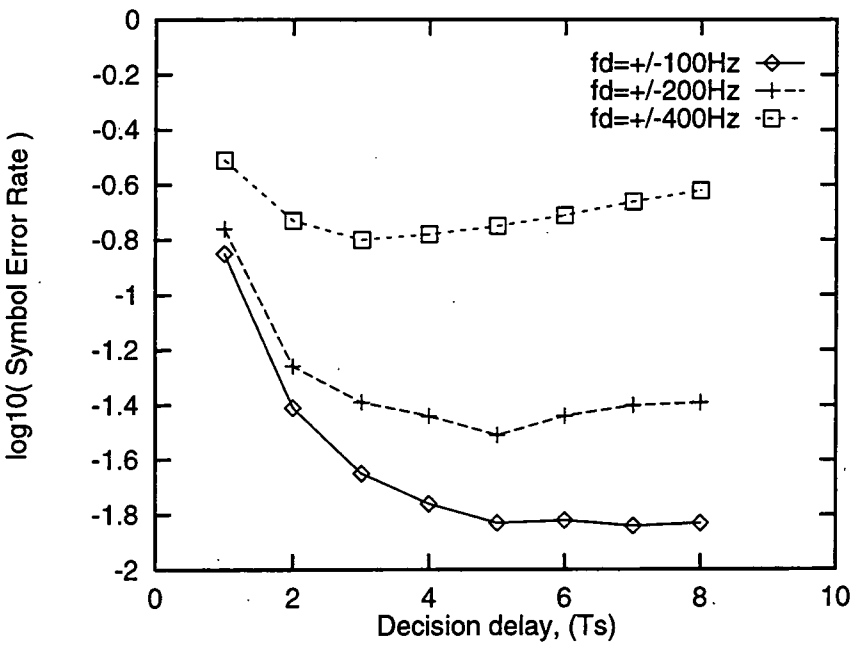

Figure 5.14: MLSE performance with varying decision delay. $L = 2$, T_s-spaced equipower paths, SNR=15dB.

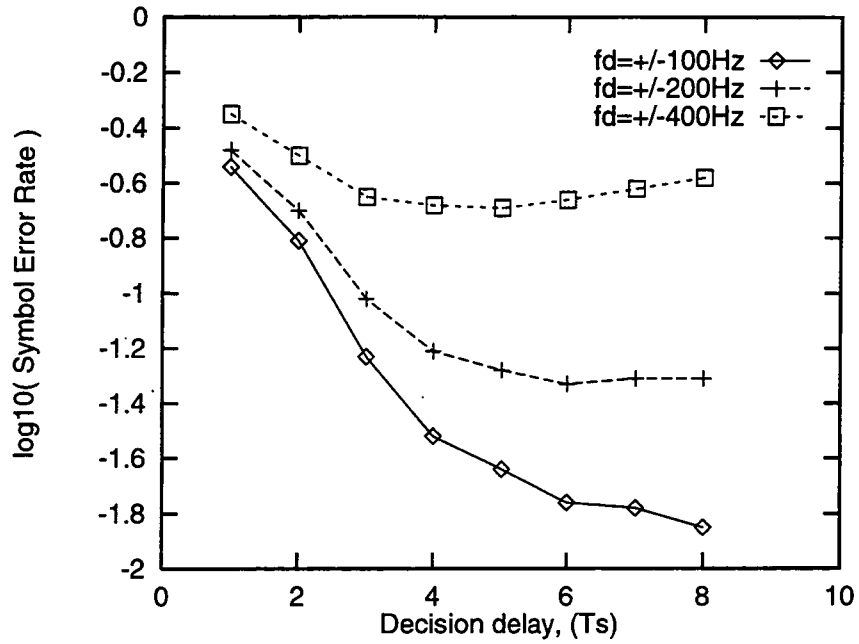

Figure 5.15: MLSE performance with varying decision delay. $L = 3$, T_s-spaced equipower paths, SNR=15dB.

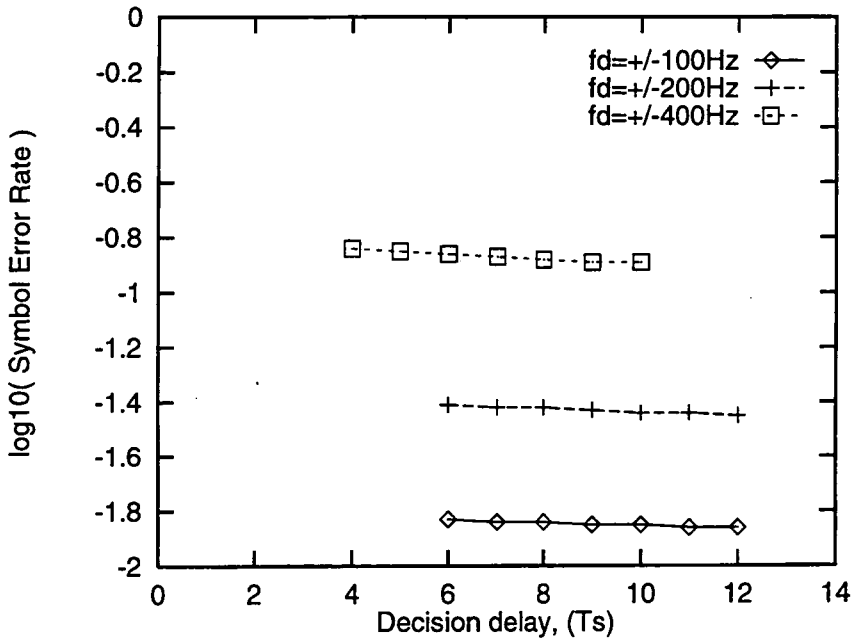

Figure 5.16: DDD-MLSE performance with varying secondary decision delay. $L = 3$, T_s-spaced equipower paths, SNR=15dB.

5.3.7 Frame Structure Effects

In this section the frame structure is varied to provide additional confirmation that the 100:20 frame structure used in the previous simulations is not the cause of the symbol error rate degradation. Increasing the length of the training sequence is shown not to affect the results significantly, indicating that channel acquisition during the 20 symbol training sequence is adequate. Increasing the data burst length is shown to degrade the symbol error rate. Figures 5.17, 5.18 and 5.19 show the results for $f_d = \pm 100$Hz, ± 200Hz and ± 400Hz respectively at an SNR$\approx$10dB for the 100Hz and 200Hz case and an SNR$\approx$15dB for the 400Hz case. The simulations used system and channel parameters as follows:

Channel parameters		
L	Path delay	Doppler spread
1	T_s	Classical, $f_d = \pm 100$Hz $= \pm 200$Hz $= \pm 400$Hz

System parameters				
Tx./Rx. filters	Frame structure	Decision delay	Channel estimator taps	μ
none	variable	4,2,2	2	0.1

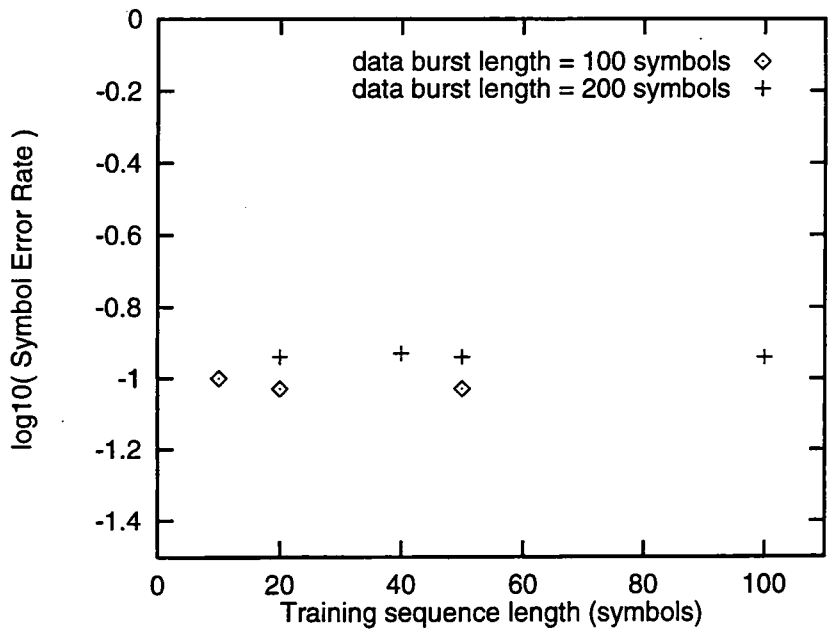

Figure 5.17: MLSE performance with varying frame structure. $L = 1$, T_s-spaced equipower paths, $f_d = \pm 100$Hz, SNR=10dB.

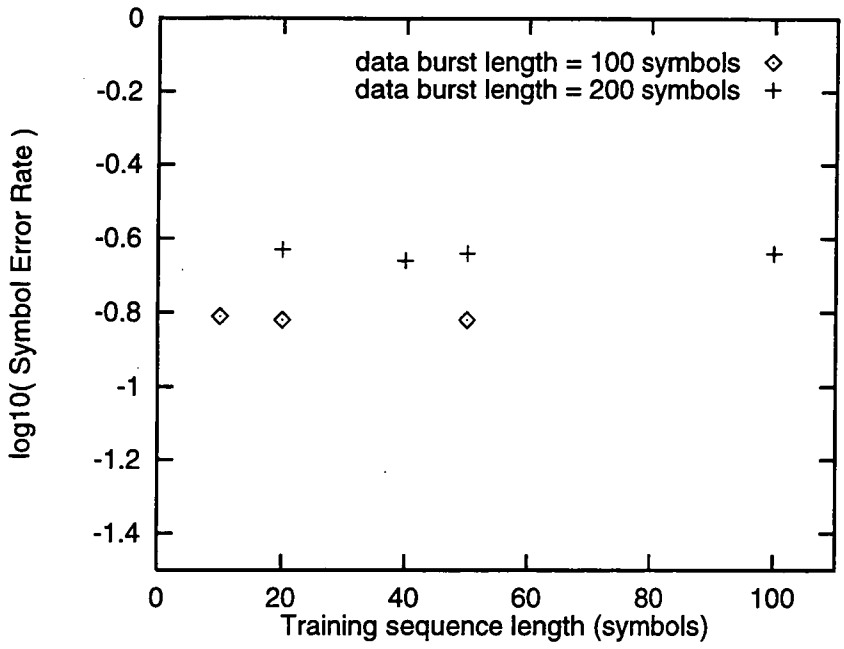

Figure 5.18: MLSE performance with varying frame structure. $L = 1$, T_s-spaced equipower paths, $f_d = \pm 200$Hz, SNR=10dB.

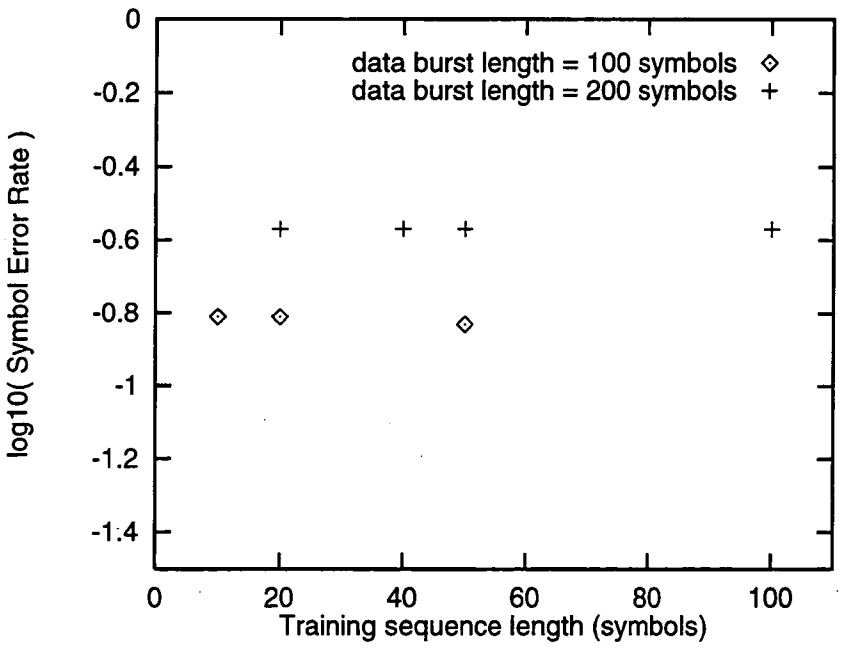

Figure 5.19: MLSE performance with varying frame structure. $L = 1$, T_s-spaced equipower paths, $f_d = \pm 400$Hz, SNR=15dB.

5.3.8 Channel Power-Delay Profile Effects

The simulations in the sections above have all used equipower path channel models. The MLSE's SER performance for a 4 T_s-spaced path channel with a tapered power-delay profile is simulated in this section. The four paths have relative powers $[1.0, 0.5, 0.25, 0.125]$ (normalised to give a channel power gain, $G_c = 1.0$). The channel estimator is operated in the 6 modes as given in Section 5.3.1. Other simulation parameters are given in the table below.

Channel parameters		
L	Path delay	Doppler spread
3	T_s	Classical, $f_d = \pm 200$Hz

System parameters				
Tx./Rx. filters	Frame structure	Decision delay	Channel estimator taps	μ
none	100:20	6	4	0.1

Figure 5.20 demonstrates that similar channel estimation problems exist in this tapered power-delay profile channel. Channel estimator tracking is highly beneficial and incorrect symbol estimates cause degradations in the channel estimator tracking capability, affecting the SER significantly.

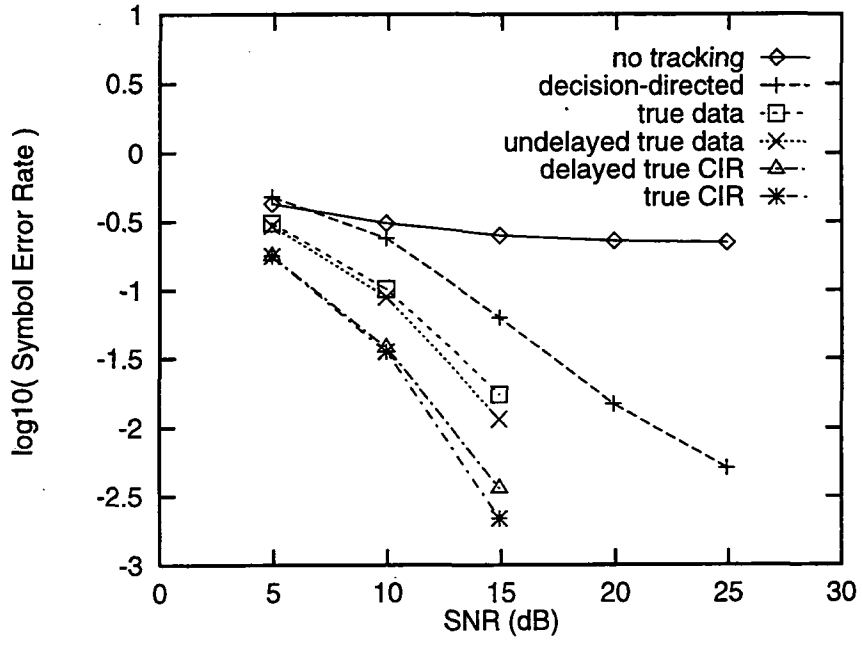

Figure 5.20: MLSE performance for a tapered power-delay profile channel. $L = 3$, T_s-spaced paths, $f_d = \pm 200$Hz.

5.3.9 Doppler Spread Effects

The effects of a non-classical Doppler spread power spectrum on the MLSE's SER performance are simulated. The Doppler characteristics of the path gains affect the time-varying nature of the channel and the level crossing rates of each path. In the simulation the Doppler spread power spectrum of each path gain is a Gaussian function with a (one sided) 3dB bandwidth of 200Hz.

Channel parameters		
L	Path delay	Doppler spread
2	T_s	Gaussian, 3dB = ±200Hz

System parameters				
Tx./Rx. filters	Frame structure	Decision delay	Channel estimator taps	μ
none	100:20	4	3	0.1

Figure 5.21 demonstrates the SER performance for the different channel estimator modes and indicates the performance gains obtained with increasingly more accurate channel estimation. The degradation obtained in decision-directed tracking is still present.

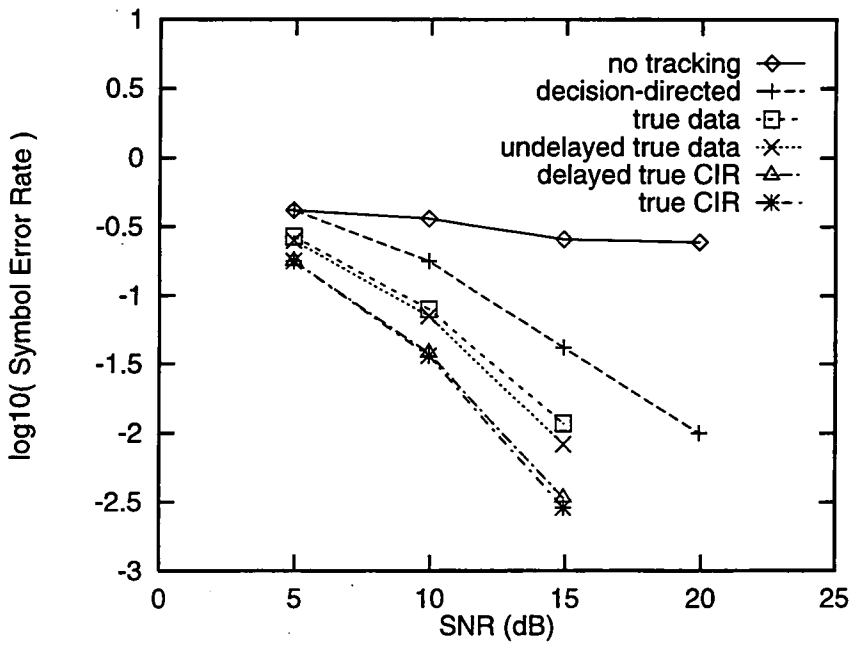

Figure 5.21: MLSE performance for a Gaussian Doppler spread channel. $L = 2$, T_s-spaced paths, 3dB= ±200Hz.

5.4 Summary

This chapter has demonstrated the characteristics of MLSE equalisers, operating in conjunction with an LMS adapted channel estimator, for a number of system and channel configurations. The symbol error rate is seen to deteriorate significantly when no tracking of the channel occurs during the data burst and this effect becomes more pronounced as the channel fades more rapidly. The performance of the conventional decision-directed tracking technique is seen to be degraded from the ideal tracking situation due to incorrect symbol feedback and equaliser decision delay. The former causes the channel estimator to be adjusted away from the true CIR as demonstrated by a rise in the mean tap error coinciding with the rise in the mean number of errors during the data burst. An error propagation mechanism exists as poor channel estimates increase the probability of a symbol estimate error which subsequently causes a larger channel estimation error. The decision delay effect is seen to be more critical for channels with higher fade rates; a fixed decision delay results in channel estimates that are increasingly out-of-date with the actual CIR as the channel varies more rapidly.

The presence of transmit and receive filters does not affect the qualitative results allowing them to be omitted and T_s-spaced channel models to be used during investigations into techniques that can overcome the shortcomings of the standard MLSE techniques. Once candidate solutions have been found for this idealised case, the effects of filters and more realistic channels may be simulated in order to increase confidence in the solution techniques.

The following chapter investigates the performance of Bayesian equalisers on fading channels in order to determine whether this alternative statistical detection technique is more robust to errors in the channel estimate than the MLSE.

Chapter 6

Bayesian Equalisers

A study of Bayesian decision feedback equaliser (BDFE) techniques was performed by Chen et al. [63] for use in fading and frequency selective radio channels. This paper showed similar and, under certain conditions, significantly improved symbol error rate performances over the standard MLSE techniques. One reason given for the improved performance was the ability of the Bayesian equaliser to produce similar quality symbol estimates as the MLSE but with much shorter decision delays. Low decision delays have advantages in decision-directed tracking of the channel estimator due to the problems associated with estimating an old CIR, especially at higher fade rates — see Chapter 5.

This chapter describes the operation of the BDFE in terms of a trellis perspective rather than a geometrical perspective [43, 69] in order to draw similarities with the Viterbi algorithm. The performance of the Bayesian equaliser is compared with that of the conventional MLSE under conditions of perfect CIR estimation and an analysis of the symbol error rate performance of the Bayesian equaliser using erroneous channel estimates is undertaken both analytically and numerically. The objective is to ratify whether the Bayesian equaliser is less prone to the effects of channel estimation errors than the conventional MLSE. If this is found to be the case, then it provides a greater confidence in the results given in [63]. These results are examined for validity in slightly different system and channel models by simulation and comparison of the SER performance of the Bayesian and MLSE equalisers. Positive results would prompt the development of a hybrid version of the MLSE and Bayesian equaliser; the Bayesian equaliser may be used to produce improved quality symbol estimates with a low decision delay for channel estimator tracking followed by an MLSE style trellis to provide final symbol estimates.

6.1 Bayesian Equaliser Techniques

The operation of the BDFE from a geometrical / Radial Basis Function (RBF) viewpoint is detailed in Chen *et al* [43] — see Section 3.5.2. The equaliser and detector form symbol estimates by choosing the most probable symbol given m received samples — the received signal vector, $\underline{\mathbf{r}}(k)$. The combined CIR and the transmitted symbol alphabet are used to calculate the received signal states,

$$r_i'(k) \qquad (i = 0, \ldots, (M^{L+1} - 1))$$

i.e. the set of received signal samples in the absence of noise. M is the size of the transmitted symbol alphabet and L the channel memory (assuming a T_s-spaced CIR model). Sequences of m received signal states are grouped together forming m-dimensional received signal state vectors, $\underline{\mathbf{r}}_j'(k)$, (or 'centres' in RBF terminology), with each one corresponding to a particular sequence of the transmitted symbols,

$$[x(k), x(k-1), \cdots, x(k-L-m+1)]$$

resulting in a total of M^{L+m} received signal state vectors. Each vector is associated with an individual symbol in the sequence so as to give a fixed equaliser decision delay, d, corresponding to a distinct mapping

$$\underline{\mathbf{r}}_j'(k) \rightarrow (x(k-d) = x_l) \qquad (l = 1, \ldots, M).$$

The received signal vector, $\underline{\mathbf{r}}(k)$, corrupted by noise, is used to form a metric for each of the M symbols in the transmitted alphabet by calculating the probability that the symbol was transmitted given the received signal vector,

$$P(x(k-d) = x_l | \underline{\mathbf{r}}(k)) \qquad l = 1, \ldots, M.$$

This probability is formed by calculating the probability of receiving the received signal vector given each 'centre' and summing the values for all centres mapping to the particular symbol, $(x(k-d) = x_l)$; individual probabilities are formed using a knowledge of the noise mechanism (usually an additive white Gaussian process),

$$P\left(x(k-d) = x_l | \underline{\mathbf{r}}(k)\right) \propto \sum_j P\left(\underline{\mathbf{r}}_j'(k) | \underline{\mathbf{r}}(k)\right) \tag{6.1}$$

$$\propto \sum_j p_n\left(\underline{\mathbf{r}}(k) - \underline{\mathbf{r}}_j'(k)\right) \tag{6.2}$$

$$\propto \sum_j \exp\left(\frac{\|\underline{\mathbf{r}}(k) - \underline{\mathbf{r}}_j'(k)\|^2}{2\sigma_n^2}\right) \tag{6.3}$$

Here $p_n(\cdot)$ is the noise probability density function and

$$\underline{\mathbf{r}}(k) = \underline{\mathbf{r}}'_j(k) + \underline{\mathbf{n}}(k)$$

where $\underline{\mathbf{n}}(k)$ is the vector of noise samples and σ_n^2 is the noise variance. The summation is performed for all values of j such that $\underline{\mathbf{r}}'_j(k) \rightarrow (x(k-d) = x_l)$. This metric generating process forms the *a posteriori* estimate of the transmitted symbol, $x(k-d)$.

The number of terms in the summation (Eqn. 6.3) may be reduced by the use of decision feedback. Previous symbol estimates are formed into an n-dimensional vector,

$$[\widehat{x}(k-d-1), \ldots, \widehat{x}(k-d-n)]$$

and only the received signal state vectors which have a corresponding subsequence in their asscociated symbol sequences, $[x(k), x(k-1), \cdots, x(k-L-m+1)]$, are used in the summation in Eqn. 6.3. n is chosen such that the oldest symbol estimate in the feedback vector matches the oldest symbol in the received signal state vector's symbol sequence. Thus

$$\begin{aligned} d+n &= L+m-1 \\ \Rightarrow \quad n &= L+m-d-1 \end{aligned} \tag{6.4}$$

The detector finally selects the largest metric, indicating the highest *a posteriori* probability, and outputs the corresponding symbol estimate, $\widehat{x}(k-d)$.

6.1.1 Trellis Interpretation of the Bayesian Equaliser

The trellis perspective of the BDFE splits the m-dimensional received signal state vectors into paths through an m-stage trellis. Figure 6.1 shows the trellis of received signal states for the case $M = 2$, $L = 1$ and $m = 4$, resulting in a total of $\left(m.M^{L+1}\right)$ states with each stage of the trellis having M^{L+1} states. A set of m states, one from each stage, constitutes the received signal state vector indicated as a path through the trellis. The sequential nature of the transmitted symbols results in restrictions in the transitions between states in one stage and the next with each state having M arrival paths and M exit paths. Thus there are M^{L+m} valid paths through the trellis corresponding to the M^{L+m} received signal state vectors.

At time $t = (k-m+1)$ there are M^L received signal states which map to the same symbol estimate, $\widehat{x}(k-d)$. Each of these states have M^{m-1} paths emmanating from them resulting in a total of M^{L+m-1} paths / received signal state vectors corresponding to a distinct symbol estimate. The *a posteriori* symbol decision metric is formed from metrics associated with each of these paths. The dotted lines in Figure 6.1 indicate the M^{m-1} valid paths from a

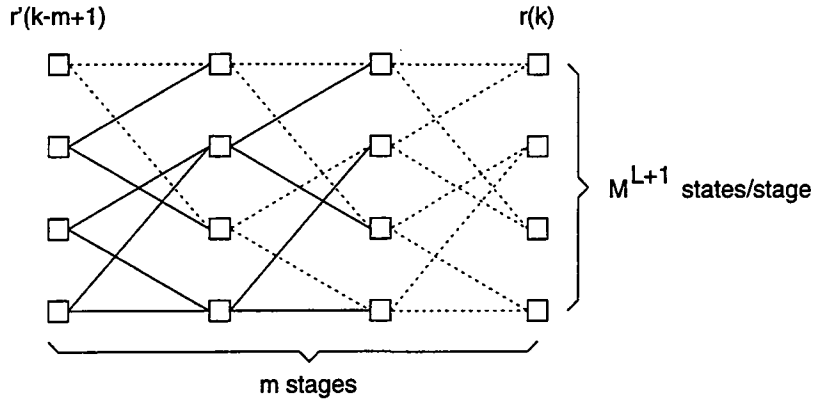

Figure 6.1: Trellis representation of the BDFE. In this example $M = 2$, $L = 1$ and $m = 4$. The dotted lines represent paths leading from a single received signal state.

single received signal state at time $t = (k - m + 1)$. The path metric for path j is formed from the m received signal samples and the m received signal states corresponding to the received signal state vector, $\underline{\mathbf{r}}'_j(k)$,

$$\lambda_j(k) = \sum_{p=0}^{m-1} |r(k-p) - r'_i(k-p)|^2 \tag{6.5}$$

$$= \|\underline{\mathbf{r}}(k) - \underline{\mathbf{r}}'_j(k)\|^2 \tag{6.6}$$

which corresponds to the square Euclidean distance between the two vectors in the geometrical interpretation of the BDFE. The path metrics are then combined as in Eqn. 6.3 to give the symbol decision metric.

Setting $m = (L + 1)$ ensures that the received signal samples used in the decision metric calculation $(r(k), \ldots, r(k - m + 1))$ contain all the energy from the symbol $x(k - L)$. The decision delay $d = (m - 1)$ is adopted so that when $m = (L + 1)$ a decision is made on $\widehat{x}(k - d) = \widehat{x}(k - m + 1) = \widehat{x}(k - L)$. The choice of $m = (L + 1)$ is not imperative; indeed the channel memory is not usually known *a priori*. The selection of $d = (m - 1)$ results in the decision metric selecting the first symbol in the associated sequence of state $r'(k - m + 1)$ — the state from which the paths start.

Decision feedback in the BDFE reduces the number of path metrics used in the decision metric calculation. The feedback vector of length $n = (L + m - d - 1)$ contains symbol estimates,

$$[\widehat{x}(k - d - 1), \ldots, \widehat{x}(k - d - n)]$$

which, substituting for n, $d = (m - 1)$ and $m = (L + 1)$ is equal to,

$$[\widehat{x}(k - m), \ldots, \widehat{x}(k - 2L)].$$

Only one received signal state for each symbol estimate has, at $t = (k-m+1)$, an associated symbol sequence containing this subsequence. Thus the decision metrics for each symbol estimate are formed from the metrics of paths starting at a single received signal state.

This trellis interpretation of the BDFE has similarities to the Viterbi implementation of MLSE equalisers. The BDFE has M^{L+1} states per trellis stage as opposed to M^L in the Viterbi algorithm but the transition metrics are calculated in the same manner in both algorithms. The path metrics differ in that the BDFE sums the m most recent transition metrics whereas the Viterbi path metrics are the sum of all the transition metrics on the path since a reset. The concept of survivor paths does not exist in the trellis interpretation of BDFE and the decision metrics are formed from a function of a number of path metrics.

6.2 Analysis of Channel Estimation Errors on BDFE Performance

The MLSE was shown to have a better performance than the BDFE for non-fading channels in Chen *et al* [43] but the relative performance between the two for fading channels is less certain. One possible reason for a potential increased performance of the BDFE is that it may be more robust to channel estimation errors. The BDFE's decision metric, formed from a set of path metrics rather than a single path metric, may be able to compensate for inaccurate channel models.

An estimated CIR, $\underline{\hat{\mathbf{h}}}(k)$ is used in both equalisers to form estimates of the received signal states,

$$\hat{r}_i'(k) = \underline{\hat{\mathbf{h}}}^T(k)\underline{\mathbf{x}}(k) \tag{6.7}$$

where $\underline{\mathbf{x}}(k) = [x(k), x(k-1), \cdots, x(k-L)]$. The channel estimate is constantly changing as it tries to track the fading channel and so the set of ideal received samples varies from sample to sample. The errors in $\hat{r}_i'(k)$ will be mutually correlated even for mutually uncorrelated errors in the channel tap estimates and are almost certainly correlated in time [59]. Any errors in the channel estimate will produce errors in the path metrics and the BDFE may be less susceptible than the MLSE to such errors.

The impact of channel estimation errors on the MLSE equaliser performance whilst operating on a slowly fading channel was investigated by Dzung [70] but did not consider the effects of decision-directed tracking. The following sections analyse the error mechanisms in the BDFE and compare the performance with the MLSE.

6.2.1 Performance with no Channel Estimation Errors

The performance of the MLSE and BDFE for varying decision delays and equaliser structures whilst using perfect channel estimates are compared. The simulations were run to investigate the effects of low decision delays on the symbol error rate performance. Such low decision delays may be used to advantage for the purposes of decision-directed tracking in the channel estimator.

In addition to the standard MLSE and BDFE, a variant of the MLSE, the 'windowed'-MLSE, (W-MLSE), has been designed and simulated to demonstrate the effects of using more than one path metic to form the decision metric when the path metrics are formed using a limited number of received signal samples. The structure of the W-MLSE is the same as the MLSE except that the path metrics are formed using only the last m received signal samples instead of all samples since a reset. Each received signal state still has only one survivor path and the mapping from received signal state to transmitted symbol remains the same. This mechanism may be viewed as a sliding-windowing of the path metric. Both the BDFE and W-MLSE form their respective decision metrics using only m received signal samples. As the window length of the W-MLSE is increased the SER performance will approach that of the MLSE.

The fading channel has the same characteristics as that used in Section 5.3.1 with three T_s-spaced equipower Rayleigh fading paths. The paths fade with a classical Doppler power spectrum with $f_d = \pm 100$Hz. The data is QPSK modulated at 100kBaud and no transmit / receive filters have been used. As the three T_s-spaced true CIR is being used as the channel estimate, no frame structure in the data needs to be employed. Complex white Gaussian noise corrupts the received signal as normal. The simulation parameters are summarised in the tables below.

Channel parameters		
L	Path delay	Doppler spread
2	T_s	Classical, $f_d = \pm 100$Hz

System parameters				
Tx./Rx. filters	Frame structure	Decision delay	Channel estimator taps	μ
none	n/a	0,1,2	3	n/a

The single decision delay version of the MLSE is used and the BDFE is configured such that $m = (d + 1)$ and $n = (L + m - d - 1) = 2$ or $n = 0$ to simulate the performance with and without decision feedback. In the case of decision feedback the actual symbol estimates were used in the feedback delay line rather than the idealised situation of using perfect symbol estimates. The BDFE uses a perfect knowledge of the receiver noise power to form its decision

metrics (Eqn. 6.3) — the degradation due to incorrect power estimates is small for errors in the range ±5dB. The W-MLSE has a metric window of length $WL = m$ and has a decision delay, $d = (m - 1)$.

The simulations compare the SER vs. SNR for MLSE, W-MLSE and BDFE equalisers whilst using an undelayed perfect knowledge of the CIR. Figures 6.2, 6.3 and 6.4 show the results for decision delays of 0, 1 and 2 symbol periods respectively. From these curves it can be seen that the MLSE consistently outperforms the BDFE and W-MLSE even with very small decision delays. The MLSE's symbol estimate is based on many more received samples than the BDFE or W-MLSE even though the path is not traversed to a great depth whilst determining the symbol estimate. In these cases the paths will not have converged as the truncation path length is so small. The BDFE curves show that decision feedback improves performance which indicates that the process of forming a decision metric from a large set of paths is not nessecarily advantageous. In Chen *et al* [43] this effect was given the geometrical interpretation that the minimum square Euclidean distance between sets of received signal states is increased using decision feedback. The BDFE with decision feedback has an improved SER compared to that of the W-MLSE demonstrating the benefit of using more than one path through the trellis to form a decision metric based on a small number of received signal samples. However, the absence of decision feedback causes a significant performance degradation in the BDFE and this effect increases as more received signal samples are used to form the symbol estimates. The quality of the BDFE's symbol estimates becomes worse than that of the W-MLSE.

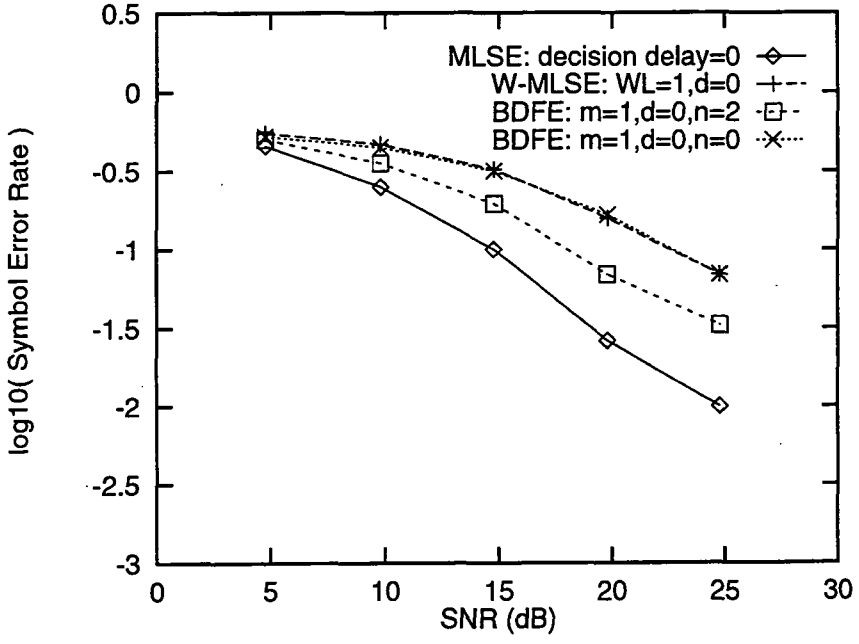

Figure 6.2: MLSE/BDFE performance using true channel estimates. Decision delay = 0.

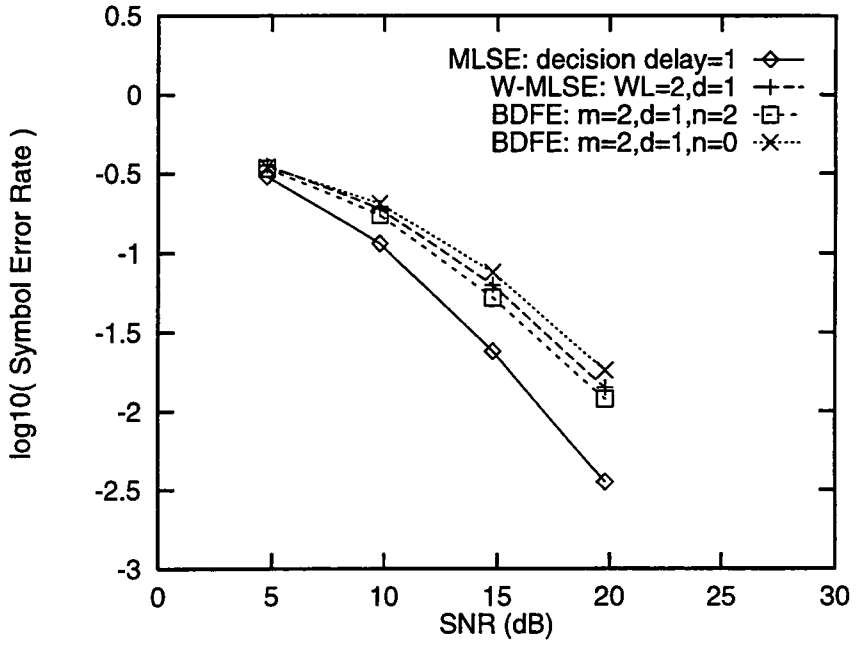

Figure 6.3: MLSE/BDFE performance using true channel estimates. Decision delay = 1.

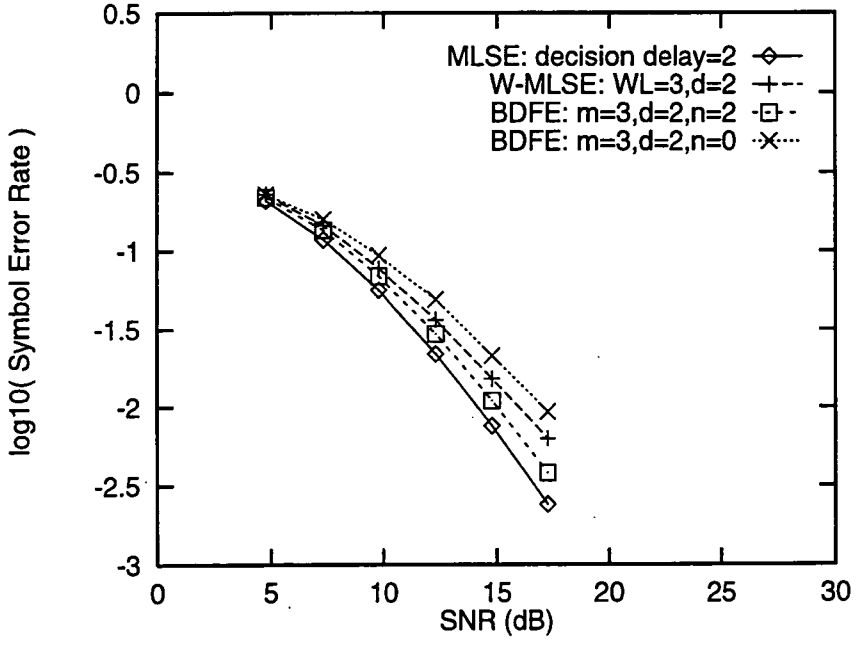

Figure 6.4: MLSE/BDFE performance using true channel estimates. Decision delay = 2.

6.2.2 Analysis of Channel Estimation Error Effects

This section examines the effect of channel estimation errors on the probability of error of the BDFE. The nonlinear nature of the equaliser makes an analytical solution difficult, at least, and will depend on the cross-correlation and auto-correlation of the channel estimation errors. After an initial analytical derivation for the simplest case, numerical techniques are explored leading to Monte-Carlo methods for approximating the symbol error rate.

The simplest case for analysis purposes is to consider a binary PAM data source being transmitted through an ideal channel of known phase but with unknown gain and corrupted by additive white Gaussian noise (Fig. 6.5). A channel estimator provides an estimate of the impulse response (a single impulse) and feeds it to the BDFE which, together with the received signal sample, estimates the transmitted symbol. The BDFE is configured to use a single received sample ($m = 1$) and a decision delay of $d = 0$. Decision feedback is not relevant in this case as the CIR is a single impulse ($L = 0$) and so $n = 0$. Errors in the channel estimate cause errors in the BDFE decision metrics which may subsequently cause symbol estimation errors.

The transmitted symbol, ($x(k) \in \{+1, -1\}$), is passed through the channel, $H(z) = h_0$, giving a received signal state, $r'(k) = h_0(k)x(k)$, which is corrupted by noise to form the received signal sample, $r(k) = h_0(k)x(k) + n(k)$. The time indicies may be dropped for convenience. The BDFE forms two decision metrics, f_{+1} and f_{-1} corresponding to the transmitted symbols $+1$ and -1 respectively:

$$f_{+1} = \alpha \exp\left(\frac{-d_0^2}{\rho}\right) \tag{6.8}$$

$$f_{-1} = \alpha \exp\left(\frac{-d_1^2}{\rho}\right) \tag{6.9}$$

where α is a positive constant, d_0^2 and d_1^2 are the square Euclidean distances between the received signal sample and received signal states and ρ is an estimate of $2\sigma_n^2$. Figure 6.6 shows the notation used in the derivation. The points marked ' $\times$ ' are the received signal states and the point marked ' $\bigcirc$ ' is the actual received signal sample. The additive noise is either n_0, given a $+1$ was transmitted, or n_1, given a -1. The estimate of the channel gain, $\widehat{h}_0$, has an associated error, e, which results in the BDFE assuming a channel gain of $(h_0 + e)$. The estimated received signal states are marked ' $\square$ '. Thus the Euclidean distances, as measured by the BDFE, have an error $\mp e$.

The BDFE will produce a symbol estimation error if

$$(f_{+1} < f_{-1} \text{ AND } r' = h_0) \text{ OR } (f_1 < f_{-1} \text{ AND } r' = -h_0). \tag{6.10}$$

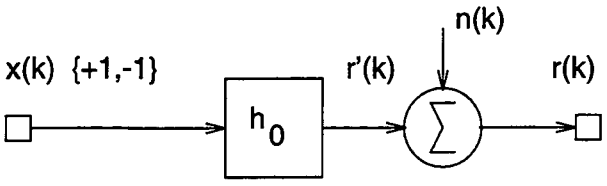

Figure 6.5: System and channel model ($L = 0$) used for the theoretical analysis.

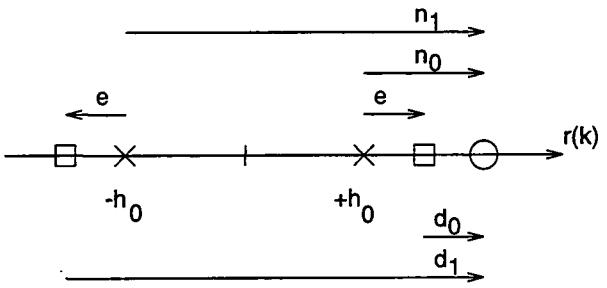

Figure 6.6: Definitions of signals for $L = 0$.

Therefore the probability of a symbol error is

$$P_e = P(f_{+1} < f_{-1}, r' = h_0) + P(f_{-1} < f_{+1}, r' = -h_0) \tag{6.11}$$

as the events $(r' = h_0)$ and $(r' = -h_0)$ are mutually exclusive.

The characteristic of the $\exp(\cdot)$ function allows substitution of f_{+1} and f_{-1} as, if $f_{+1} < f_{-1}$ then $d_0^2 > d_1^2$ and if $f_{-1} < f_{+1}$ then $d_1^2 > d_0^2$:

$$\begin{aligned}
\Rightarrow P_e &= P(d_0^2 > d_1^2, r' = h_0) + P(d_1^2 > d_0^2, r' = -h_0) && (6.12)\\
&= P(d_0^2 > d_1^2 | r' = h_0)P(r' = h_0) + P(d_1^2 > d_0^2 | r' = -h_0)P(r' = -h_0) && (6.13)\\
&= \frac{1}{2}P(d_0^2 > d_1^2 | r' = h_0) + \frac{1}{2}P(d_1^2 > d_0^2 | r' = -h_0) && (6.14)
\end{aligned}$$

as $P(r' = h_0) = P(r' = -h_0) = \frac{1}{2}$ for independent, identically distributed transmitted symbols. The first term of Eqn. 6.14 may be expanded thus:
Given $r' = h_0$, then $r = h_0 + n_0$ and

$$\begin{aligned}
d_1 &= r - (h_0 + e)\\
&= n_0 - e && (6.15)\\
d_2 &= r - (-h_0 - e)\\
&= n_0 + 2h_0 + e && (6.16)
\end{aligned}$$

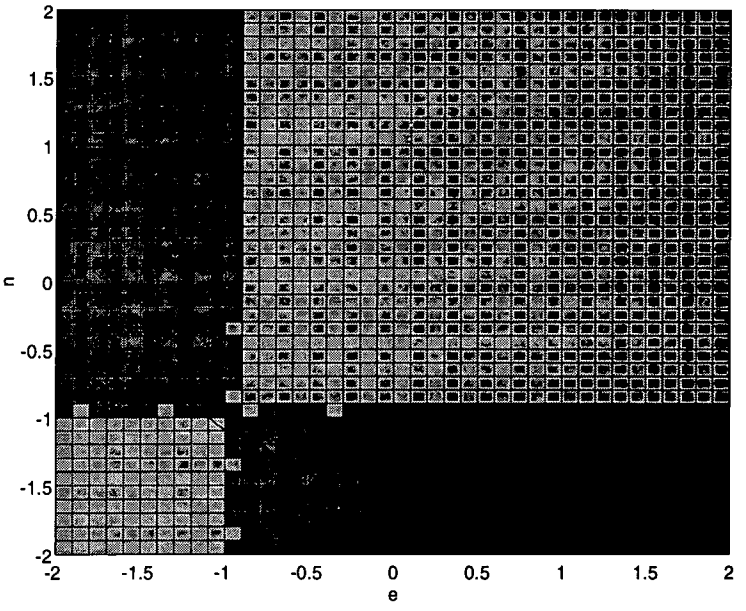

Figure 6.7: Classification regions for a single path channel with gain $h_0 = 1.0$. The darker shaded areas result in errors and the probability of error is obtained by integrating the pdf over these regions. n is the noise variable and e the channel estimation error variable.

Therefore

$$d_0^2 = n_0^2 + e^2 - 2n_0e \tag{6.17}$$

$$d_1^2 = n_0^2 + 4h_0^2 + e^2 + 2n_0e + 4h_0n_0 + 4h_0e \tag{6.18}$$

and

$$P(d_0^2 > d_1^2 | r' = h_0) = P(h_0^2 + n_0e + h_0n_0 + h_0 < 0) \tag{6.19}$$

This inequality may be split into two parts (Figure 6.7 shows these regions graphically):

$$P(d_0^2 > d_1^2 | r' = h_0) = P(e < -h_0, n_0 > -h_0) \text{ OR } P(n_0 < -h_0, e > -h_0) \tag{6.20}$$

These two events are mutually exclusive and, assuming the noise and channel estimator error are independent:

$$P(d_0^2 > d_1^2 | r' = h_0) = P(e < -h_0)P(n_0 > -h_0) + P(n_0 < -h_0)P(e > -h_0). \tag{6.21}$$

Assuming e and n_0 have Gaussian distributions with means $\mu_e = \mu_n = 0$ and variances σ_e^2

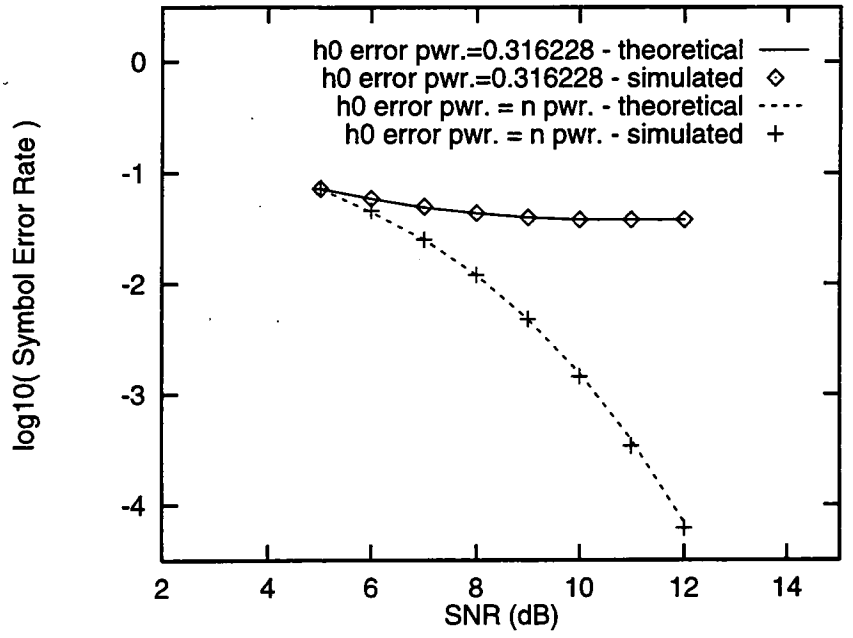

Figure 6.8: Comparison of theoretical and numerical solutions for the SER.

and σ_n^2 respectively, Eqn. 6.21 may be written:

$$P(d_0^2 > d_1^2 | r' = h_0) = Q\left(\frac{h_0}{\sigma_e}\right) + Q\left(\frac{h_0}{\sigma_n}\right) - 2Q\left(\frac{h_0}{\sigma_e}\right) Q\left(\frac{h_0}{\sigma_n}\right) \tag{6.22}$$

The second term of Eqn. 6.14 may be expanded in a similar manner leading to the same expression for $P(d_1^2 > d_0^2 | r' = -h_0)$ as Eqn. 6.22. The two expressions may be combined into Eqn. 6.14 to give a final expression of the probability of error:

$$P_e = Q\left(\frac{h_0}{\sigma_e}\right) + Q\left(\frac{h_0}{\sigma_n}\right) - 2Q\left(\frac{h_0}{\sigma_e}\right) Q\left(\frac{h_0}{\sigma_n}\right) \tag{6.23}$$

Figure 6.8 confirms this expression by comparing the curve with results from a Monte-Carlo based simulation. The upper curve shows the SER for $\sigma_e^2 = 0.316228$ (-5dB relative to the received signal power) and the lower curve for $\sigma_e^2 = \sigma_n^2$. The simulated points are in close agreement with those predicted by Eqn. 6.23.

In reality the assumption that e and n are mutually independent is invalid as the channel estimation error will be dependent on the receiver noise. However, the method by which P_e is derived may still be used provided that a description of the joint probability, $P(e, n)$ is available.

The values of channel estimate errors and noise samples for which classification errors occur become more complex as the number of channel paths increase and have real and imaginary components. An increase in the size of the transmitted symbol set and the number

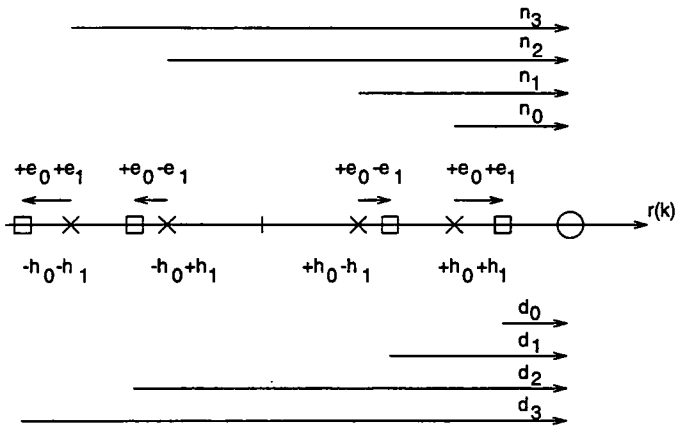

Figure 6.9: Definitions of signals for $L = 1$.

of received signal samples (m) used by the BDFE further complicates the analytical solution, if any. To demonstrate this, the next most simple case of a 2 real path channel ($L = 1$), $H(z) = h_0 + h_1 z^{-1}$, and BPSK data ($x(k) \in \{+1, -1\}$) is considered. The received signal states $r'_0 = (h_0 + h_1)$ and $r'_1 = (h_0 - h_1)$ map to the symbol $x = +1$ and $r'_2 = (-h_0 + h_1)$ and $r'_3 = (-h_0 - h_1)$ map to the symbol $x = -1$. The BDFE is configured with $m = 1$, $d = 0$ and $n = 0$ and uses a 2 tap channel estimator to provide estimates of the path gains: $\widehat{h}_0 = h_0 + e_0$ and $\widehat{h}_1 = h_1 + e_1$. The decision metrics are given by:

$$f_{+1} = \alpha \left[\exp\left(\frac{-d_0^2}{\rho}\right) + \exp\left(\frac{-d_1^2}{\rho}\right) \right] \tag{6.24}$$

$$f_{-1} = \alpha \left[\exp\left(\frac{-d_2^2}{\rho}\right) + \exp\left(\frac{-d_3^2}{\rho}\right) \right]. \tag{6.25}$$

The signal terminology is shown in Figure 6.9.

The probability of error is given by,

$$\begin{aligned} P_e = \; & P(f_{+1} < f_{-1}, r' = r'_0) \\ & + P(f_{+1} < f_{-1}, r' = r'_1) \\ & + P(f_{-1} < f_{+1}, r' = r'_2) \\ & + P(f_{-1} < f_{+1}, r' = r'_3) \end{aligned} \tag{6.26}$$

since the received signal states are mutually exclusive. The mapping of the inequalities in Eqn. 6.26 to the corresponding inequalities in terms of d_0, d_1, d_2 and d_3 is not straightforward due to the $\exp(\cdot)$ function. A graphical representation of the problem is shown in Figure 6.10 which shows the regions in which the BDFE gives a symbol estimation error for the first term in Eqn. 6.26, i.e. when $r' = r'_0$ and $n = n_0$.

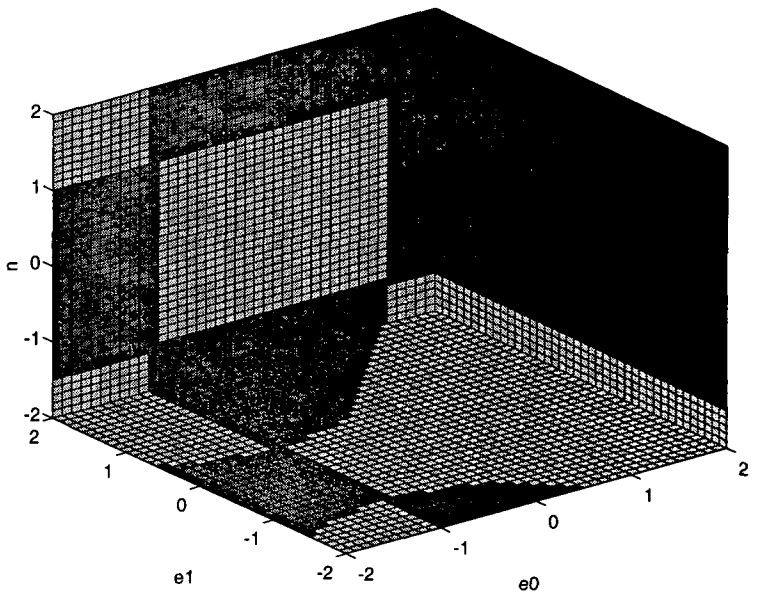

Figure 6.10: Classification regions for a 2 path channel with gains $h_0 = 1.0$ and $h_1 = 0.5$. The darker shaded areas result in errors and the probability of error is obtained by integrating the pdf over these regions. e_0 and e_1 are the channel estimation error variables and n the noise variable.

In light of these assumptions and problems, developing further analytical solutions was deemed to be of limited practical use. Numerical solutions to the problem were viewed to be able to provide a more efficient way to understand the effects of erroneous channel estimates. By determining and quantising the regions which result in classification errors, the probability density functions for the variables may be integrated numerically over these regions to give an estimation of the probability of error. Joint probability density functions may be included with less difficulty than in the analytical method. The accuracy of this method depends on the degree of quantisation of the regions and for typical examples with three complex channel paths and a noise variable, the integration over a 7-dimensional space has a large computational burden. Alternatively, standard Monte-Carlo techniques may be used to estimate the error probability by averaging the number of classification errors measured over the duration of a simulation.

The three methods (analytical, numerical integration and Monte-Carlo) were compared for a single path channel as in Section 6.2.2 and each gave consistent results given sufficiently fine quantisation in the numerical integration technique and simulation duration in the Monte-Carlo technique. In view of the difficulty of the analytical method and of the computational burden of numerical integration, Monte-Carlo techniques have been used in the subsequent simulations to find an approximation to the probability of symbol error.

6.2.3 SER for Time-invariant Minimum and Non-minimum Phase Channels

The results from Section 6.2.1 indicate that the Bayesian equaliser has a poorer performance than the MLSE whilst using a perfect knowledge of the ISI mechanism. The channel simulated was time-varying and so will have had an ensemble of instantaneous minimum and non-minimum phase channels. Such instantaneous channels have different mappings from the received signal states, $r'(k)$, to transmitted symbols, $x(k)$. For instance, the channels $H_1(z) = 0.447 + 0.894z^{-1}$ and $H_2(z) = 0.894 + 0.447z^{-1}$ have the same values of received signal states but an opposite mapping to transmitted symbols. Thus channel $H_1(z)$ results in $r'(k) = -0.447$ for a transmitted symbol sequence $[x(k)x(k-1)] = [+1 - 1]$ whereas channel $H_2(z)$ has the same value of the received signal state for a transmitted sequence $[-1 + 1]$. The Bayesian equaliser, maintaining the logical transitions between received signal states from one stage in the trellis to the next, produces different transition, path and decision metrics for the two channels despite the same set of values of the received signal states. This section investigates whether the phase characteristics of the channel influences the Bayesian equaliser's SER significantly whilst operating with an estimate of the CIR.

Simulations were performed on a number of 2 real, symbol-spaced path channels with path gains of h_0 and h_1 respectively. The values of these gains were restricted such that $(h_0^2 + h_1^2 = 1)$ to ensure a constant channel power gain. QPSK data was passed through the channel and corrupted by complex additive white Gaussian noise with component powers $\sigma_n^2 = 0.1$ to give an SNR of 10dB. The channel estimates were derived using the true channel taps corrupted by real additive white Gaussian noise with a power of 10dB relative to the received signal power prior to the addition of receiver noise. The system and channel characteristics are summarised as follows:

Channel parameters		
L	Path delay	Doppler spread
1	T_s	time-invariant

System parameters				
Tx./Rx. filters	Frame structure	Decision delay	Channel estimator taps	μ
none	n/a	0,1	2	n/a

Figure 6.11 shows the results for an MLSE equaliser with no decision delay, a BDFE with $m = 1$, $d = 0$, $n = 1$ and $n = 0$ and a W-MLSE with $d = 0$ and a window length, $WL = 1$. The curves show that there is very little difference between the MLSE and BDFE with decision feedback but the W-MLSE and BDFE ($n = 0$) have poorer performance for minimum phase channels.

Figure 6.12 shows the performance for the equalisers with decision delays corresponding to one symbol period. The MLSE has the lowest symbol error rate for both minimum and non-minimum phase channels. The BDFE with decision feedback ($n = 1$) has an inferior performance for non-minimum phase channels and is outperformed by both the W-MLSE and BDFE $n = 0$. For minimum phase channels the relative performances are reversed and the BDFE $n = 0$ has a higher error rate than the W-MLSE.

Further simulations were carried out on a fading channel with three complex, symbol-spaced equipower paths and the symbol error rates were plotted as a function of SNR. The system and channel configurations were the same as used in Section 6.2.1 except that the maximum Doppler shift of the paths was $f_d = \pm 200$Hz. The fading channel produces various minimum and non-minimum channels allowing the symbol error rate to be estimated over an ensemble of channel types. The Doppler spread of the channel paths does not affect the results significantly as long as the simulation averages over a significant number of fades. The channel estimates were produced by adding complex white Gaussian noise (with power equal to the receiver noise power) to the true channel taps. The equalisers had decision delays of two symbol periods and the BDFEs and W-MLSE made their decisions based on three received signal samples. Figure 6.13 shows the SER curves which demonstrate that the BDFE $n = 2$ slightly outperforms the W-MLSE at higher SNRs but both are marginally inferior to the MLSE. The BDFE $n = 0$ error rate is again the highest for all four equalisers compared.

The curves in Figure 6.13 may be compared to those in Figure 6.14 which show the results of the three equalisers operating on the same channel but using individual, 3 T_s-spaced tap channel estimators. The estimators were adapted using the LMS algorithm with $\mu = 0.1$ and operated in a decision-directed tracking mode with a 100:20 frame structure. The relative performance of the equalisers is similar with the BDFE $n = 2$ performing slightly worse than the MLSE. Both the W-MLSE and BDFE without decision feedback have significantly higher SERs especially at larger SNRs with the BDFE $n = 0$ having the highest SER.

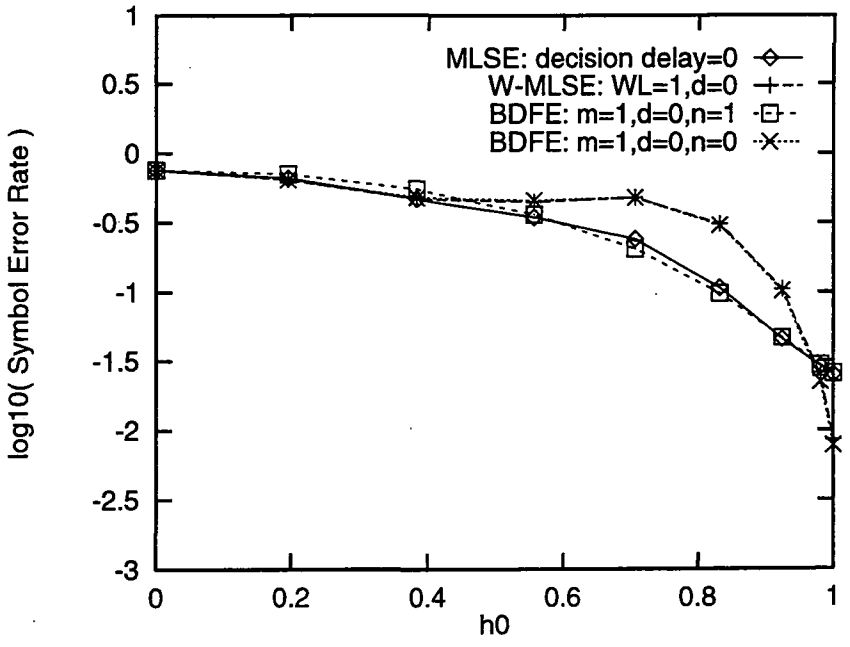

Figure 6.11: Comparison of MLSE, W-MLSE and BDFE equalisers with $d = 0$ for 2 real path, time-invariant channels. The channel estimate for each equaliser is generated by adding Gaussian noise to the true CIR.

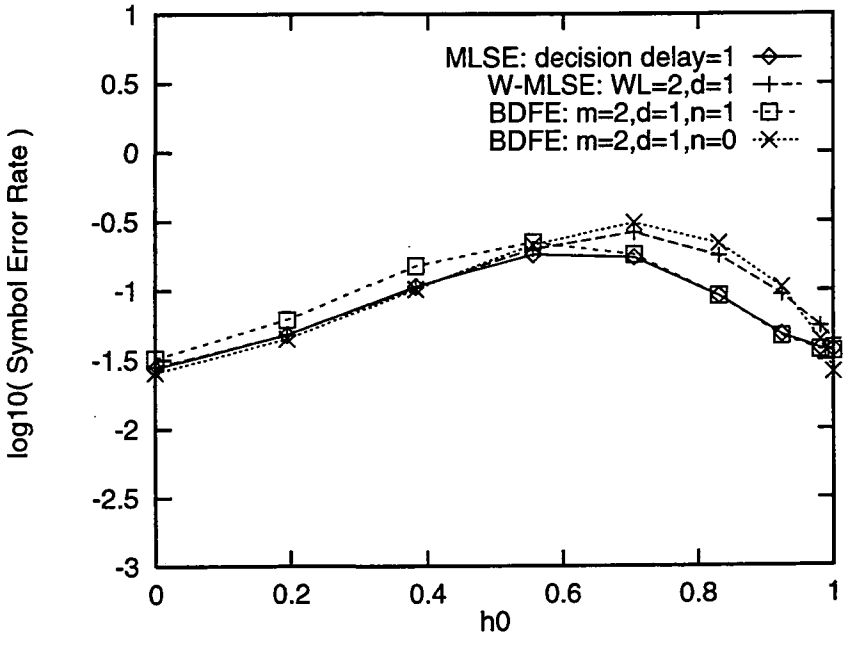

Figure 6.12: Comparison of MLSE, W-MLSE and BDFE equalisers with $d = 1$ for 2 real path, time-invariant channels. The channel estimate for each equaliser is generated by adding Gaussian noise to the true CIR.

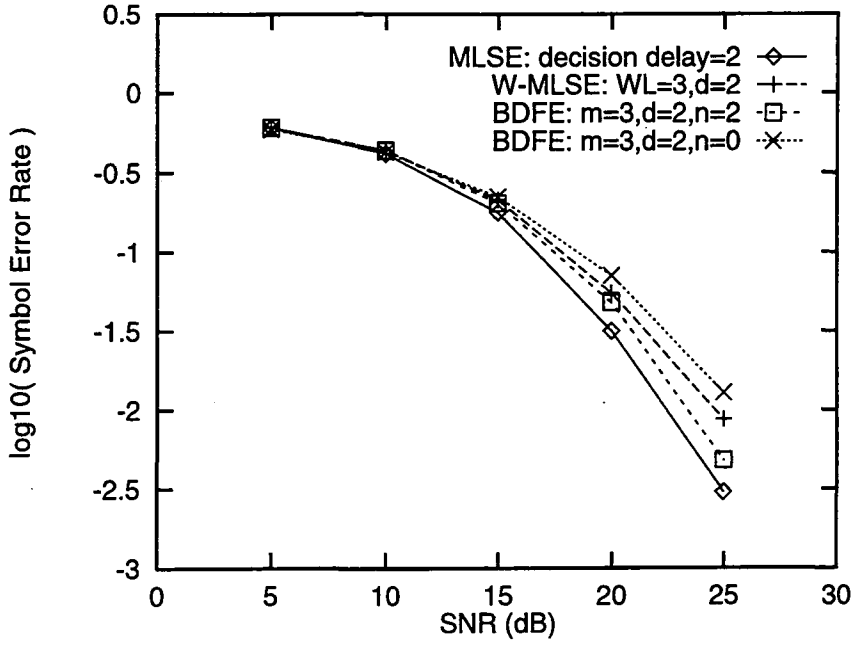

Figure 6.13: Comparison of MLSE, W-MLSE and BDFE equalisers with $d = 2$ for a 3 complex path fading channel. The channel estimate for each equaliser is generated by adding Gaussian noise to the true CIR.

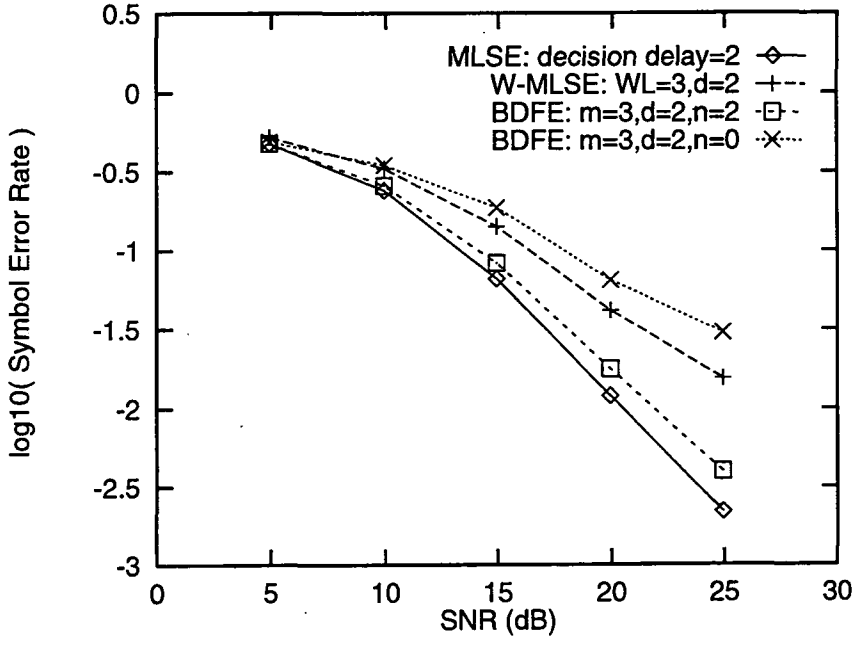

Figure 6.14: Comparison of MLSE, W-MLSE and BDFE equalisers with $d = 2$ for a 3 complex path fading channel. The channel estimate for each equaliser is generated by a separate, LMS adapted, channel estimator.

6.2.4 SER for Fading Channels

The results of the preceeding sections have indicated that the Bayesian equaliser does not outperform the MLSE either when accurate CIR information is available or when the CIR estimate is derived from the actual CIR corrupted by AWGN. The process of forming a decision metric from a set of path metrics rather than from a single path metric is beneficial when restrictions are placed on the number of received signal samples from which the metrics are derived but only when decision feedback is employed in the BDFE structure. This section presents results of simulations of the SER performance of Bayesian equalisers operating with a channel estimator configured in the modes described in Section 5.3.1. The system and channel model configurations are as follows:

Channel parameters		
L	Path delay	Doppler spread
2	T_s	Classical, $f_d = \pm 200$Hz ± 400Hz

System parameters				
Tx./Rx. filters	Frame structure	Decision delay	Channel estimator taps	μ
none	100:20	2	3	0.1

The BDFE was operated both with and without decision feedback ($n = 2$ and $n = 0$) and with $m = 3$ and $d = 2$.

Figures 6.15 and 6.16 show the SER vs. SNR for the BDFE with $n = 2$ and $n = 0$ respectively whilst equalising the channel with $f_d = \pm 200$Hz. The channel estimator affects the SER in a similar manner to the MLSE (see Chapter 5). Channel tracking significantly improves the quality of the symbol estimates for both equalisers and incorrect estimates have a pronounced adverse effect on the tracking and SER. The effects of decision delay are marginal at best as a result of the low decision delay employed by the equaliser ($d = 2$) — one of the beneficial properties of the BDFE [43]. The absolute performance of the BDFE with $n = 0$ is significantly worse than the BDFE with decision feedback. Comparing the decision-directed tracking curves between figures, decision feedback improves the SER by around a factor of 2 at an SNR = 15dB ($\log_{10}(\Delta \text{ SER}) = 0.3$).

Figures 6.17 and 6.18 show the results for the two equalisers equalising the $f_d = \pm 400$Hz channel. Again the equalisers show similar characteristics with both being sensitive to errors in the CIR estimate caused by incorrect symbol estimates being used during channel tracking. The effect of decision delay is more noticable as the channel has faster time variations causing the channel to have changed more within the $d = 2T_s$ decision delay period. The BDFE $n = 0$ again performs worse than the BDFE with decision feedback but the degradation

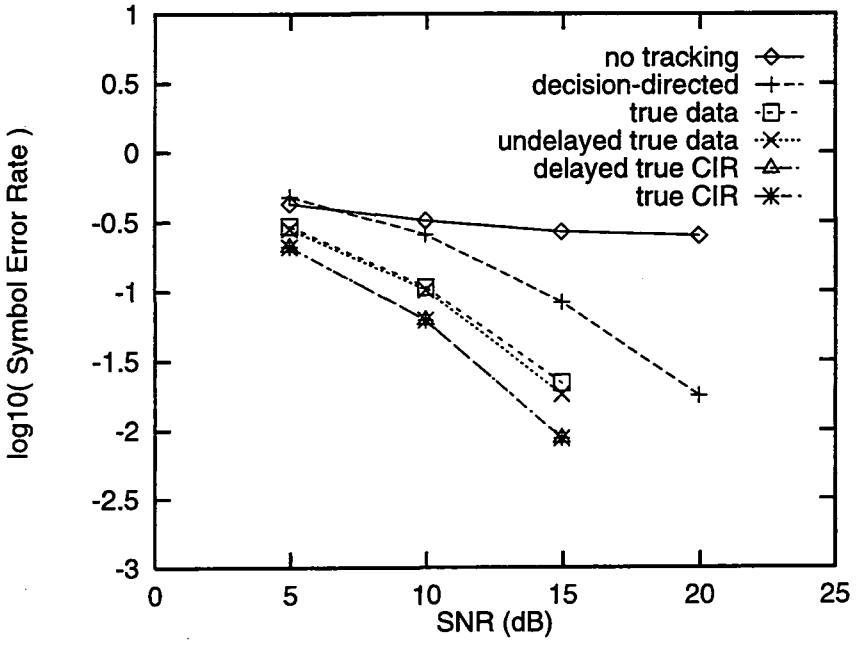

Figure 6.15: Performance of the BDFE $n = 2$ for an $L = 2$, $f_d = \pm 200$Hz channel. The curves show the SER whilst operating under various channel estimator configurations.

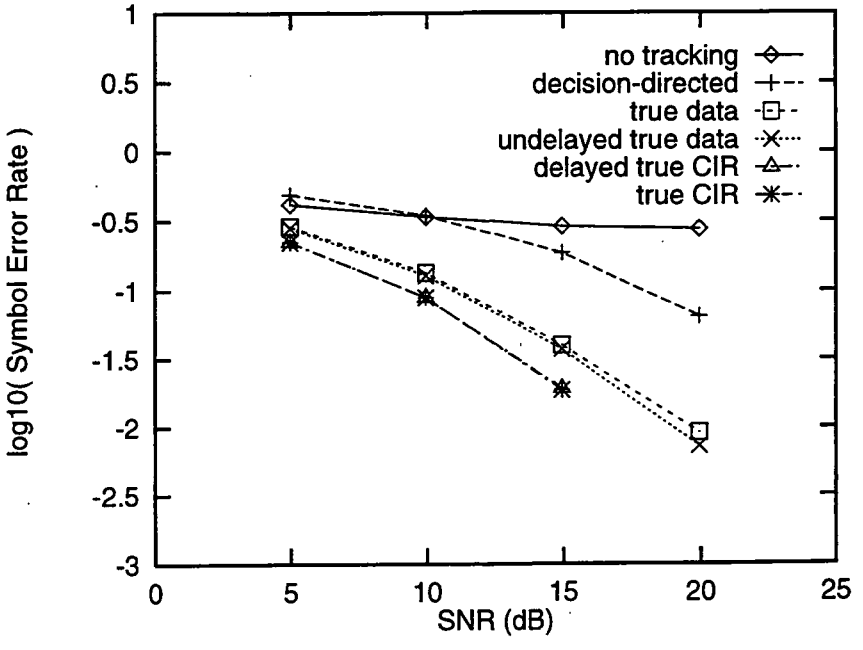

Figure 6.16: Performance of the BDFE $n = 0$ for an $L = 2$, $f_d = \pm 200$Hz channel. The curves show the SER whilst operating under various channel estimator configurations.

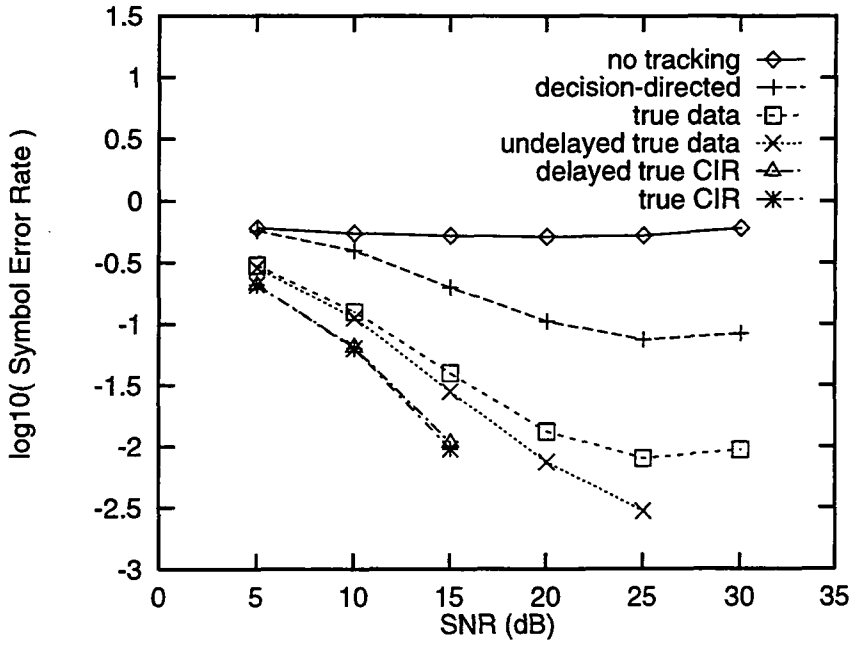

Figure 6.17: Performance of the BDFE $n = 2$ for an $L = 2$, $f_d = \pm 400$Hz channel. The curves show the SER whilst operating under various channel estimator configurations.

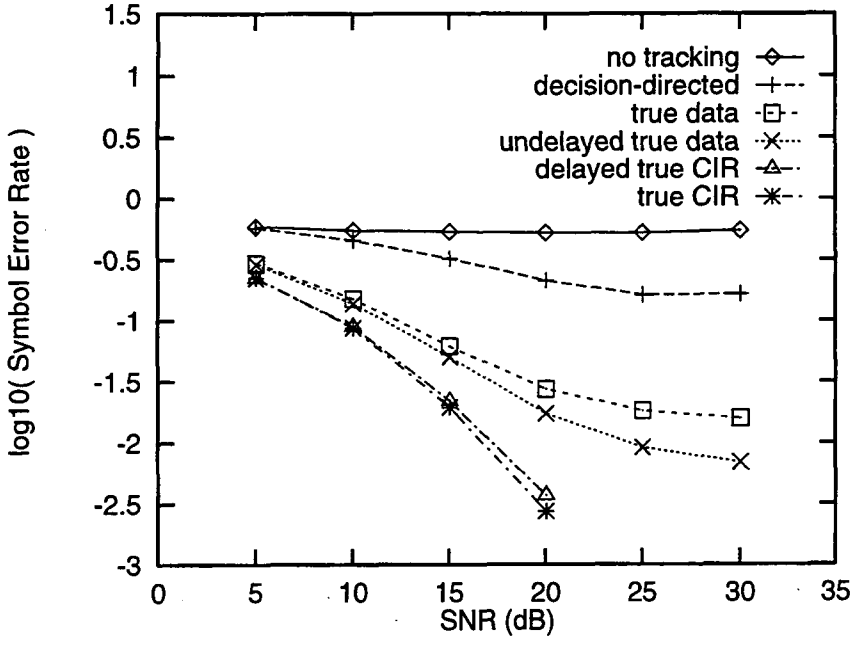

Figure 6.18: Performance of the BDFE $n = 0$ for an $L = 2$, $f_d = \pm 400$Hz channel. The curves show the SER whilst operating under various channel estimator configurations.

is less marked with the BDFE $n = 2$ having an SER $\approx$ 1.6 times lower than the $n = 0$ configuration at an SNR = 15dB.

6.3 Summary

This chapter has highlighted the operation of the Bayesian decision feedback equaliser from a trellis perspective. The process of forming the decision metric is shown to involve a function of a set of path metrics through the trellis. Each path metric is formed in a similar manner to path metrics in the Viterbi algorithm except that the metrics are formed from a much smaller set of received signal samples. The combination of the path metrics to form the decision metric is based on maximum *a priori* methods but is an approximation as the received signal states and the noise power are not fully known. The BDFE algorithm is relatively insensitive to the estimate of the noise power compared to the estimates of the received signal states.

The BDFE is shown to produce poorer symbol estimates than the MLSE whilst using the true channel impulse response to generate perfect estimates of the received signal states, confirming results in papers by Chen *et al* [43, 63]. Decision feedback reduces the number of path metrics used to form the BDFE's decision metric and this improves the SER performance rather than reducing it. The feedback restricts the paths used in the decision metric calculation to those paths emanating from a single state. This SER characteristic extends to situations where the equaliser uses channel estimates rather than the true CIR to generate the received signal states. Thus increasing the number of paths does not neccessarily increase the robustness to channel estimation errors. However the use of more than one path does increase the performance when the path metrics are formed from a small number of received signal samples. A 'windowed' version of the MLSE which forms its decision metric from a reduced number of received signal samples has been developed to demonstrate these benefits.

An analytical approach to calculating the probability of error of the equaliser whilst using erroneous channel estimates has been attempted resulting in the derivation of an expression for the most simple case of a single path channel. The results obtained have been confirmed through simulation. The extension of this technique to more realistic scenarios has not been undertaken due to the difficulties in describing the decision regions of the BDFE which are formed from a sum of exponential functions. Additionally there are difficulties in describing the noise and channel estimation error probability distributions as these depend on the precise channel estimation technique and the noise and estimation error variables are not independent.

As a result of the problems associated with the analytical approach, simulation of the BDFE has been used to ratify whether or not the BDFE is less sensitive to channel estimation errors. The effects of minimum and non-minimum phase channels on the SER of the equalisers

has been investigated with results indicating that the BDFE does not improve over the MLSE. Fading channels have been simulated to generate results for an ensemble of minimum and non-minimum phase channels using CIR estimates derived from noisy versions of the true CIR and from LMS adapted, decision-directed channel estimation. These simulations have produced similar results.

Finally the BDFE and MLSE have been compared for channels similar to those in Chen *et al* [63]. The BDFE does not outperform the MLSE and suffers from similar problems in regard to the quality of the channel estimates affecting the SER. This is in contrast to the final results presented by Chen [63] which showed dramatic improvements obtained by the BDFE with decision feedback. The benefit of using more than one path metric to form symbol estimates is evident only when a small number of received signal samples are available but increasing the number further (BDFE without decision feedback) causes a deteriation in the SER.

The development of a hybrid version of the BDFE and MLSE has been not been continued due to the lack of performance improvement of the BDFE over the MLSE, especially with low decision delays. Thus any potential benefits of using a BDFE section to provide symbol decisions for the channel estimator and an MLSE section for the final symbol decisions will be neutralised by the insufficient performance of the BDFE section.

The next chapter examines the use of multiple channel estimators applied specifically to the MLSE. Channel estimation is at the heart of the reasons for poor equaliser performance in both the MLSE and BDFE operating on fading and frequency selective channels. A method of improving the channel estimation, particularly in tracking the channel during a data burst, is likely to have a beneficial effect on the SER. Schemes which are more robust to symbol estimation errors are of particular interest given the performance improvements of the equalisers operating with accurate transmitted symbol information, as shown in this chapter and Chapter 5.

Chapter 7

Multiple Channel Estimator Techniques

The application of multiple channel estimators to MLSE algorithms is considered in this chapter. The concepts of this technique together with novel simplifications are explained and an evaluation of the relative complexity of the algorithms is undertaken. The algorithms' performances are compared for a number of channels from a generic fading and frequency-selective test channel to HF channels and land mobile radio channels.

A feature of the multiple channel estimator (MCE) algorithm is that it makes symbol estimates with zero decision delay for the purposes of channel estimation and, rather than making hard decisions on individual symbols, the technique allows previous symbol estimates produced for the channel estimators to be updated. The result is that the problem of estimating the current channel impulse response from old symbol estimates is lessened. The 'soft' decision feature reduces the detrimental effects of incorrect symbol decisions on the tracking capabilities of the LMS adaptation algorithm.

The use of a set of channel estimators provides an element of 'diversity' in the estimates and only the better estimates are used to produce symbol decisions. The effect is that poor channel estimates, perhaps resulting from incorrect symbol estimates, are superseded by those which result in a higher 'performance index'.

The following section explains the operation of the MCE algorithm and Section 7.2 examines novel methods of simplifying the algorithm with the aim of reducing the computational complexity. Section 7.3 examines the computational complexity of the MCE, MLSE and DDFSE algorithms and Section 7.4 presents the results of simulations of the performance of the MCE algorithm together with the new variants developed.

7.1 Multiple Channel Estimator Algorithm

A method of improving the performance of the MLSE via the use of multiple channel estimators was given by Kubo *et al* [71] and Raheli *et al* [72]. These techniques employ a separate channel estimator for each state in the Viterbi trellis and update the channel estimators using zero decision delay symbol estimates.

The multiple channel estimator (MCE) technique differs from the conventional MLSE in that the transition metrics linking states in one stage of the trellis to the next are formed conditional on distinct symbol sequences and channel estimates. In the standard Viterbi trellis implementation of the MLSE, each state in the trellis corresponds to a distinct symbol sequence,

$$[\, x(k-1), \cdots, x(k-L) \,]$$

where $x(k-i)$ are the transmitted symbols and L is the channel memory. Given an M transmitted symbol alphabet, transition metrics are formed for each of the M paths arriving at a given state, one from each state with the same symbol sub-sequence, $[\, x(k-1), \cdots, x(k-L+1) \,]$ and with the same value of $x(k)$. The transition metrics are formed from the squared Euclidean distance between the received signal sample, $r(k)$, and the estimated received signal states, $\widehat{r}'(k)$:

$$\lambda = |r(k) - \widehat{r}'(k)|^2. \tag{7.1}$$

The estimated received signal states are given by:

$$\widehat{r}'(k) = \sum_{i=0}^{L} \widehat{h}_i(k-d-1)x(k-i) \tag{7.2}$$

where $x(k-i)$ are given by the symbol sequence corresponding to the transition and $\widehat{h}_i(k-d-1)$ are the channel tap estimates from the single, common channel estimator with a delay of d symbol periods. The channel tap estimates are used as an approximation to the current channel taps, $h_i(k)$.

The MCE technique replaces $\widehat{h}_i(k-d-1)$ in Eqn. 7.2 by $\widehat{h}_i^j(k-1)$ where $0 \leq j \leq (M^L - 1)$. Thus the algorithm has a separate channel estimate for each of the M^L states in the trellis and the channel estimates are formed without a delay. All transitions departing from state j use channel estimate $\widehat{h}_i^j(k-1)$, state j's corresponding symbol sub-sequence $[\, x(k-1), \cdots, x(k-L) \,]$ and the symbol $x(k)$ to form the transition metrics.

The path metrics of the M transitions arriving at each state (formed from the addition of the transition metric and the departing state's metric) are compared and the smallest is selected as the survivor transition as per the Viterbi algorithm. The survivor's path metric becomes the state's metric at the next iteration of the algorithm and the channel estimate from the survivor's departing state *replaces* the contents of the state's channel estimator.

Thus the more probable symbol sequences and channel estimates are propagated to the next stage of the trellis. The state's channel estimate is then updated by applying the LMS or RLS adaptation algorithm. In the case of the LMS algorithm the transition metric of the survivor (i.e. the error signal) is used together with the transition's symbol sequence, $[\ x(k), \cdots, x(k-L)\ ]$ and a stepsize parameter to update the channel estimate. The 'soft' decision feature of the MCE algorithm comes from the feedback of the survivor's symbol *sequence* to the state's channel estimator rather than a single symbol. Thus the symbol estimates used to update a particular channel estimate at one iteration do not necessarily correspond to the time shifted symbol estimates at the next iteration. However, it is noted that the symbol sequence used to update the channel *estimator* corresponding to a particular Viterbi state will be the same, except for the symbol $x(k-L)$, but the channel *estimate* of this estimator will have been replaced by the estimate from the channel estimator of the survivor's departing state prior to being updated.

The symbol sequence feedback mechanism allows a simplification in the adaptation of the channel estimators. In order to form the channel estimator's error signal, $e_j(k)$, the estimated received signal sample, $\widehat{r}_j(k)$, must be calculated and subtracted from the received signal sample, $r(k)$. This error signal is precisely the same as that used to form the survivor's transition metric in Eqn. 7.1, i.e.

$$\begin{aligned} e_j(k) &= r(k) - \widehat{r}'(k) \\ &= r(k) - \widehat{r}_j(k) \end{aligned} \tag{7.3}$$

as the estimated received signal state, $\widehat{r}'(k)$, is the same as the estimate of the received signal sample produced by channel estimator j.

The MCE algorithm leads to the combined most probable symbol sequences and channel estimates being propagated to the next stage of the Viterbi algorithm as the metrics are formed conditional on both a symbol sequence and a channel estimate.

The application of reduced state techniques (eg. DDFSE [34]) with multiple channel estimators is straightforward — each reduced state has its own channel estimator and the transition metrics are formed using this estimator together with the state's symbol subsequence and decision feedback sequence. The number of reduced states in the trellis is given by M^u, where $1 \leq u \leq L$. The channel estimates and symbol sequences are propagated through the trellis in a similar manner to the MCE. Additionally, adaptation of the channel estimators is performed by the same method as the MCE.

7.2 Simplification of the MCE Algorithm

This section describes a novel simplification of the MCE algorithm. For practical channels with impulse responses spanning several symbol periods and with multi-state modulation schemes such as QPSK, the number of channel estimators becomes large and the computational burden in updating these estimators via LMS or RLS adaptation algorithms is severe. The use of delayed decision-feedback sequence estimation (DDFSE) [34] or reduced state sequence estimation (RSSE) [35] techniques have been suggested to reduce the MCE algorithm's complexity [71, 72]. This complexity reduction method reduces the number of states in each stage of the trellis and thus has a corresponding reduction in the number of channel estimators as each state still has its own channel estimator.

The new MCE variant proposed, the reduced-multiple channel estimator (R-MCE) algorithm, simplifies the MCE algorithm by reducing the number of channel estimators whilst keeping the number of states in the trellis unchanged. Thus the complexity reduction is not as great as with the application of the DDFSE with multiple channel estimators (DDFSE-MCE) but, as will be demonstrated in Section 7.4, the performance loss is reduced.

In the R-MCE algorithm the trellis states are partitioned in a precise manner and each partition has its own separate channel estimator. All transitions departing from states within a given partition use the same channel estimate during calculation of the transition metrics of the Viterbi algorithm. The remainder of the algorithm, other than the survivor channel estimate propagation and the channel estimator adaptation is the same as the MCE algorithm.

Given an M-ary alphabet and a channel with an L symbol period memory, a trellis consisting of M^L states in each stage is formed as usual. The proposed R-MCE algorithm can use M^u $(0 \leq u \leq L)$ channel estimators depending on the computation / performance reduction that is acceptable: the choice of $u = L$ results in the same algorithm as described in Kubo *et al* and Raheli *et al* [71, 72]. The states are partitioned such that all paths leading to states in a channel estimator's partition have a common symbol sub-sequence $[\, x(k), \cdots, x(k-u+1)\,]$ — thus each partition has a distinct symbol sub-sequence associated to it. At each iteration of the algorithm the M^u channel estimators require individual estimates of the symbol sequence $[\, x(k), \cdots, x(k-L)\,]$ in order to update their channel estimates. A given channel estimator's symbol sequence is derived from the most probable path to any one of the states in the domain of channel estimator. All candidate paths have a common symbol sub-sequence $[\, x(k), \cdots, x(k-u+1)\,]$ and the selection determines the sequence $[\, x(k-u), \cdots, x(k-L)\,]$. In the case $u = 0$, the single channel estimator uses the most probable path in the whole trellis to form its symbol sequence for the channel estimate update.

An example is given for the case $M = 2$, $L = 3$ in Figure 7.1 which shows the mapping of states to symbol sequences and the partitioning of states to channel estimators for the cases

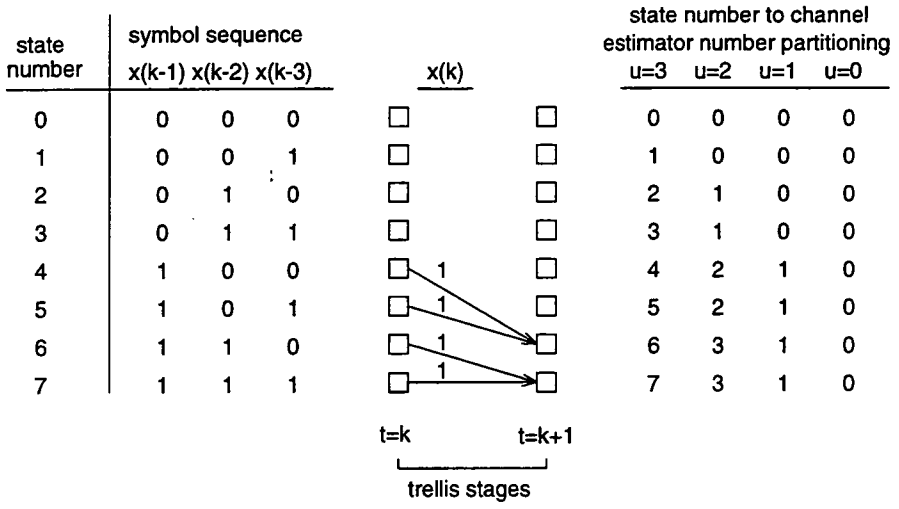

Figure 7.1: Example of state to channel estimator partitioning for a binary data alphabet and a three symbol-period memory channel.

$u = 3$, $u = 2$, $u = 1$ and $u = 0$. For the transitions shown and given $u = 2$, the incremental path metrics for paths (**state** 4→**state** 6) and (**state** 5→**state** 6) are calculated using channel estimator 2 and the symbol sequences [1,1,0,0] and [1,1,0,1] respectively. The incremental metrics for the paths (**state** 6→**state** 7) and (**state** 7→**state** 7) use channel estimator 3 and symbol sequences [1,1,1,0] and [1,1,1,1] respectively. The most probable of these four paths is selected and the channel estimate corresponding to this path's departing state replaces channel estimate 3 prior to being updated using the path's symbol sequence, error signal and associated adaptation algorithm (eg. LMS). For example, if the path (**state** 5→**state** 6) is the most probable then channel estimate 3 at time $(t = k+1)$ is given by channel estimate 2 at time $(t = k)$ after being updated using the symbol sequence [1,1,0,1] and associated error / incremental path metric, $|r(k) - r'(k)|^2$. This procedure is repeated for all channel estimators.

The method of partitioning the states to channel estimators varies from the technique in Raheli *et al* [73] which selects channel estimates from the n most probable transitions and propagates these to the next stage of the trellis. The proposed method is computationally more simple as the partitioning of states to channel estimators is fixed and searching for the most probable transitions is not required. However, the technique in Raheli *et al* [73] allows more flexibility in the number of channel estimators employed.

7.3 Complexity Comparisons

In this section, the relative computational complexity of the MCE and R-MCE algorithms together with the DDFSE-MCE algorithm is examined in order to ascertain the additional load of the MCE algorithms. The calculation of the precise computational load depends upon the hardware used to implement the algorithm. For systems employing higher symbol rates the processing may be implemented using dedicated hardware (e.g. ASICs) in order to achieve the necessary speed. Such hardware will be optimised such that the word lengths will be reduced to the minimum necessary and parameters adjusted for ease of implementation. For instance, the LMS stepsize, μ could be restricted to $\frac{1}{2^n}$ $(n \geq 0)$ so that multiplication by μ can be implemented as an n-bit shift right. If the symbol rate is lower then the algorithms may be implemented using software running on DSPs where the number and type of instructions have a significant influence. Most DSPs have a pipelined architecture providing savings in computation times by reducing the program fetch and decode cycle times. Considering these implementation factors, the complexity calculations have been expressed as the number of additions / subtractions and multiplications required to implement the core elements of the algorithms. Memory management tasks, pointer updating procedures and associated overheads have not been included as they are implementation dependent and can often be performed to some extent in parallel with the core processing functions. Exact comparisons are dependent on the device's architecture but the calculations presented should broadly reflect the relative complexity of the algorithms.

There are three main areas of the MLSE / MCE equaliser algorithms in which the core computations reside:

1. The computation of the M^{L+1} received signal states. This requires an $(L+1)$ symbol sequence to be convolved with the CIR estimate, $\underline{\hat{\mathbf{h}}}(k)$, for each received signal state.
2. The calculation of the Viterbi error metrics and the survivor selection.
3. The adaptation of the channel estimator(s), the complexity of which depends on the adaptation algorithm.

The calculations which follow assume the use of the LMS algorithm.

The reduction of the number of channel estimators in the R-MCE algorithm decreases the computational load by the fact that fewer estimators need updating. Further savings may be made in the calculation of the transition metrics as more paths use the same channel estimator allowing a more efficient computation of the estimated received signal states. The use of reduced state techniques further reduces the computational load.

7.3.1 Received Signal State Computations

The received signal state computation is a significant factor in each algorithm. If a single channel estimate is to be used then efficient methods of calculating the M^{L+1} convolution sums exist (Eqn. 7.2).

MLSE

Rather than calculating each state separately, which requires $(L+1)$ complex multiplications and L complex additions, an M-by-$(L+1)$ array may be formed with the elements of the array consisting of unique combinations of the product of the M-ary symbol alphabet and the $(L+1)$ components of the CIR estimate as shown in the table below. In this example a 4 symbol alphabet and a 4 path channel is assumed.

h_0x_0	h_1x_0	h_2x_0	h_3x_0
h_0x_1	h_1x_1	h_2x_1	h_3x_1
h_0x_2	h_1x_2	h_2x_2	h_3x_2
h_0x_3	h_1x_3	h_2x_3	h_3x_3

Each term in the product is, in general, complex valued which requires four real multiplications and two real additions. Thus the calculation of the matrix requires the following number of real multiplications and additions:

$\times$: $4M(L+1)$
$+$: $2M(L+1)$

If the real and imaginary components of the transmitted symbol alphabet consist of equal absolute valued terms (e.g. QPSK) then further savings in the calculation of the product terms may be made. Under these conditions no multiplications are necessary (terms with non-unity absolute valued terms may be scaled appropriately as the final stage of received signal state calculation). If the real and imaginary components of the transmitted symbol alphabet are symmetrical about 0 (e.g. QPSK) then the number of additions may be halved as the product terms of half the transmitted alphabet are the negative of the other half. If the symbols are purely real or imaginary valued then the number of additions required is zero and the number of multiplications is halved.

Specific combinations of the product terms in the matrix are then summed to give the received signal states. Savings may be made in these summations by adopting a 'block' or a 'tree' approach to the calculations. The 'block' method is more efficient than the 'tree' method but is not as amenable to the R-MCE algorithm requirements and does not produce large savings for relatively small values of L ($L < 4$). Both methods improve the efficiency by reducing the repetition of calculation of sub-terms of the received signal states.

The basis of the 'block' method is to split the $(L+1)$ terms into 2-component blocks resulting in $\frac{L+1}{2}$ blocks for odd values of L or $\frac{L}{2}$ blocks plus one single component block for even values of L. Each 2-component block requires M^2 complex additions to form the values of all possible combinations of the M symbols. The blocks are then formed into progressively larger groups by combining blocks together. This is illustrated in Figure 7.2(a) for $M = 2$ and $L = 5$ which shows a partial set of the symbol sequences for the received signal states partitioned into blocks and groups. The number of complex additions required for this procedure is given by:

$$\begin{array}{rl} \sum_{i=2}^{\frac{L+1}{2}} M^{2i} & - \text{ odd } L \\ M^{L+1} + \sum_{i=2}^{\frac{L}{2}} M^{2i} & - \text{ even } L \end{array}$$

Thus the total number of real additions is given by:

$$\begin{array}{rll} + : & M^2(L+1) + 2\sum_{i=2}^{\frac{L+1}{2}} M^{2i} & - \text{ odd } L \\ & M^2 L + 2M^{L+1} + 2\sum_{i=2}^{\frac{L}{2}} M^{2i} & - \text{ even } L \end{array}$$

These totals may be halved for symbols with symmetrical components.

The tree approach forms the received signal approach by calculating and storing common components in a tree fashion. The received signal states are calculated in a sequential order with common components being formed at each node in the tree starting from the 'root'. Figure 7.2(b) shows this structure for $M = 2$ and $L = 5$. The additions for a given received state are reduced by using the common component of the current and previous state as a starting point for the additions. Thus the computation of the received signal state corresponding to the symbol sequence [000110] uses the common component of the previous state (with symbol sequence [000101]) which corresponds to the partial sum resulting from the sub-sequence [0001]. The number of real additions required for this approach is given by:

$$+ : \quad 2\sum_{i=2}^{L+1} M^i$$

This figure may be halved for symbols with symmetrical components.

The following table compares the number of real multiplications and additions for the two approaches (inclusive of matrix calculations) assuming $M = 4$ and no savings due to the symbol alphabet special cases.

L	Block		Tree	
	$\times$	$+$	$\times$	$+$
1	32	48	32	48
2	48	184	48	184
3	64	608	64	704
4	80	2664	80	2760
5	96	8848	96	10960

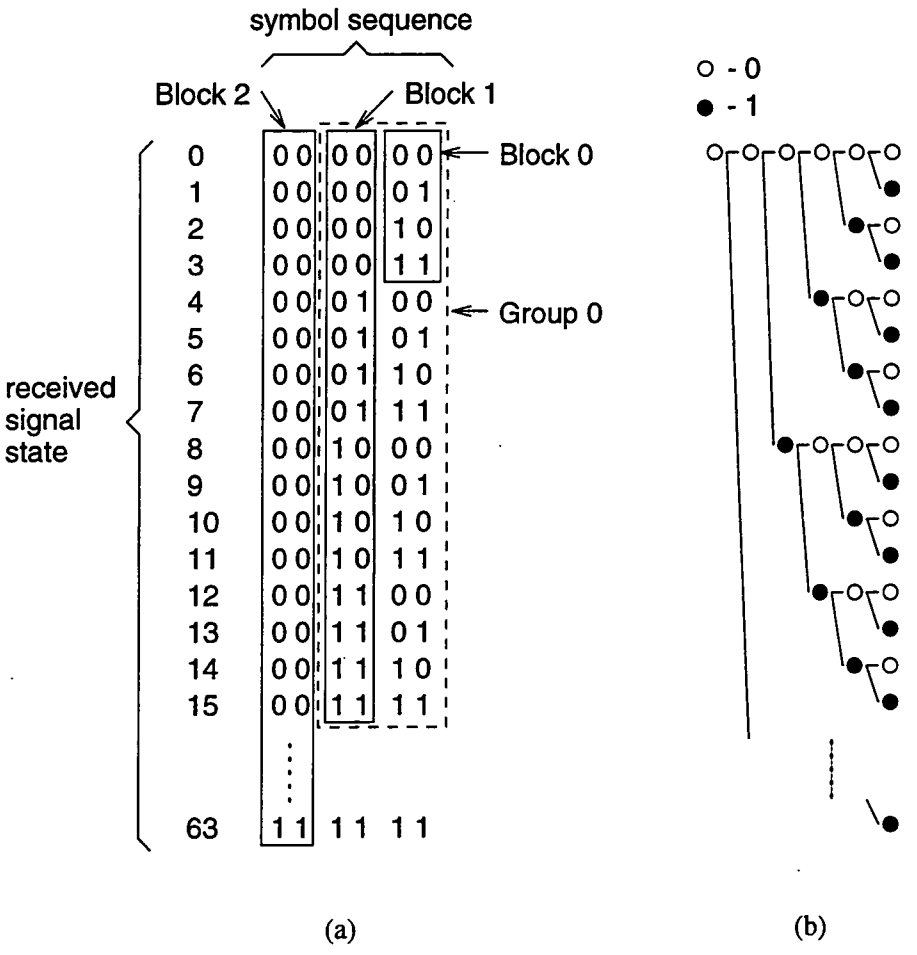

Figure 7.2: Representations of (a) block and (b) tree methods of calculating the received signal states.

MCE / R-MCE

In the MCE / R-MCE algorithm, many channel estimators are used resulting in variations in the components $h_i x_i$. This prevents the matrix approach together with block or tree addition being used. The complexity of calculating the received signal states can still be reduced but to a progressively lesser degree as the number of channel estimators increases.

The use of M^u channel estimators results in M^{L+1-u} received signal states using the same channel estimator — see Figure 7.1. The partitioning of the trellis states to channel estimators results in a common term of u components corresponding to the symbol subsequence $[x(k-1),\ldots,x(k-u)]$ for each channel estimator. These common terms require a total number of real computations:

$\times$: $4uM^u$
$+$: $(4u-2)M^u$

The number of multiplications can be halved for purely real or imaginary symbols and is zero for symbols with real / imaginary terms of the same absolute value.

The calculation of the received signal states requires a further $(L+1-u)$ components which can be simplified by forming an M-by-$(L+1-u)$ matrix of components per channel estimator. This requires a total number of real computations:

$\times$: $4M^{u+1}(L+1-u)$
$+$: $2M^{u+1}(L+1-u)$

The number of multiplications can be halved for purely real or imaginary symbols, halved for symmetrical symbol terms and is zero for symbols with real / imaginary terms of the same absolute value. The number of additions can be halved for symmetrical symbol terms.

The final stage of adding up the common component and the 'matrix' components in a 'tree' fashion results in the following number of real additions:

$+$: $2M^u \sum_{i=1}^{L+1-u} M^i \quad u > 0$
$\quad\; 2M^u \sum_{i=2}^{L+1-u} M^i \quad u = 0$

The following table demonstrates the complexity for $L = 3, 4$ and $M = 4$:

L	$u=0$		$u=1$		$u=2$		$u=3$		$u=4$	
	$\times$	$+$	$\times$	$+$	$\times$	$+$	$\times$	$+$	$\times$	$+$
3	64	704	256	376	640	992	1792	1664	n/a	n/a
4	80	2760	272	2856	896	3168	2816	4224	8192	7680

DDFSE-MCE

The DDFSE has a reduced number of states at each stage in the trellis and a separate feedback vector for each reduced state containing estimates of the remainder of the transmitted

symbol sequence (see Section 3.4.2). The application of multiple channel estimators leads to one channel estimator per reduced state with the number of reduced states (and channel estimators) being given by M^u where $1 \leq u \leq L$, resulting in the feedback vectors having $(L-u)$ components. The contents of the feedback vectors are, in general, different and so the simplifications used in the calculation of the received signal states cannot be applied. A separate convolution sum is required for each trellis state plus the additions of the component due to the symbol $x(k)$. This leads to the following real computations:

$$\begin{aligned} \times &: 4M^u(L+M) \\ + &: M^u(4L+4M-2) \end{aligned}$$

The number of multiplications is zero for symbols with real / imaginary terms of the same absolute value and is halved for symbols with purely real or imaginary components. The number of additions becomes $M^u(3L+3M-2)$ for the latter condition.

7.3.2 Viterbi Algorithm Computation

The Viterbi algorithm computations are identical for the MLSE and R-MCE algorithms but are less for the DDFSE-MCE. The transition and path metrics are formed the received signal and the received signal states and the survivors to each state decided by comparing the candidate path metrics. For the MLSE and R-MCE the number of real computations are:

$$\begin{aligned} \times &: M^L(2M) \\ + &: M^L(5M+1) \end{aligned}$$

and for the DDFSE-MCE:

$$\begin{aligned} \times &: M^u(2M) \\ + &: M^u(5M+1) \end{aligned}$$

These calculations assume the compare functions can be implemented by a subtraction (equivalent to an addition).

The final stage of finding the minimum path metric requires M^L compares (real additions) for the MLSE and R-MCE and M^u for the DDFSE-MCE.

7.3.3 Channel Estimator Adaptation Computation

The channel estimator(s) are assumed to be adapted by the LMS algorithm. The estimated received signal needs to be calculated, the error signal formed and the channel estimate updated. In the case of the R-MCE and DDFSE-MCE the error signal is the same as the transition metric of the survivor symbol sequence and thus the signal need not be recalculated. The MCE techniques require additional compares to find the minimum path metric of all states in the domain of the channel estimator.

The MLSE computational load is given by (real computations):

$\times$: $10(L+1)$

$+$: $8(L+1)+2$

The number of multiplication is reduced to $2(L+1)$ for symbols with real / imaginary terms of the same absolute value.

The R-MCE and DDFSE-MCE require the following real computations:

$\times$: $M^u(6L+6)$

$+$: $M^u(4L+4)+uM^{L-u}$

The number of multiplication is reduced to $M^u(2L+2)$ for symbols with real / imaginary terms of the same absolute value.

7.3.4 Examples of Total Computational Complexity

The total complexity of the three algorithms are compared for $L = 3$ and 4, $M = 4$ and for the cases of a general transmitted symbol alphabet and for QPSK in the tables below. In addition to the number of real computations required the relative complexity compared to the MLSE is given below the number of computations for each algorithm. This relative complexity is expressed simply as a multiplicative factor.

The following table lists the load of the MLSE:

L	symbol alphabet	MLSE $\times$	MLSE $+$
3	general	616	2050
		1.00	1.00
	QPSK	520	1746
		1.00	1.00
4	general	2178	8338
		1.00	1.00
	QPSK	2058	7006
		1.00	1.00

The R-MCE loads are:

L	symbol alphabet	R-MCE									
		$u=0$		$u=1$		$u=2$		$u=3$		$u=4$	
		×	+	×	+	×	+	×	+	×	+
3	general	600	2124	816	2264	1536	2540	3840	4099	n/a	n/a
		0.97	1.04	1.32	1.10	2.50	1.24	6.23	2.00	n/a	n/a
	QPSK	520	2112	544	2216	640	2412	1024	3843	n/a	n/a
		1.00	1.21	1.05	1.27	1.23	1.38	1.97	2.20	n/a	n/a
4	general	2158	8412	2440	8632	3424	9152	5760	11148	17920	18438
		0.99	1.01	1.12	1.04	1.57	1.10	2.64	1.34	8.23	2.21
	QPSK	2058	8392	2088	8568	2208	8960	2688	10636	4608	17414
		1.00	1.20	1.01	1.22	1.07	1.28	1.31	1.52	2.24	2.49

Finally the DDFSE-MCE computations are:

L	symbol alphabet	DDFSE-MCE							
		$u=1$		$u=2$		$u=3$		$u=4$	
		×	+	×	+	×	+	×	+
3	general	240	272	960	1032	3840	4099	n/a	n/a
		0.39	0.13	1.56	0.50	6.23	2.00	n/a	n/a
	QPSK	64	244	256	920	1024	3651	n/a	n/a
		0.12	0.14	0.49	0.53	1.97	2.09	n/a	n/a
4	general	280	352	1120	1184	2480	4620	17920	18436
		0.13	0.04	0.51	0.14	1.14	0.55	8.23	2.21
	QPSK	72	320	288	1056	1152	4108	4608	16388
		0.03	0.05	0.14	0.15	0.56	0.59	2.24	2.34

From these tables it can be seen that the complexity of the R-MCE is considerably reduced compared to that of the MLSE when QPSK data is transmitted due to the simplifications possible when calculating the received signal states. With $u = L$ (giving the largest number of channel estimators) the complexity increases by a factor of around 2 for $L = 3$ and by a factor of around 2.4 for $L = 4$. The DDFSE-MCE has a significant decrease in complexity compared to the R-MCE except when $u = L$ (i.e. no reduction in complexity). This decrease is due to the smaller number of states within the trellis.

The scale of the complexity increases for the R-MCE is not huge and thus the implementation of the algorithm is feasible for moderate symbol rate systems.

7.4 Simulation of MCE Performance

Comparisons of the symbol error rates obtained for the MLSE, R-MLSE and DDFSE-MCE algorithms have been simulated for a number of different channel memory lengths, power-delay profiles and fade rates. The following sections detail the simulated systems and channels and present the results of the simulations.

7.4.1 Equipower / Tapered Power-Delay Profile Channels

The systems and channels simulated in this section are similar to those simulated in Chapter 5. The data was QPSK modulated at a symbol rate ($1/T_s$) of 100kBaud and transmitted with a frames structure 100:20 (data burst length : training sequence length) after an initial 500 symbol training sequence. The channel was simulated using one sample/symbol and thus assumes ideal Nyquist transmit and receive filters. The channels simulated had 3, 4 or 5 independent, complex, T_s-spaced paths with either equipower paths or with a tapered power-delay profile. Each path underwent Rayleigh fading, varying with a classical Doppler power spectrum and maximum Doppler spread of $f_d = \pm 400$Hz. This Doppler spread corresponds to a mobile velocity of 480km/h or 240km/h for RF carrier frequencies of 900MHz and 1800MHz respectively giving a product $f_d T_s = 0.004$.

Figure 7.3 plots the symbol error rate versus SNR for the standard MLSE and the R-MCE algorithm with estimator reduction using 16, 4 and 1 channel estimators corresponding to $(u = 2, \cdots, 0)$. The curve $u = 2$ corresponds to the MCE algorithm, i.e. no reduction in the number of channel estimators. The channel had three equipower paths and the channel estimator used three T_s-spaced taps and was trained and tracked using the LMS algorithm with a stepsize of $\mu = 0.1$. All equalisers had a decision delay of two symbol periods which was found to give the best performance for the MLSE algorithm operating in the decision-directed tracking mode.

Figures 7.4 and 7.5 show the results for the channels with 4 and 5 equipower paths respectively. The channel estimators used 4 and 5 taps respectively and were adapted with the LMS algorithm with a stepsize of 0.1. For both these examples the equaliser decision delay was set to three symbol periods which gave the best performance for the MLSE algorithm (decision-directed tracking).

These curves show that the R-MCE algorithm can outperform the MLSE operating in a decision-directed tracking mode and the benefits of employing a number of channel estimators tracked using zero decision delay together with a 'soft' decision capability are apparent. Reducing the number of channel estimators in the R-MCE algorithm degrades the symbol error rate of the R-MCE and the performance drops below the conventional MLSE if the number of channel estimators drops below four. The R-MCE ($u > 1$) technique provides an

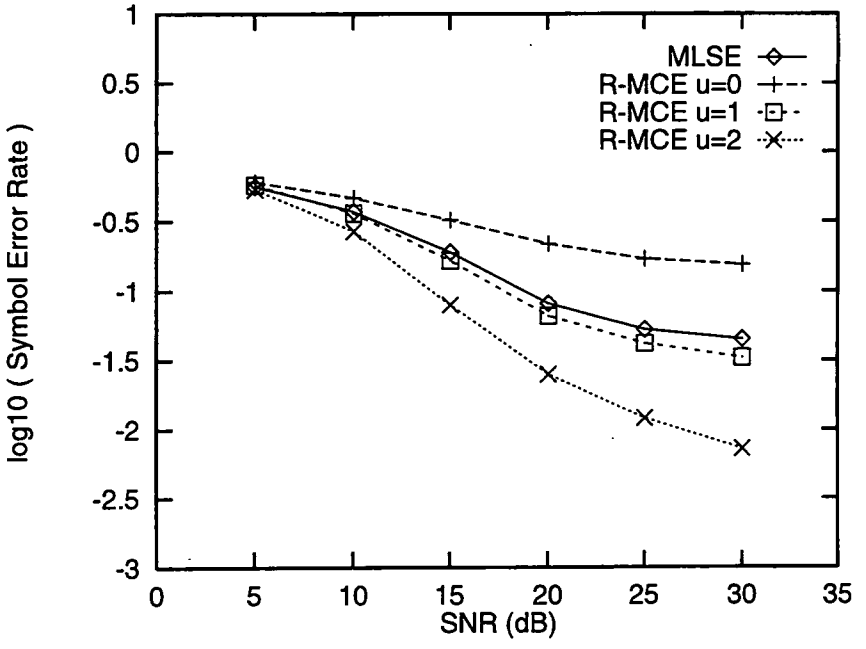

Figure 7.3: Symbol error rate vs. SNR for the MLSE and R-MCE for a 3 equipower path channel ($L = 2$).

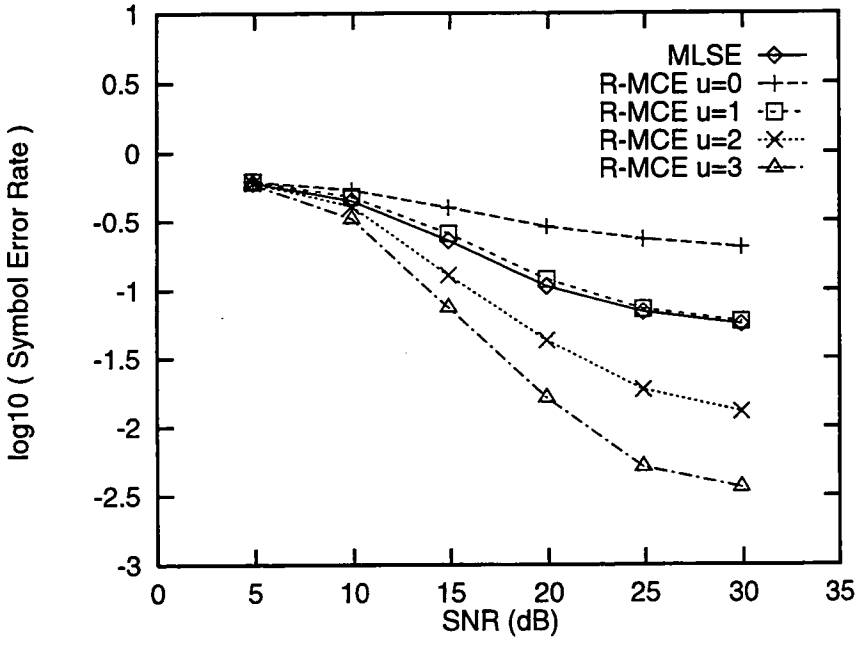

Figure 7.4: Symbol error rate vs. SNR for the MLSE and R-MCE for a 4 equipower path channel ($L = 3$).

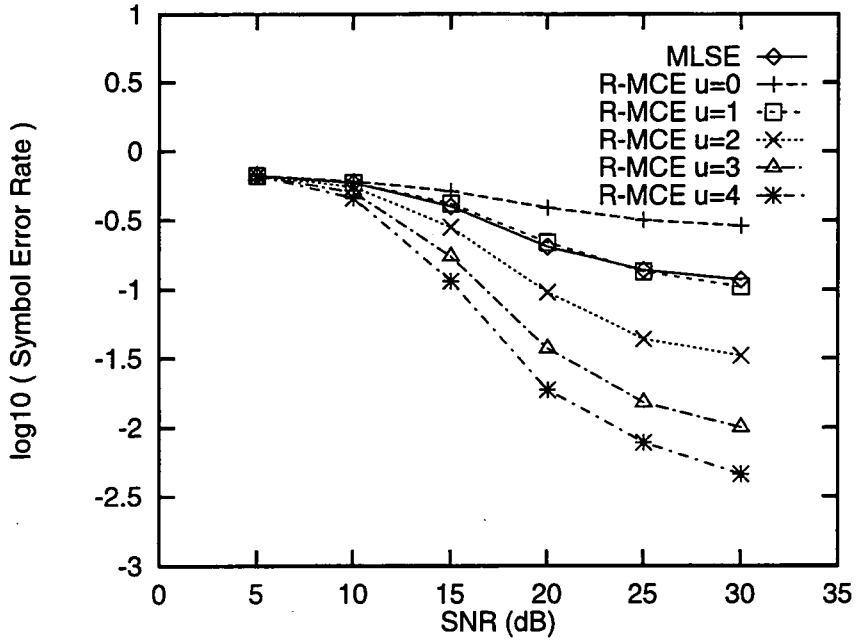

Figure 7.5: Symbol error rate vs. SNR for the MLSE and R-MCE for a 5 equipower path channel ($L = 4$).

improvement in the symbol error rates particularly for SNRs greater than 15dB. The R-MCE $u = 0$ uses a single channel estimator as does the MLSE but its SER is higher. The two algorithms are not equivalent as the R-MCE produces its decisions with zero decision delay and is capable of updating previous decisions since it feeds a symbol sequence estimate back to its channel estimator rather than a single symbol estimate.

Figures 7.6 and 7.7 show the results for channels with tapered power-delay profiles and with 4 and 5 paths respectively. The relative powers of the paths are $[\ 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}\ ]$ and $[\ 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}\ ]$. All paths are complex, symbol-period spaced and fade independently with a classical Doppler spectrum and maximum Doppler spread of $f_d = \pm 200$Hz. The channel estimators used 4 and 5 taps respectively and were adapted using the LMS algorithm with a stepsize of 0.1. The decision delay is 6 symbol periods which optimises the standard MLSE algorithm. The curves again show that the R-MCE technique can outperform the MLSE and provides a reasonable perfomance / complexity trade-off for $u > 2$.

Figure 7.8 shows the relative performance between MLSE, DDFSE-MCE and the R-MCE algorithm for the same channel used in Figure 7.4. The MLSE has 64 states and uses one channel estimator, the R-MCE has 64 states and 16 channel estimators and the DDFSE-MCE has 16 states and channel estimators. The performance loss due to the reduction in the number of states in the DDFSE-MCE algorithm is apparent. This result is characteristic of the relative performance of the R-MCE and DDFSE-MCE — for a given reduction in the

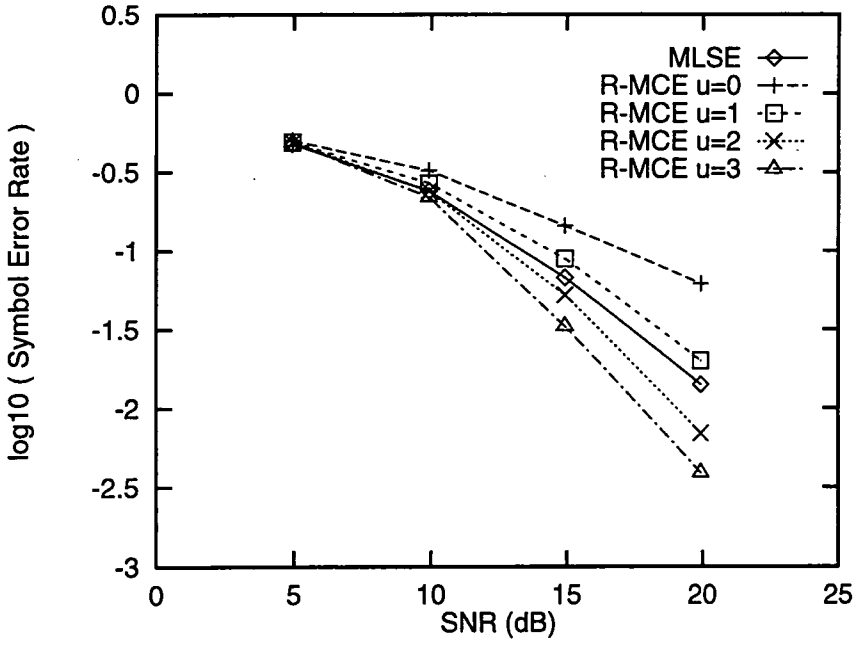

Figure 7.6: Symbol error rate vs. SNR for the MLSE and R-MCE for a 4 path channel with a tapered power-delay profile ($L = 3$).

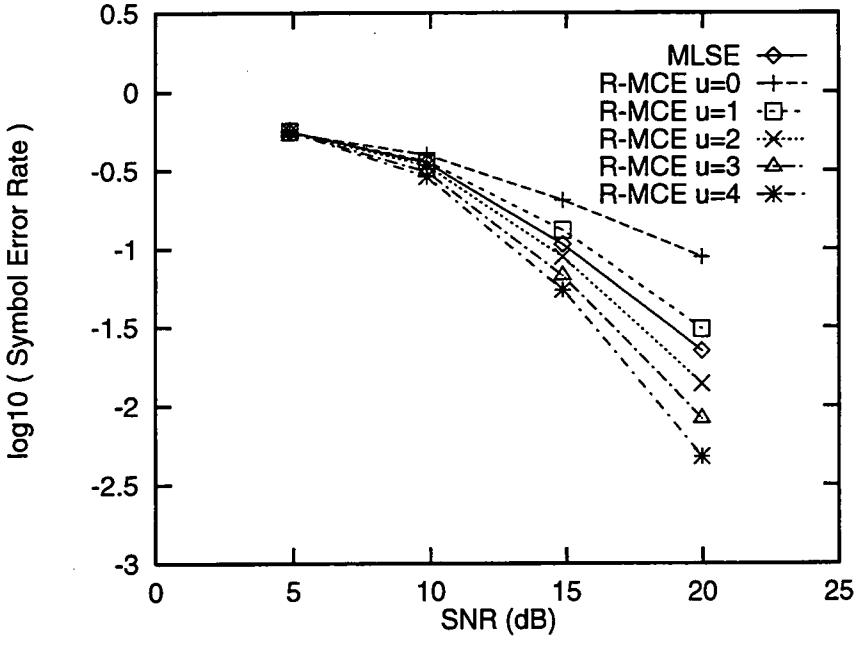

Figure 7.7: Symbol error rate vs. SNR for the MLSE and R-MCE for a 5 path channel with a tapered power-delay profile ($L = 4$).

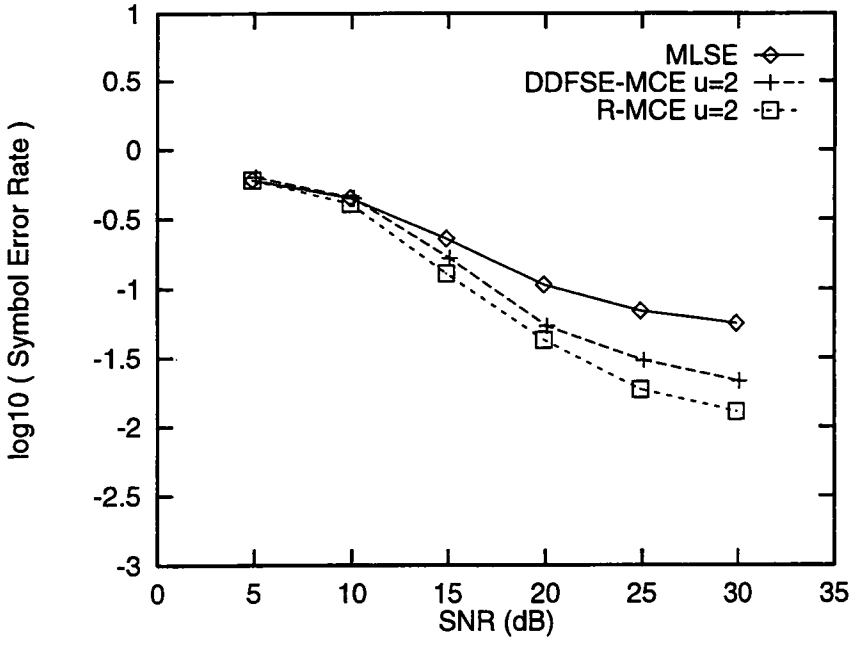

Figure 7.8: Symbol error rate vs. SNR for the MLSE, R-MCE and DDFSE-MCE for a 4 equipower path channel ($L = 3$). The R-MCE algorithm uses 16 channel estimators and the DDFSE-MCE uses 16 channel estimators and states.

number of channel estimators / states, the R-MCE outperforms the DDFSE-MCE.

Further results on the R-MCE algorithm show that if the fade rate drops then the performance of the technique does not give any worse performance than the MLSE, but tends to maintain an improvement. This result assumes that all other parameters (eg. equaliser decision delay and LMS stepsize) remain constant. For a time-invariant channel, reducing the stepsize and increasing the equaliser decision delay significantly improves the MLSE performance beyond that of the R-MCE techniques which feed symbol estimates with zero decision delay into their channel estimators. The reduced decision delay trade-off between enhanced tracking and poor symbol estimates is no longer applicable as the channel is time invariant. As the fade rate drops, increasing the decision delay of the MLSE is beneficial and the MLSE's SER improves over that of the R-MCEs.

Figure 7.9 shows the build-up of the normalised mean tap error over the duration of the data burst for a number of equaliser configurations. The system and channel simulated were the same as those in Figure 7.4 and the results averaged over 1000 data bursts. The two MLSE curves show the increase in MTE for the decision-directed tracking mode and for tracking with the delayed true (transmitted) data. The remaining curves show the MTE for the four configurations of the R-MCE equaliser. The MTE is derived from the channel estimator corresponding to the current state with minimum path metric. Thus the same channel estimator is not necessarily compared over the duration of the data burst or simulation. The

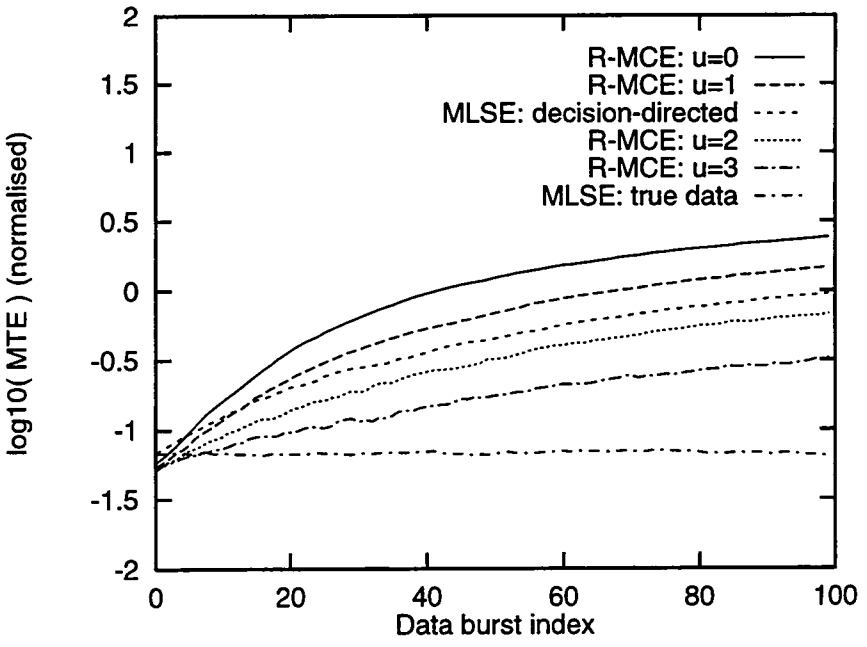

Figure 7.9: Normalised mean tap error vs. data burst index for MLSE and R-MCE equalisers. $L = 3$, T_s-spaced equipower paths, $f_d = \pm 400$Hz.

MTE is lower for the R-MCE with $u = 2$ and 3 which also give lower SERs (Fig. 7.4). The reduction in the channel estimtator error is seen to translate into a lower SER.

7.4.2 HF Channels

Given the positive results obtained in Section 7.4.1 the performance of the R-MCE algorithms operating on an HF channel are investigated. The system transmits QPSK data at 2.4kBaud with a frame structure of 40:40 (data burst length : training sequence length). Prior to the first data burst a sequence of 500 training symbols are transmitted in order for the channel estimator in the receiver to acquire an initial estimate of the channel. The data symbols are passed through a root raised cosine filter with roll-off factor of $\alpha = 0.5$ prior to transmission. The channel is simulated at either a factor $q = 2$ or $q = 4$ times the symbol rate depending on the precise channel multipath delays used. In the receiver the signal is corrupted by AWGN and filtered through a root raised cosine filter with $\alpha = 0.5$ and downsampled to the symbol rate assuming perfect symbol timing synchronisation.

The channel is model using the two equipower path model with relative time delays of either 0.52ms or 1.042ms. These values were chosen as the closest values to the CCIR recommended values of 0.5ms and 1.0ms given the symbol rate and upsampling factors [14]. With a symbol period of $\approx$ 0.418ms ($\frac{1}{2.4\times10^3}$) these correspond to a delay of 1.25 and 2.5 symbol periods respectively. The two paths vary independently with the time variations in

the gain and phase shift being modelled by passing complex, white Gaussian noise through a filter with Gaussian shaped frequency response with a one sided 3dB bandwidth of either 0.5Hz, 2Hz or 10Hz. The $f_d T_s$ product for these values of fading are 0.0002, 0.0008 and 0.004. These may be compared to those of the simulations in the previous section (0.004 and 0.002) to provide a reference for the relative speed of fading of the HF channels. The complex path gains were generated at a sample rate of 50Hz and interpolated by a factor of 96 or 192 to the channel simulation rates of 4.8kHz ($\frac{2}{T_s}$) and 9.6kHz ($\frac{4}{T_s}$) respectively.

For the 10Hz fading channel and the 40 symbol data burst leads to an approximate 0.17 fades/burst compared to 0.4 fades/burst for the 400Hz, 4 path fading channel used in the previous section (100kBaud symbol rate and a 100 symbol data burst). Thus, on average, there will be fewer individual path fades per data burst with the HF channel but, due to the presence of only two paths, the received signal is marginally more likely to go through a deep fade due to both paths fading simultaneously.

Figures 7.10, 7.11 and 7.12 show the SER vs. SNR for the MLSE (decision-directed tracking) and R-MCE equalisers operating on the channel with 1.042ms path delay and 10Hz, 2Hz and 0.5Hz fading respectively. The channel estimator used four symbol spaced taps (covering a total of 1.667ms of delay) and was adapted using the LMS algorithm with $\mu = 0.1$. The receive filter spreads the received energy around the two paths leading to energy lying inbetween and outside the nominal path delays. The choice of μ gives relatively good results and was determined by trial and error. The decision delays of the equalisers were 5, 9 and 11 symbol periods respectively which gave the best results for the decision-directed MLSE.

The results shown in Figure 7.10 show that the R-MCE outperforms the MLSE for all values of u. The absolute decrease in SER is not as large as the results of the previous section but nevertheless the performance improvement is appreciable for $u > 1$ and for an SNR $>$ 15dB. As the fade rate decreases, the relative performance of the R-MCE is reduced below that of the MLSE but it is noted that in these simulations the decision delay is optimised for the decision-directed MLSE. The increase in decision delay provides a better trade-off between estimating an old channel and feeding erroneous symbol estimates to the channel estimator. As the R-MCE feeds back symbol estimates with zero decision delay the benefits are negated by poor symbol estimates and the fact that the channel is fading slowly. The benefits of the R-MCE are present only for the more rapidly fading channel and if the system is designed to operate optimally (i.e. with low decision delay) for such conditions, a drop in the fade rate still leads to the R-MCE having a lower SER than the MLSE given that the decision delay is kept constant. However, this system design factor depends on the relative likelihood and duration of such high fade rates compared to the lower rates and whether the system is to be designed so that an increase in SER can be tolerated when the fade rate is high or that the system is operating sub-optimally when

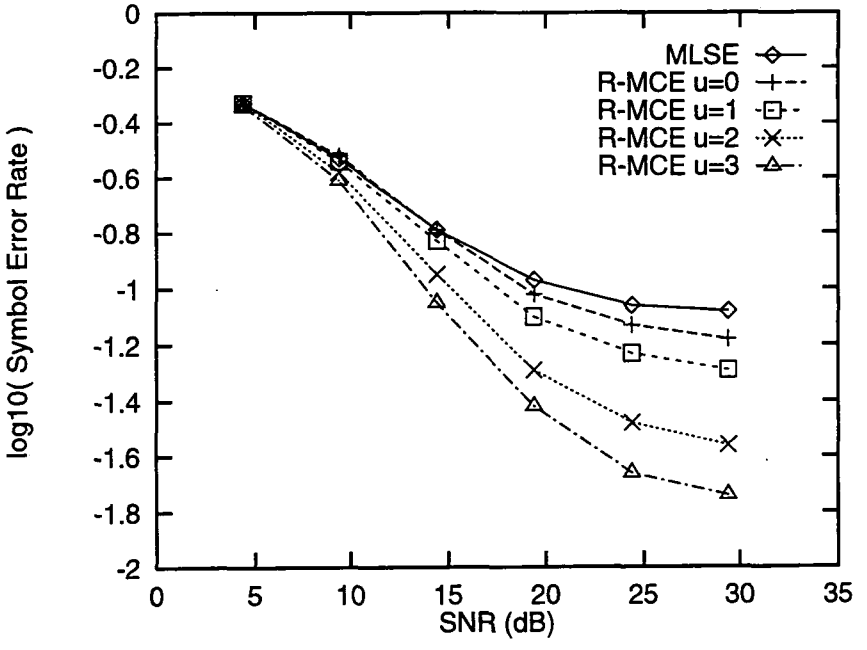

Figure 7.10: SER vs. SNR for a 2 path HF channel with multipath delay spread = 1.042ms and 10Hz fading.

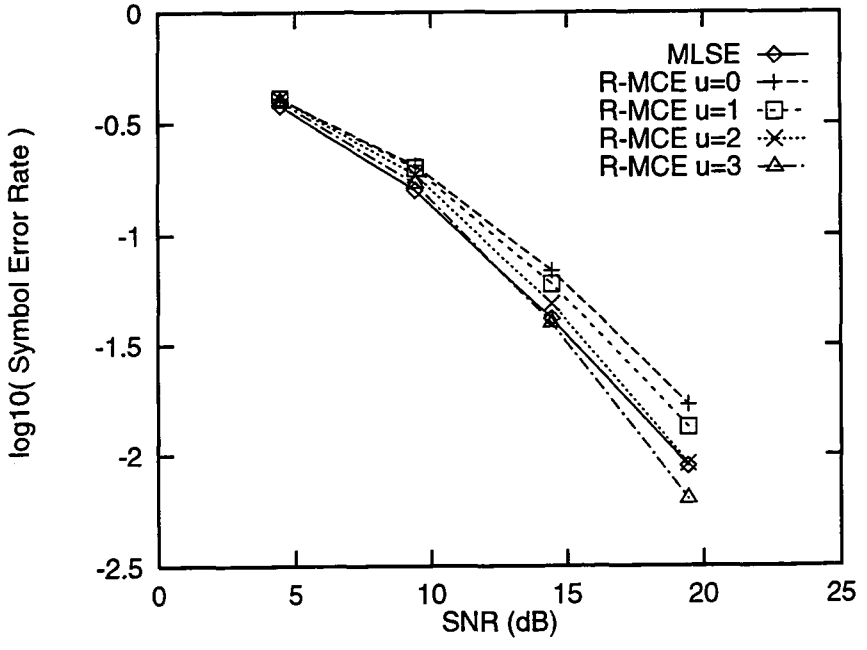

Figure 7.11: SER vs. SNR for a 2 path HF channel with multipath delay spread = 1.042ms and 2Hz fading.

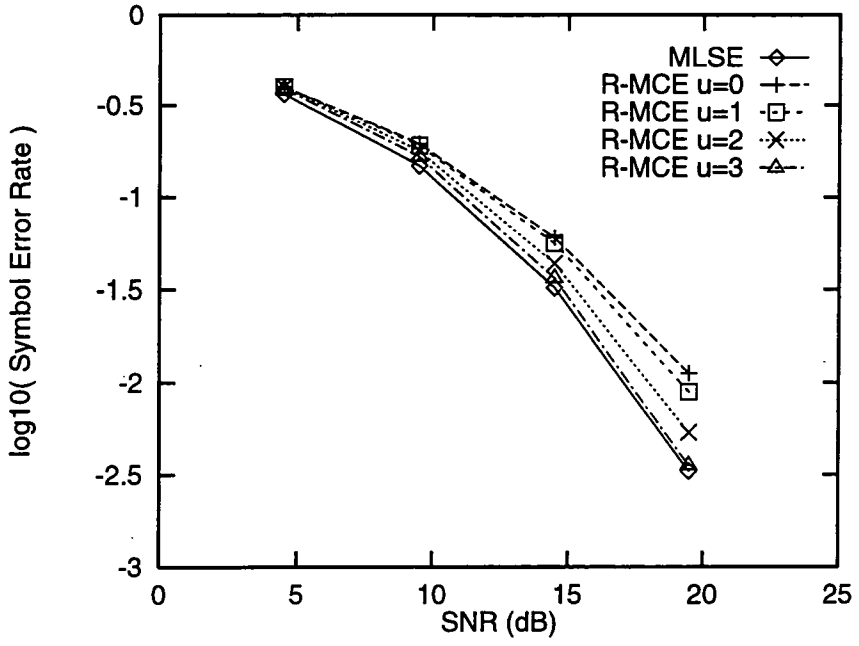

Figure 7.12: SER vs. SNR for a 2 path HF channel with multipath delay spread = 1.042ms and 0.5Hz fading.

the fade rate is low. For the HF channel the relative frequency of the fade rates is not well known and is dependent on the link being used but some results are available in [74, 75] indicating that Doppler spreads in the order of 10Hz do occur.

Figure 7.13 shows the SER for equalisers operating on the 1.042ms, 0.5Hz channel with an SNR $\approx$ 15dB as a function of decision delay. As the decision delay increases the decision-directed MLSE has a relative increase in performance compared to the R-MCE and outperforms the $u = 3$ equaliser for delays greater than 8 symbol periods. Thus if the system has a fixed, low decision delay then the R-MCE outperforms the MLSE even when the fade rate drops but not be a significant margin.

Figure 7.14 shows the results of the final simulation which uses the channel with 0.52ms multipath delay spread and 10Hz fading corresponding to the CCIR flutter fading model [14]. The channel was simulated at four times the symbol rate and the receiver channel estimator modelled the channel with three T_s-spaced paths and the estimator was adapted via the LMS algorithm with $\mu = 0.1$. It is seen that the R-MCE outperforms the MLSE for $u \geq 1$ but again the absolute SER improvement is small for an SNR $<$ 15dB.

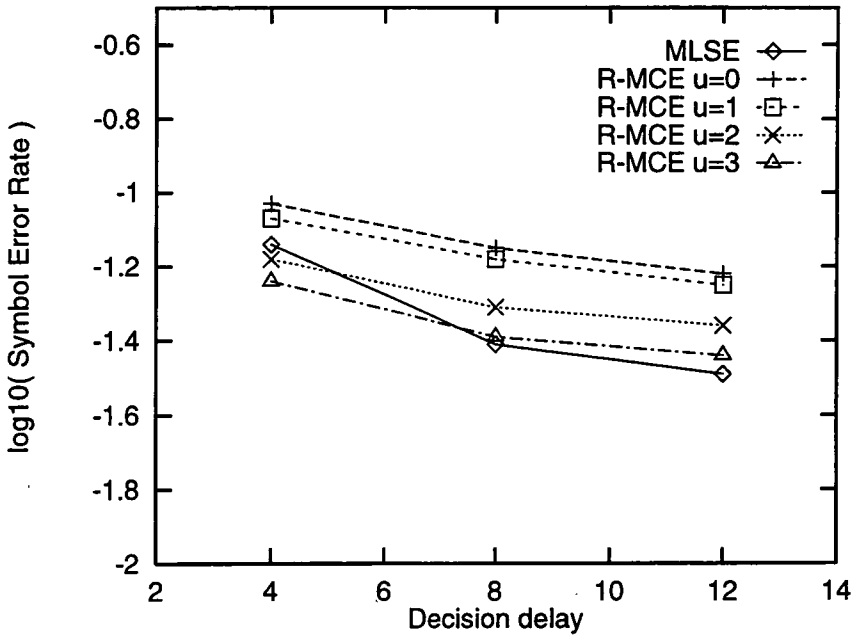

Figure 7.13: SER vs. decision delay for a two path HF channel with multipath delay spread = 1.042ms and 0.5Hz fading.

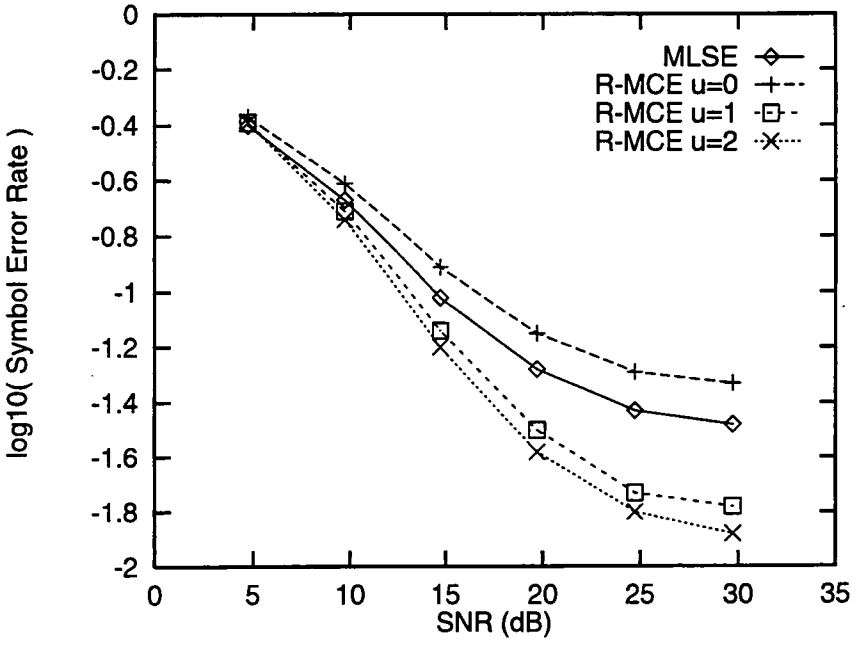

Figure 7.14: SER vs. decision delay for a two path HF channel with multipath delay spread = 0.52ms and 10Hz fading.

7.4.3 TETRA QS/EQ Channel

The Trans European Trunked RAdio (TETRA) standard is the European standard for digital private mobile radio (PMR) — see Appendix A. The channel simulated in this section is defined as either a test channel (EQ) or when the mobile is operating in a quasi-synchronous (QS) mode whereby more than one base station transmits the same information to the mobile receiver. The synchronisation errors in transmitting the data due to varying propagation times between the base stations and the mobile leads to an increased multipath delay spread. The simulated system modulates the data via QPSK at a symbol rate of 18kBaud as opposed to the TETRA specification of $\frac{\pi}{4}$-DQPSK as coherent demodulation with perfect symbol timing synchronisation is assumed. Both transmitter and receiver employ root raised cosine filters with $\alpha = 0.35$. The frame structure is slightly different to that used in the standard [76] due to the relatively long data bursts compared to the symbol period and the fact that in the 'normal continuous downlink' frame structure (used in the QS mode), training sequences are provided at each end of the burst in addition to mid-burst. This allows the data sequence to be detected both forwards and backwards i.e. by applying the equalisation algorithms to the received signal samples both in the received and reversed time sequences. This results in a larger delay in the detection process but still occurs in real-time. Thus the data burst can be effectively simulated by halving it in duration and only equalising the received signal sequence in the forward direction. The affect on the SER if the full length data burst is equalised in the forward direction only, giving a high number of fades/burst, is demonstrated. The resulting frame structure is 54:11 (data burst length : training sequence length). Prior to the first data burst 500 training symbols are transmitted to enable the channel estimator to accquire an initial estimate of the channel.

The channel model has four paths spaced with relative delays of 0μs, 11.6μs, 73.2μs and 99.3μs respectively. The channel is modelled in these simulations at a rate $q = 16$ times the symbol rate, leading to the relative delays being approximated by 0μs, 10.4μs, 72.9μs and 100.7μs. With a symbol period of $\frac{1}{18\times10^3}$s $\approx 55.6\mu$s the first two paths occur within the first symbol period and the second two paths within the second symbol period.

The mobile velocity is assumed to be 200km/h and, at a carrier frequency of 400MHz, the maximum Doppler shift of the signal is $\approx$ 70Hz. The time variations in the baseband path's complex gains are modelled by passing white Gaussian noise, sampled at 1kHz, through a filter with a classical Doppler frequency spectrum and interpolated by a factor of 288 to the channel sample rate (288KHz). The relative power of the four paths are 0dB, 0dB, -10.2dB and -16.0dB

With 70Hz fading and a data burst length of 54 symbols (3ms duration), the individual paths have an average of 0.21 fades/burst (c.f. 0.4 fades/burst for the equipower path channel example and 0.17 fades/burst for the HF channel example).

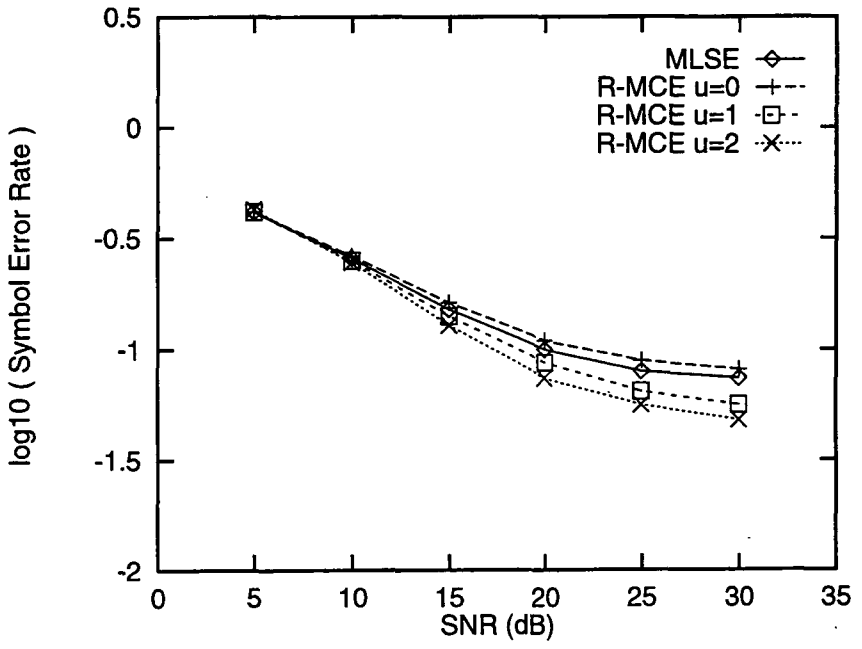

Figure 7.15: SER vs. SNR for the TETRA QS/EQ200 channel model with a 3 T_s-spaced path channel estimator.

Figure 7.15 shows the comparative SER vs. SNR for the MLSE and R-MCE equalisers. The channel estimators used three T_s-spaced paths and were adapted using the LMS algorithm with $\mu = 0.1$. The equalisers had a decision delay of one symbol period which gave the best results for the decision-directed MLSE. The results show that the R-MCE gives an improvement for $u \geq 1$ but at an SNR = 17dB — the designed SNR ($= E_b/N_0$) for use in the TETRA standard, the improvement is very marginal. As the MLSE is optimised with a very short decision delay, the gains in making the channel estimators decision with zero delay for the R-MCE is negligible and the improvements gained are due to the 'soft' decision characteristics. As the majority of the symbol energy arrives within the first symbol period, the symbol sequence detection mechanisms are only of limited use as the amount of ISI is relatively low.

Figure 7.16 plots the SERs for 70Hz fading channel but with the channel estimator taps covering 167μs (4 T_s-spaced path taps) as opposed to 111μs in the previous figure. This system was simulated to investigate whether the three path channel estimator spanned the effective channel (the channel plus the transmit and receive filter impulse responses) sufficiently. Again the break even point is with $u \geq 1$ but the absolute improvement in SER is negligible indicating that the three path channel model is sufficient. The SERs obtained if the channel estimator uses a two T_s-spaced path model are higher as the channel estimtor does not cover the span of the multipath delay spread. However the R-MCE $u = 1$ still improves over the MLSE.

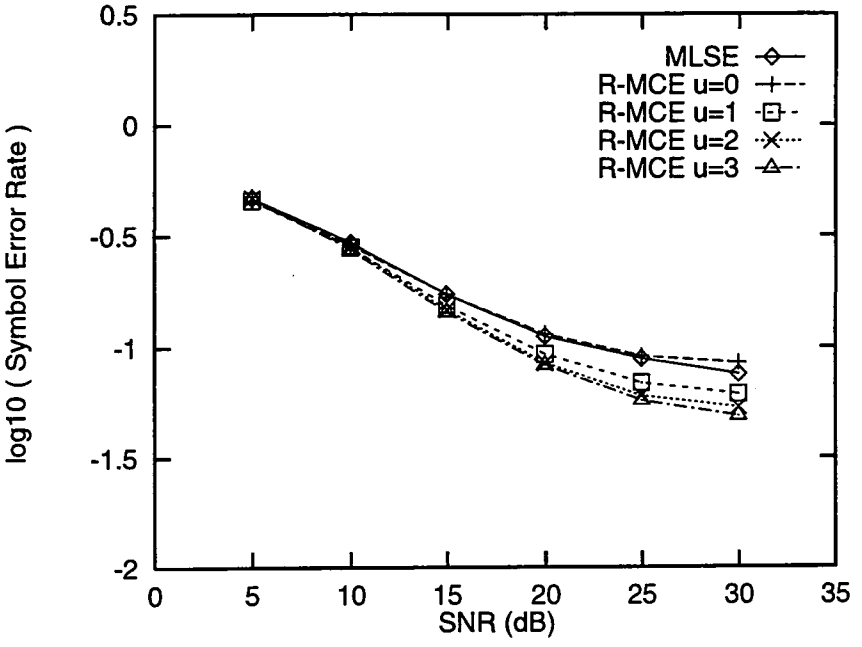

Figure 7.16: SER vs. SNR for the TETRA QS/EQ200 channel model with a 4 T_s-spaced path channel estimator.

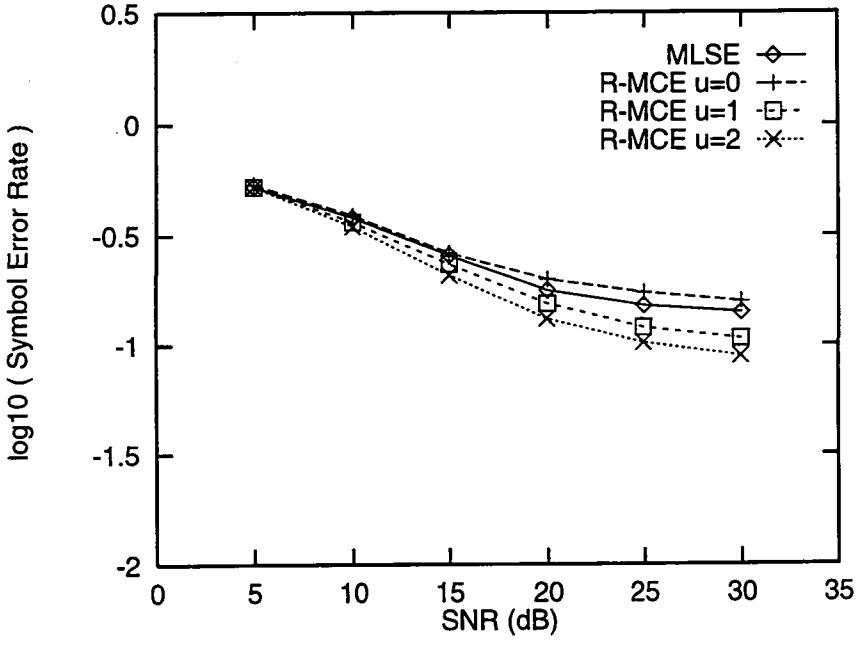

Figure 7.17: SER vs. SNR for the TETRA QS/EQ200 channel model using a frame structure of 104:11.

The final simulation demonstrates the effects of increasing the data burst length to the full 108 symbol duration whilst maintaining an 11 symbol training sequence. This gives a data burst of 6ms and each path has an average of 0.42 fades/burst which is much closer to the simulations in Section 7.4.1 with $f_d = \pm 400$Hz. All other system parameters were held constant. Figure 7.17 shows that the relative performance of the R-MCE compared to the MLSE remains the same but that the absolute SER increases for all the equalisers.

7.5 Summary

The performance gains obtained by using multiple channel estimators, instead of a single estimator, to generate the estimated received signals for the MLSE equaliser have been demonstrated. The algorithm results in the path metrics being calculated conditional on separate symbol sequences and channel estimates with the most probable sequences and estimates being propagated to the next stage of the Viterbi algorithm. In addition the symbol decisions fed to the channel estimator can be updated creating a soft decision mechanism.

The computational complexity of this algorithm is relatively large and so an alternative method of reducing the complexity has been developed. This technique reduces the number of channel estimators whilst keeping the number of trellis states constant and allocates the states to the estimators in a predetermined manner in contrast to alternative complexity reduction methods. The new algorithm (the R-MCE) provides a range of trade-offs between performance and complexity with greater improvements being obtained with higher fade rates and channels with longer multipath delay spreads. These types of channels are more common as the carrier frequency and symbol rate employed increases. The performance decrease as the number of channel estimators is reduced is not as great as that obtained with the DDFSE-MCE algorithm which reduces both the number of channel estimators and trellis states. For the channels simulated, the number of estimators required before the performance drops below that of the MLSE is either 4 or 16 ($u = 1$ or 2). These figures are obtained by choosing an equaliser decision delay which results in the best performance of the MLSE. In practice the decision delay may well be set to optimise the performance for the highest fade rate expected allowing the maximum complexity reduction permitted to be determined.

The R-MCE achieves its performance improvements partly through the use of zero decision delay for the channel estimator symbol decisions and partly due to the soft decision capability. The mean tap error of the channel estimator is reduced by these methods and this has a corresponding affect on the symbol error rate. The next chapter investigates the feasibility of using alternative soft decision mechanisms in order to aid channel estimation / tracking. These techniques use confidence information of the symbol estimates fed back to the channel estimator in order to modify the estimator adaptation process. The basis of these techniques is to limit the change in the channel estimate when the equaliser is less confident

in its symbol decision and thereby attempt to reduce the build-up in the mean tap error that occurs due to tracking with erroneous symbols.

Chapter 8

Soft Decision Tracking

The use of soft decisions / confidence measures of symbol estimates obtained from equalisers for the enhancement of channel estimator tracking is examined in this chapter. The motivation for this technique and previous work on the subject is discussed and results of analysis and simulation of the effects on the LMS adaptation algorithm are presented. The advantages of the technique are seen to be dependent on the distribution of the equaliser's errors and so the statistics of these errors are investigated in order to assess the suitability of the soft decision tracking technique.

8.1 Basis of Soft Decision Tracking

MLSE and Bayesian equalisers require an estimate of the channel impulse response in order to produce estimates of the transmitted symbols. For time-varying channels, symbol estimates from the output of the equaliser may be used by the channel estimator to aid channel tracking, i.e. decision-directed tracking. In the ideal case, the equaliser estimates the transmitted symbols perfectly and the channel estimator's tracking performance is dependent on the receiver noise and the LMS adaptation stepsize, μ. In such circumstances, the channel estimation error may be decomposed into two parts [2, 47]: weight-vector noise and weight-vector lag noise. The former is the error associated with approximating the error surface gradient vector and the latter is the error due to a difference in the ensemble mean of the weight-vector, $\underline{\widehat{\mathbf{h}}}(k)$, and the true channel impulse response, $\underline{\mathbf{h}}(k)$. The lag noise is due to the true channel impulse response varying with time.

Errors in the equaliser's symbol estimates will add another component to the estimation error and will cause an increase in the average channel estimation error. The estimator's weights (the complex gains of the estimated CIR) will be adjusted suboptimally and this can cause the equaliser to make further symbol errors forming an error propagation mecha-

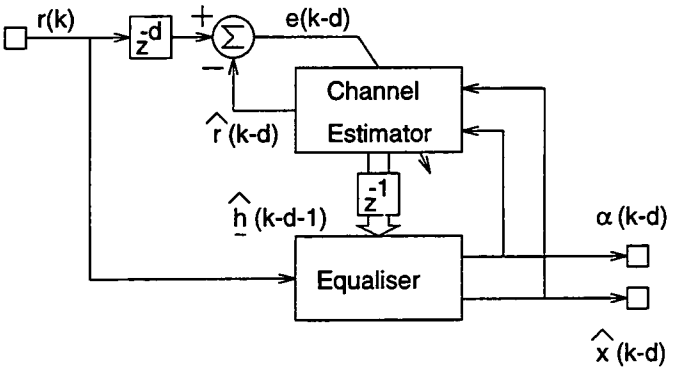

Figure 8.1: Schematic of equaliser and channel estimator with soft decisions / confidence values, α.

nism. The detrimental effects of this error mechanism on the MLSE's symbol error rate were demonstrated in Chapter 5. Alternatively, soft decisions can be fed into the channel estimator instead of the 'hard' symbol decisions or confidence measures can be used to modulate the adaptation algorithm's updating of the estimator's weights (Fig. 8.1). The hypothesis is that these new schemes will reduce the estimation error during instances of incorrect symbol estimates and reduce the chances of error propagation.

In Morgul *et al* [77] a soft decision technique was applied to a Kalman filter based channel estimator and the estimator's convergence after a step change in the channel impulse response was simulated. The method utilises an 'erasure declaring Viterbi detector' which generates a binary confidence value in the estimated symbol (indicating an 'erasure' or low-confidence decision) together with two parameters which indicate the reliability of the erasure declaration. As the process of declaring an erasure does not give completely accurate results, the parameters are the probabilities of false alarm and mistake, i.e. the probabilities of declaring an erasure when the decision is correct and of not declaring an erasure when the decision is erroneous. These probabilities are assumed to be known *a priori* and are incorporated into the Kalman filter adaptation algorithm. The results of the simulations suggested that the erasure declaring process was the most critical component of the technique.

The soft decisions or confidence measures may be generated with varying ease by equalisers which base their decisions on a statistical basis e.g. MLSE type equalisers or Bayesian equalisers. Indeed, soft decisions from equalisers are often used to aid subsequent channel decoding in the receiver [2, 4, 5]. A method of generating soft decisions from the MLSE is given in Hagenauer *et al* [78] which forms an estimate of the *a posteriori* probability for each symbol based on the transition metrics of the Viterbi algorithm.

The suitability of any soft decision mechanism is primarily dependent on whether the adaptation algorithm can make use of such information in order to improve the tracking

capabilities and thus reduce the channel estimation error. Given positive results in this area the problem of generating the soft decisions may be addressed.

8.2 Analysis of Soft Decisions Applied to the LMS Tracking Algorithm

The effects of either scaling symbol decisions by a confidence value prior to insertion into the channel estimator or using the confidence value to modulate the estimate updates are investigated both from a theoretical basis and by simulation. The channel estimator is assumed to be updated by the LMS algorithm in all cases.

The method of scaling the symbol estimates is to weight the decisions such that the higher the confidence, the closer the soft-decisions become to the hard decisions (the transmitted symbol alphabet) e.g. $x(k) \in \{\pm 1 \pm j\}$ for QPSK modulation. This has the effect that erroneous decisions have a lesser impact on the channel estimator's received signal estimate and hence on the error signal used to update the channel estimate (given that the confidence measure is correct). The BDFE's decision metrics are highly suited to this approach as the metrics are an estimate of the *a posteriori* probability for each symbol in the symbol alphabet. The soft decision may be formed in two slightly different ways:

1. Metric weighted: the soft decision is simply the hard decision weighted by the metric for that symbol. This may be expressed as:

$$x^{sd} = \frac{\lambda(x^d)x^d}{\sum_{i=0}^{M-1}\lambda(x_i)} \tag{8.1}$$

2. Mean symbol: the soft decision is the mean symbol value:

$$x^{sd} = \frac{1}{\sum_{i=0}^{M-1}\lambda(x_i)}\left(\sum_{i=0}^{M-1}\lambda(x_i)x_i\right) \tag{8.2}$$

In the equations above, x^{sd} is the soft decision, $\lambda(x)$ is the estimated probability for the symbol x, and x^d is the equaliser's hard decision. The summations of $\lambda(x)$ are to ensure a normalised probability. These mechanisms result in the soft decision lying in the range:

$$x^{sd} = (p + jq) \quad -1 \leq p, q \leq +1 \tag{8.3}$$

for the metric weighted method and

$$x^{sd} = (p + jp) \quad 0.25 \leq |p| \leq +1 \tag{8.4}$$

for the mean symbol method. Both these ranges are for QPSK symbols and for the latter the minimum decision metric that the most probable symbol can have is 0.25. Note that the metric weighted method results in the symbol decision being weighted by a real number and the mean symbol method by a complex number. Thus, in general, the soft decisions produced by scaling the hard decisions may be given by:

$$x^{sd} = \alpha x^{d}. \tag{8.5}$$

The method of modulation of the channel estimator's stepsize, μ, by confidence values is achieved by storing a vector of confidence values, $\underline{\alpha}(k)$, with each component being the respective value for the (hard) symbol estimates used in the channel estimator.

8.2.1 Derivation of Modified LMS Algorithm

The modified versions of the LMS algorithm are derived for both soft decision schemes: scaling of the hard decisions and modulation of the stepsize. The methods are shown to result in different algorithms and the effects of the differences are simulated in the Section 8.2.2.

Hard Decision Scaling

Under conditions of scaling of the hard decision by either a real number (metric weighted method) or a complex number (mean symbol method), the estimated received signal of the channel estimator is given by:

$$\widehat{r}(k) = \underline{\widehat{\mathbf{h}}}^{T}(k)\,[\underline{\alpha}(k).\underline{\widehat{\mathbf{x}}}(k)] \tag{8.6}$$

where the decision delay, d, is assumed to be zero without loss of generality and (.) indicates component-wise multiplication and *not* a scalar product, i.e. $[a_1, a_2]^T.[b_1, b_2]^T = [a_1 b_1, a_2 b_2]^T$. See Figure 8.1 for further definitions of the signals. Note that this definition of the estimated received signal results in different values the two methods of scaling the hard decisions.

The error signal, $e(k)$ is given by:

$$\begin{aligned} e(k) &= r(k) - \widehat{r}(k) && (8.7)\\ &= \underline{\mathbf{h}}^{T}(k)\underline{\mathbf{x}}(k) + n(k) - \widehat{r}(k) && (8.8) \end{aligned}$$

The error function, $J(k)$ is defined as:

$$\begin{aligned} J(k) &= e(k)e^{*}(k) && (8.9)\\ &= \left(r(k) - \underline{\widehat{\mathbf{h}}}^{T}(k)\,[\underline{\alpha}(k).\underline{\widehat{\mathbf{x}}}(k)]\right)\left(r^{*}(k) - [\underline{\alpha}(k).\underline{\widehat{\mathbf{x}}}(k)]^{H}\,\underline{\widehat{\mathbf{h}}}^{*}(k)\right) && (8.10) \end{aligned}$$

and its gradient is obtained by differentiating w.r.t. $\underline{\hat{\mathbf{h}}}(k)$:

$$\begin{aligned}
\nabla J(k) &= \frac{\partial J(k)}{\partial \underline{\hat{\mathbf{h}}}(k)} && (8.11)\\
&= -\left[\underline{\alpha}(k).\underline{\hat{\mathbf{x}}}(k)\right] r^*(k) + \left[\underline{\alpha}(k).\underline{\hat{\mathbf{x}}}(k)\right]\left[\underline{\alpha}(k).\underline{\hat{\mathbf{x}}}(k)\right]^H \underline{\hat{\mathbf{h}}}^*(k) && (8.12)\\
&= -\left[\underline{\alpha}(k).\underline{\hat{\mathbf{x}}}(k)\right] e^*(k) && (8.13)
\end{aligned}$$

The weight update is given by:

$$\begin{aligned}
\underline{\hat{\mathbf{h}}}(k+1) &= \underline{\hat{\mathbf{h}}}(k) - \mu\left(\nabla J(k)\right)^* && (8.14)\\
&= \underline{\hat{\mathbf{h}}}(k) + \mu e(k)\left[\underline{\alpha}(k).\underline{\hat{\mathbf{x}}}(k)\right]^* && (8.15)
\end{aligned}$$

This update equation is not the same for the two different methods of scaling the hard decisions as both $e(k)$ and $\underline{\alpha}(k)$ vary between the two methods.

Confidence Value Modulation

Modulating the stepsize, μ, by the confidence values does not alter the formation of the channel estimator's received signal estimate compared to the standard hard decision channel estimator. Thus the estimated received signal is given by:

$$\hat{r}(k) = \underline{\hat{\mathbf{h}}}^T(k)\underline{\hat{\mathbf{x}}}(k) \tag{8.16}$$

The error signal is given by Eqns. 8.7 and 8.8 and the error function, $J(k)$ is defined as:

$$\begin{aligned}
J(k) &= e(k)e^*(k)\\
&= \left(r(k) - \underline{\hat{\mathbf{h}}}^T(k)\underline{\hat{\mathbf{x}}}(k)\right)\left(r^*(k) - \underline{\hat{\mathbf{x}}}^H(k)\underline{\hat{\mathbf{h}}}^*(k)\right) && (8.17)
\end{aligned}$$

The error gradient is obtained as before by differentiating w.r.t. $\underline{\hat{\mathbf{h}}}(k)$:

$$\begin{aligned}
\nabla J(k) &= \frac{\partial J(k)}{\partial \underline{\hat{\mathbf{h}}}(k)}\\
&= -\underline{\hat{\mathbf{x}}}(k)r^*(k) + \underline{\hat{\mathbf{x}}}(k)\underline{\hat{\mathbf{x}}}^H(k)\underline{\hat{\mathbf{h}}}^*(k) && (8.18)\\
&= -\underline{\hat{\mathbf{x}}}(k)e^*(k) && (8.19)
\end{aligned}$$

The weight update is given by:

$$\begin{aligned}
\underline{\hat{\mathbf{h}}}(k+1) &= \underline{\hat{\mathbf{h}}}(k) - \left[\mu\underline{\alpha}(k)\right].\left[\nabla J(k)\right]^* && (8.20)\\
&= \underline{\hat{\mathbf{h}}}(k) + \left[\mu\underline{\alpha}(k)e(k)\right].\underline{\hat{\mathbf{x}}}^*(k) && (8.21)
\end{aligned}$$

Again the value of $e(k)$ is different from the two possible corresponding terms in Eqn. 8.15.

8.2.2 Simulation of Soft Decision Tracking

The quantative effects on the channel estimator's mean tap error (MTE) are simulated for the methods of metric weighted decisions (scaling the hard decision by a real number) and the confidence value modulation technique. The mean symbol soft decision method has not been simulated due to the large number of possible values of α. The equaliser has not been simulated in order to isolate the results from any problems in the generation of the soft decisions. The effects of tracking with accurate symbol estimates but with varying confidence values is studied together with the effects of both independent and correlated error events.

The system simulated used QPSK modulation at 100kBaud without the use of transmit or receive filters. After an initial 500 symbol training sequence a frame structure of 100:100 was employed. The channel was a three T_s-spaced equipower path channel with the paths fading independently with a classical Doppler spectrum and Doppler spread of $f_d = \pm 400$Hz. The channel estimator had three T_s-spaced taps and was udpated by the modified LMS algorithms given in Section 8.2.1 with $\mu = 0.1$. The simulations were run at an SNR = 15dB.

Figures 8.2 and 8.3 show the build-up of MTE across the data burst averaged over 1000 bursts for the two methods of soft decisions. The data burst index label in the figures indicates the symbol position within the data burst. The channel estimator's symbol estimates were 100% correct but with varying confidence values, α. The lowest curves show that if the confidence in the symbol estimates is 100% ($\alpha = 1.0$) then the MTE remains at a low level (determined by the weight-vector noise and weight-vector lag noise). As the confidence drops the MTE rises and then levels out at a higher MTE. This additional MTE component is due to the suboptimum tracking of either the scaled symbol estimates or the reduced adaptation stepsize. As can be seen, the method of modulating the stepsize is much less prone to the effects of a 'false alarm', i.e. the equaliser makes the correct decision but with a low confidence.

Figures 8.4 and 8.5 demonstrate the effect on the MTE of the channel estimator driven with independent, random symbol errors with a probability 0.1. The confidence value of the symbol estimates during the error events is varied and shown in the figures. Otherwise the correct symbol estimates are fed to the channel estimator with a confidence value of 1.0. All curves have been averaged over 1000 data bursts. Both methods of soft decision show a reduced value of MTE as the confidence in the erroneous symbol estimates drops, with the metric weighted version having the greatest effect. However, the absolute reduction in MTE is small, especially for the confidence value modulation method.

Further results show that similar decreases in the MTE occur over a range of different independent, random error event probabilities. Under these conditions of error events, the decrease in MTE is expected to result in a decrease in symbol error rate although the quantative decrease has not been simulated. Such simulations require the development of an equaliser which can generate the suitable confidence values.

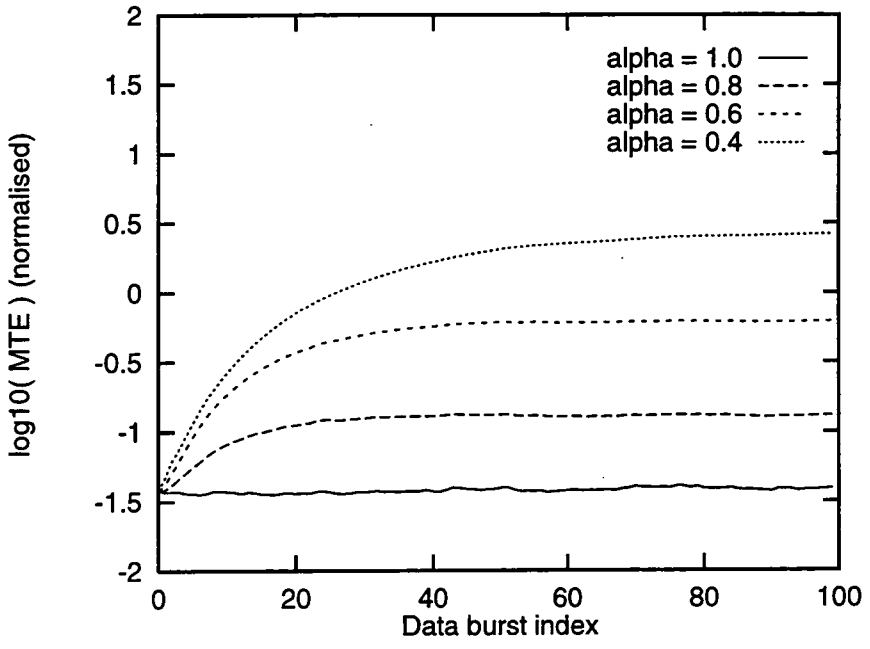

Figure 8.2: Metric weighted soft decision tracking with accurate symbol estimates of varying confidence.

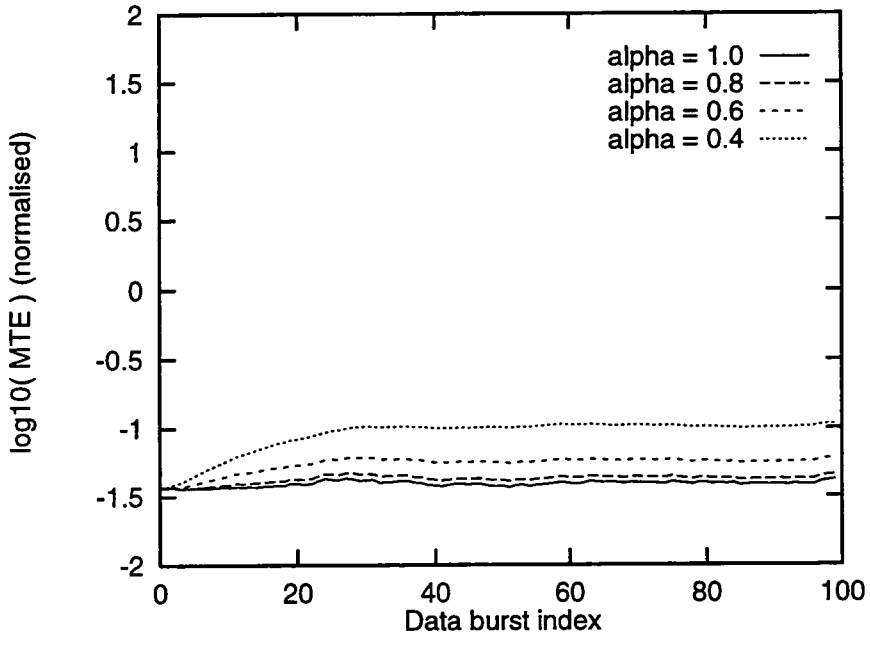

Figure 8.3: Confidence value modulation tracking with accurate symbol estimates of varying confidence.

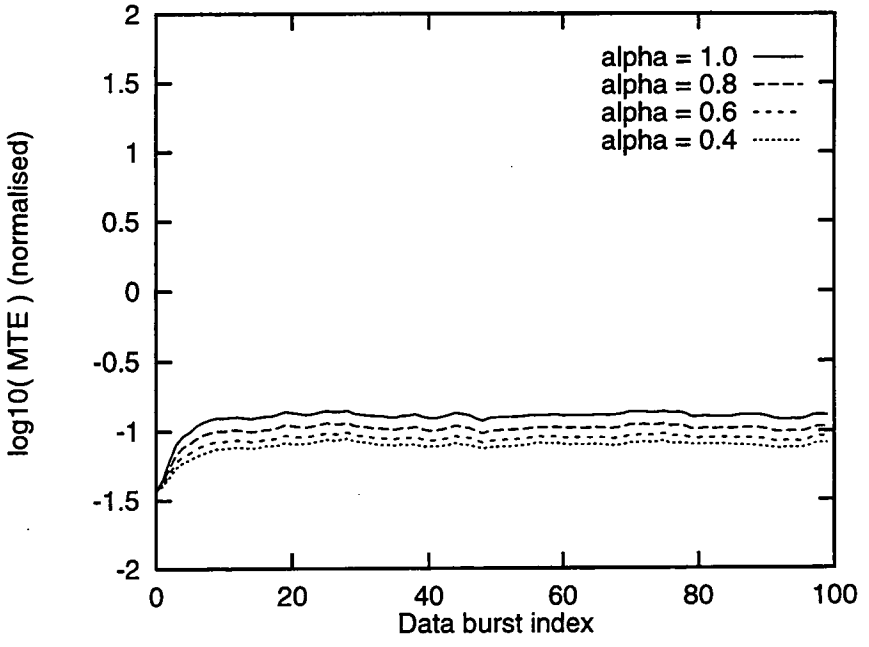

Figure 8.4: Metric weighted soft decision tracking with independent, random errors in the symbol estimates. α corresponds to the confidence value during error events.

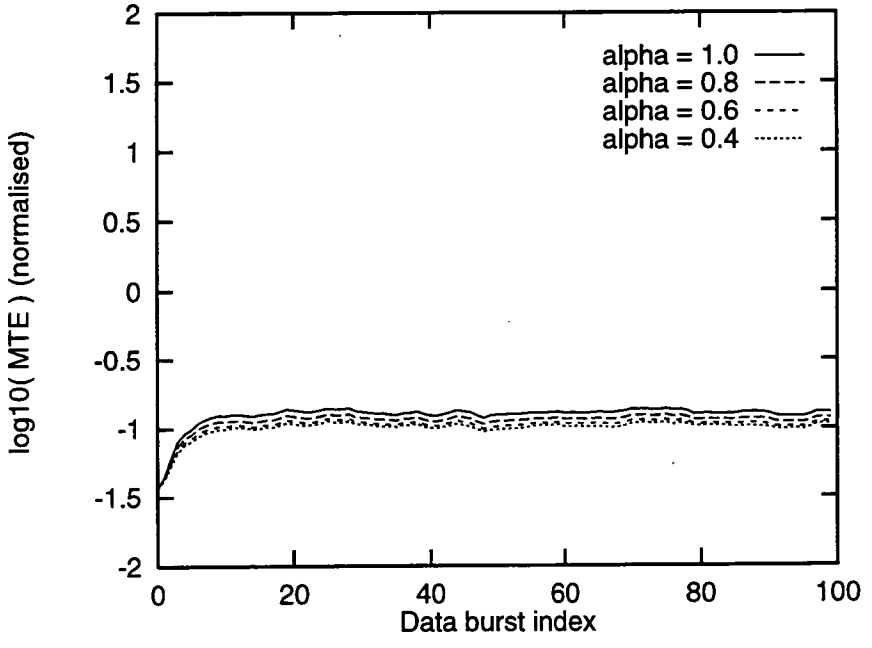

Figure 8.5: Confidence value modulation tracking with independent, random errors in the symbol estimates. α corresponds to the confidence value during error events.

The effects of the equaliser not being 100% confident when the correct symbol estimate is made are shown for the two soft decision methods in Figures 8.6 and 8.7 respectively. The probability of an error event is again 0.1. The metric weighted soft decision technique is seen to again suffer when tracking with lower confidence but correct symbol estimates and this negates the benefits shown in Figure 8.4. The confidence value modulation technique does not suffer greatly from the lower confidence correct decisions but the gains due to the low confidence erroneous decisions are small leading to only a small overall improvement.

Finally the performance of the soft decision techniques are simulated when a burst of equaliser errors occur. Figures 8.8 and 8.9 show the results for the two soft decision methods for a range of error burst lengths. The probability of the initial (independent) error events are 0.02 and each of these error events are followed by further errors, creating an error burst of a desired length. During error events the equaliser has a confidence value of $\alpha = 0.8$ whilst during periods of correct decisions the equaliser is 100% confident. The curves demonstrate that the MTE increases rapidly with burst length for both techniques and that the absolute difference between the two is negligible. The results for similar simulations but with the equaliser's confidence value set at 0.4 during periods of incorrect decisions show a decrease in MTE of about 0.25dB for both soft decision methods.

All the curves above that show results for tracking with error events of some sort have been averaged over 1000 data bursts with the error events occuring randomly throughout the data bursts. Thus the curves tend to show the average level of MTE due to such error characteristics. Figures 8.10 and 8.11 show the average build-up of MTE during a burst of 50 errors occuring midway through the data burst. The equaliser is 100% confident during correct decisions and the confidence values during the error burst are indicated in the figures. These curves show a characteristic jump in the MTE when the burst occurs, the increase being greatest for the confidence value modulation soft decison method. This is due to a larger error signal being generated at the output of the channel estimator when the unscaled erroneous hard decisions are fed into the estimator. The increase in MTE becomes lower as the confidence in the erroneous decisions drops. These curves show that the metric weighted method increases the time before the MTE rises above a given threshold and therefore this technique is less prone to bursts of errors provided that the equaliser can recognise when it is making errors.

The results of the simulations in this section show that, although the soft decision methods can reduce the build-up in MTE during erroneous symbol decisions, the techniques are limited by both the equaliser's ability to detect when it is likely to be making an error and when it is not and also of the statistics of the equaliser's errors. Bursts of errors cause a rapid increase in the MTE which has the detrimental effect of causing further symbol errors. In the following section the MLSE's error statistics are characterised in order to identify typical error burst lengths and frequencies.

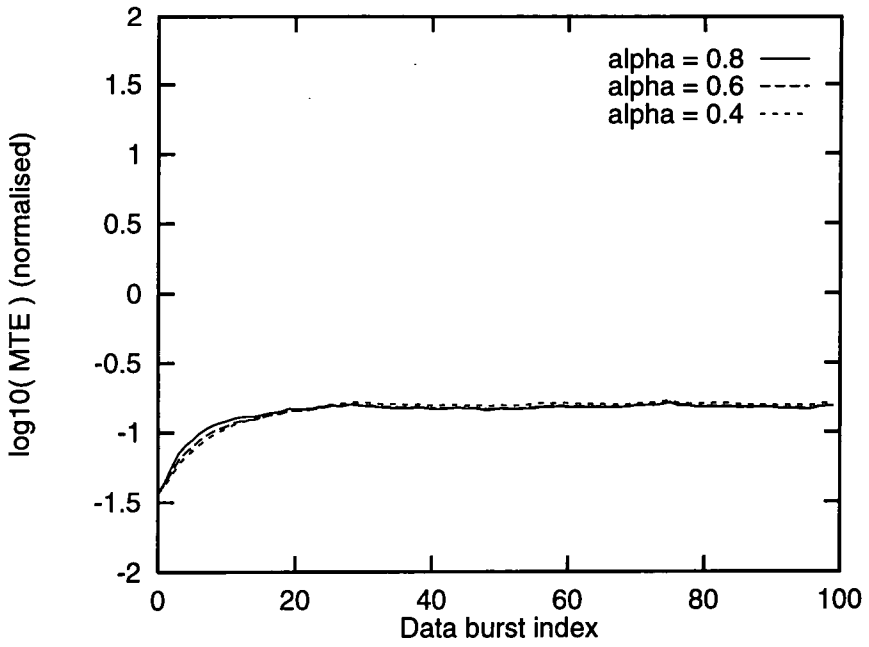

Figure 8.6: Metric weighted soft decision tracking with independent, random errors and reduced confidence in the correct estimates. α corresponds to the confidence value during error events and the confidence of correct symbol estimates is 80%.

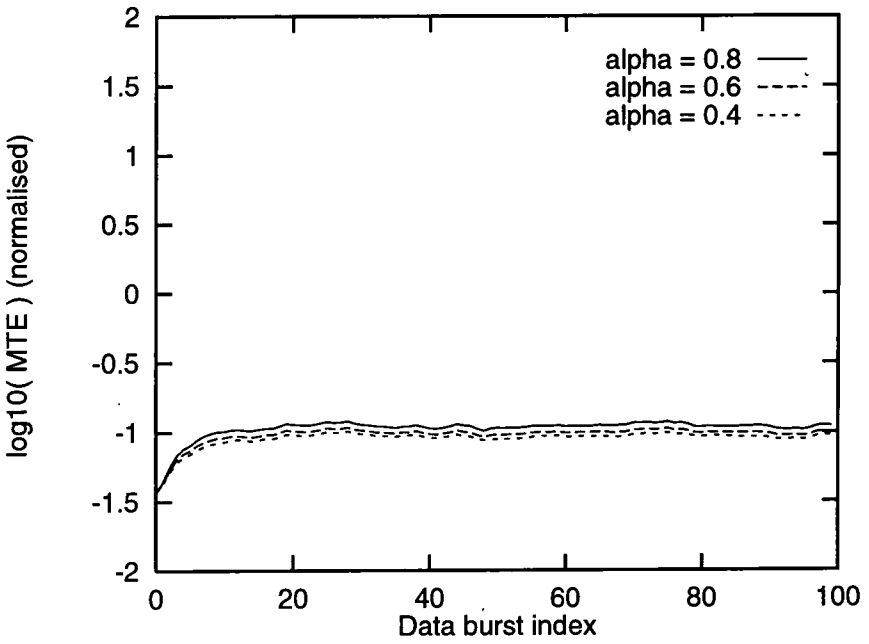

Figure 8.7: Confidence value modulation tracking with independent, random errors and reduced confidence in the correct estimates. α corresponds to the confidence value during error events and the confidence of correct symbol estimates is 80%.

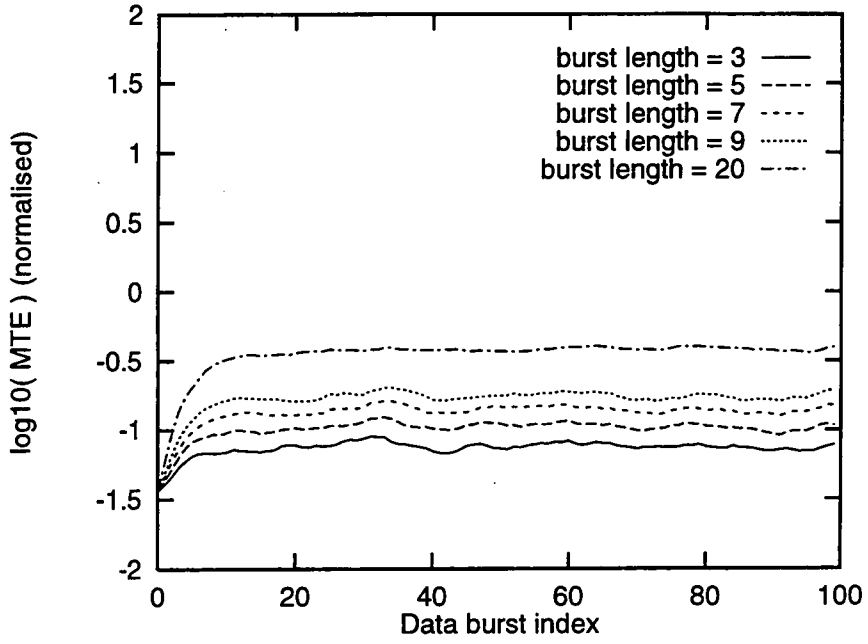

Figure 8.8: Metric weighted soft decision tracking with bursts of errors of varying duration.

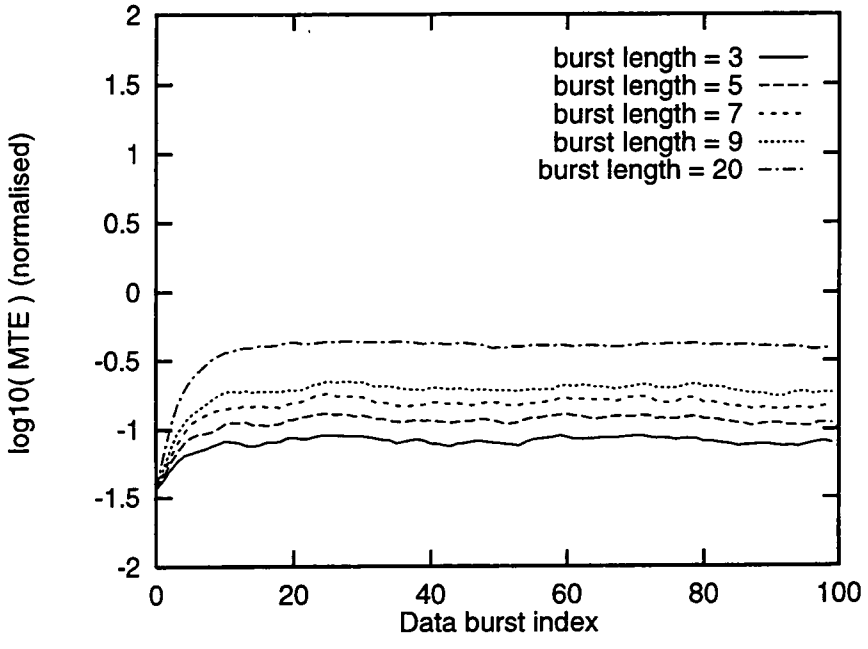

Figure 8.9: Confidence value modulation tracking with bursts of errors of varying duration.

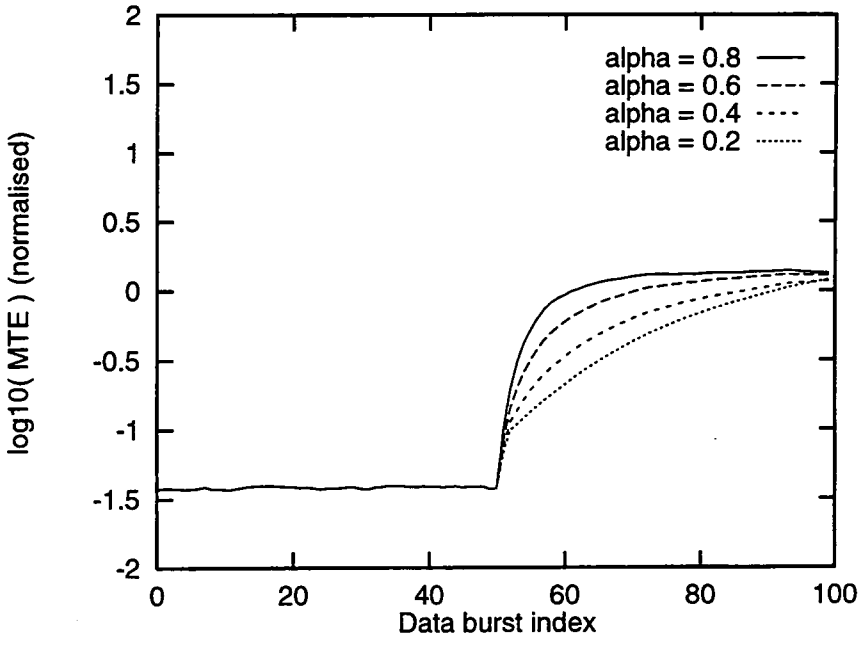

Figure 8.10: Metric weighted soft decision tracking with a burst of 50 errors occuring midway through the data burst. α is the confidence value during the error burst.

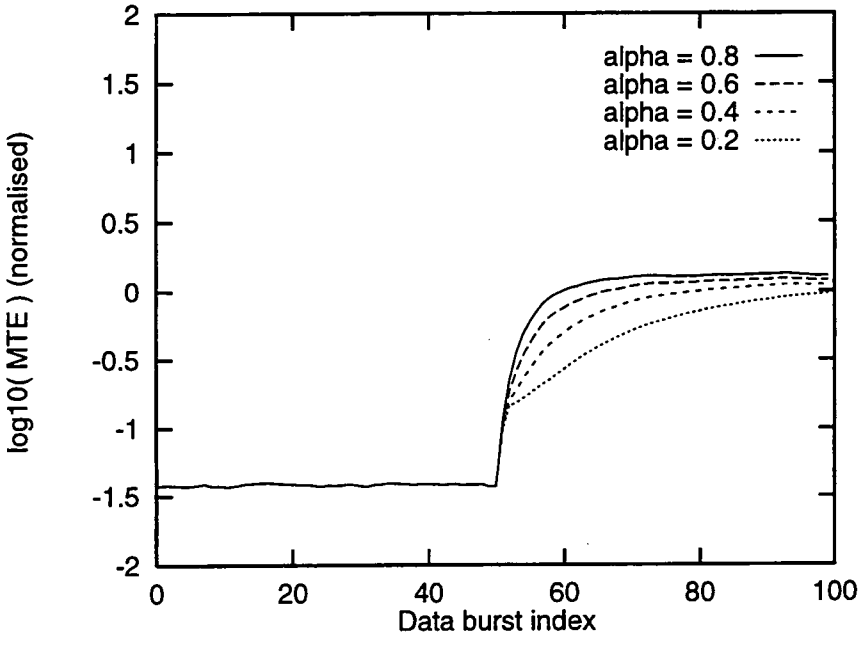

Figure 8.11: Confidence value modulation tracking with a burst of 50 errors occuring midway through the data burst. α is the confidence value during the error burst.

8.3 Error Statistics of MLSE Equaliser Errors

Some properties of the MLSE's symbol estimation errors are measured in order to determine the frequency and duration of error bursts. The channel on which the errors are measured are the same as in Section 8.2.2, i.e. a three T_s-spaced equipower path channel with a Doppler spread of $f_d = \pm 400$Hz. The channel estimator used three T_s-spaced taps and updated the channel estimate via the LMS algorithm with a stepsize $\mu = 0.1$ using the conventional hard symbol decisions during decision-directed tracking. The QPSK modulated data was transmitted with a frame structure of 100:20, the received SNR = 15dB and the MLSE had a decision delay of three symbol periods.

The first simulation logged the numbers of errors in each data burst over the simulation run length of 1587 data bursts. Figure 8.12 shows the distribution of the number of errors per data burst for the MLSE and channel estimator operating in a decision-directed mode and using true delayed data tracking. The two curves show that during decision-directed tracking the number of data bursts containing no errors is large and a much smaller number of bursts contain large numbers of errors. This is in contrast to the true delayed data tracking distribution which shows that a much larger number of bursts contain only a few errors. This observation is highlighted in Figure 8.13 which shows the percentage of the total number of errors occuring with less than or equal to the number of errors per burst shown. Thus it is seen that with decision-directed tracking 5% of the total number of errors are caused by bursts with ≤ 10 errors/burst and 11% by bursts with ≤ 20 errors/burst. In the true delayed data tracking case 64% of the total number of errors are caused by bursts with ≤ 10 errors/burst and 96% by bursts with ≤ 20 errors/burst. Decision-directed tracking, which creates an error propagating mechanism, results in the majority of the errors caused by relatively few bursts with large numbers of errors.

This result is checked for validity by calculating the correlation between the logged number of errors per burst. If the correlation shows that blocks with large numbers of errors occur consecutively then this would indicate that the channel estimator has lost track of the channel impulse response and that the 20 symbol training period is insufficient to reacquire the CIR. This condition could occur if all 3 paths went through a deep fade simultaneously leading to a deep fade in the received signal power. Figure 8.14 shows the measurements of the unbiased correlations for the decision-directed and true delayed data tracking cases. From the shape of the curves it is seen that the correlation for a small number of shifts is not significantly larger than those for larger shifts, indicating that the errors are evenly distributed throughout the duration of the simulation rather than being clumped together over a short period.

The characteristics of the spread of errors within the data burst are measured by calculating the unbiased correlation of the error events within a data burst and averaged over the duration of the simulation. These results are shown in Figure 8.15 for the decision-directed

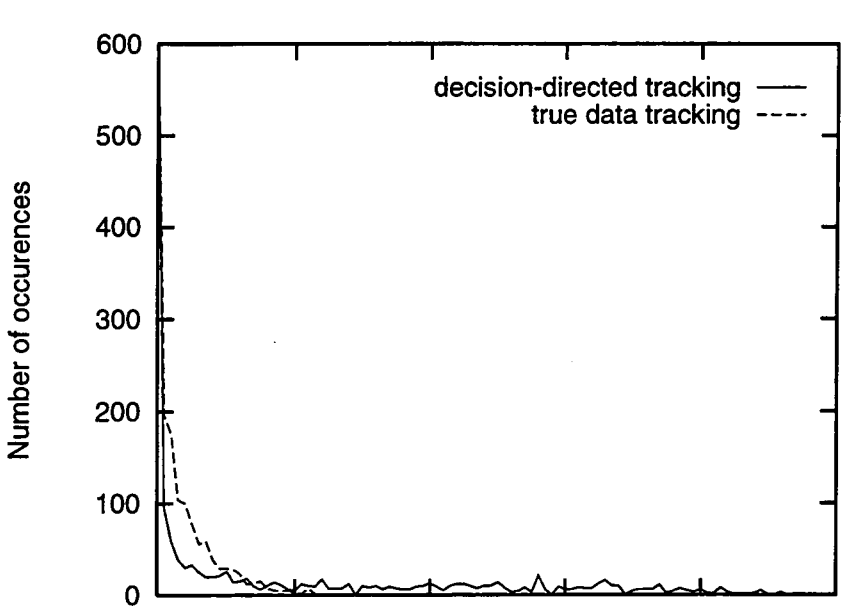

Figure 8.12: Distribution of the number of errors per data burst.

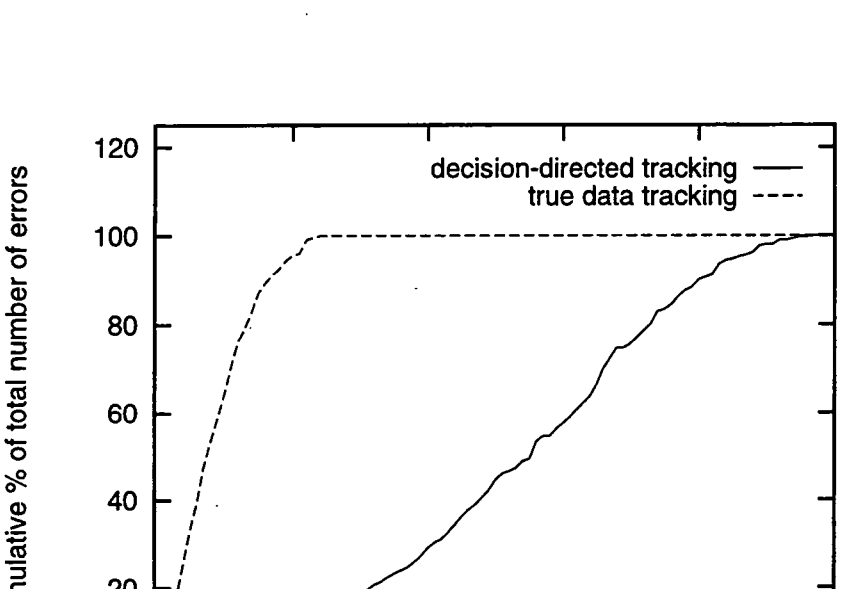

Figure 8.13: Cumulative percentage of the total number of errors as a function of the number of errors per data burst.

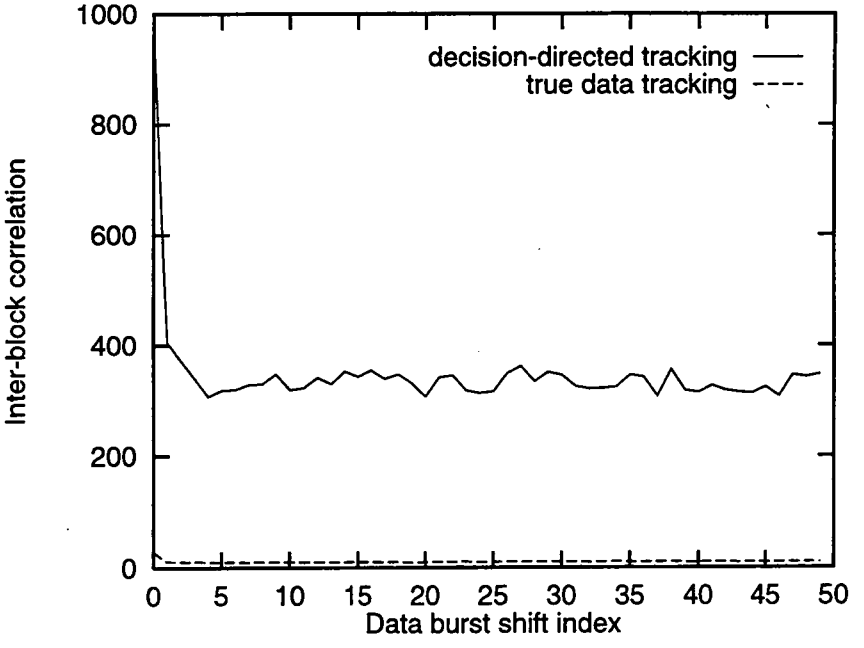

Figure 8.14: Correlation of the number of errors per block over the duration of the simulation.

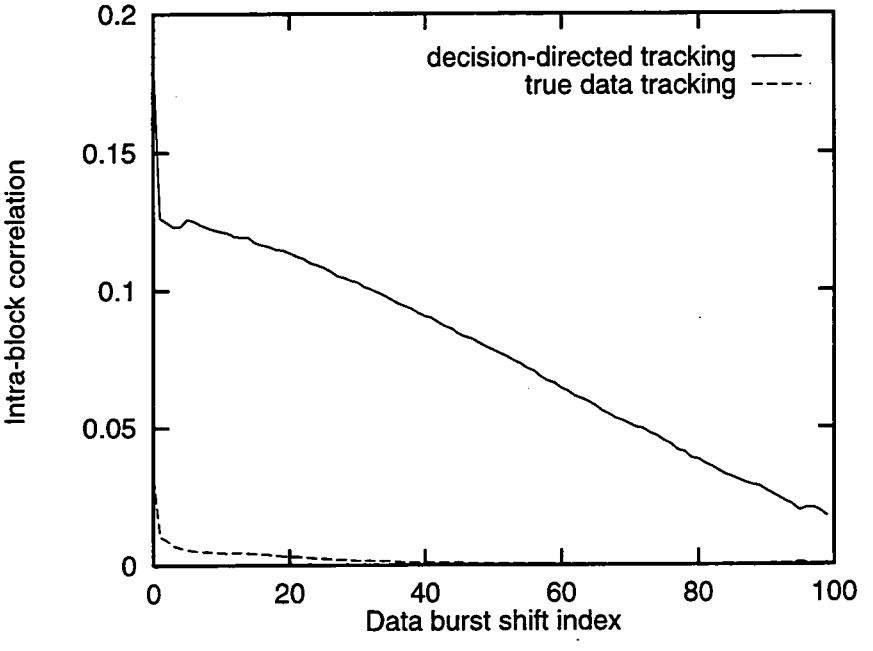

Figure 8.15: Correlation of the error events within the data burst averaged over the duration of the simulation.

and true delayed data tracking configurations and demonstrate that the error events are much more correlated in the case of decision-directed tracking. This observation indicates that the errors have a burst structure within the data burst. The correlation values obtained include the effects of error propagation due to (hard) decision-directed tracking. If soft decision techniques do aid tracking then the error feedback problem will be reduced leading to a lower correlation of the error events. However, the results of Section 8.2 show that the soft decision techniques do not reduce the MTE down to the level obtained with true delayed data tracking and so the error events are still likely to be significantly correlated compared to the true delayed data tracking errors. Thus the soft decision methods discussed in Section 8.2.2 are unlikely to have a major impact on the SER. The true delayed tracking curve shows that these errors are far less correlated. Note that the absolute value of correlation will be smaller as the number of errors is less than in the decision-directed case but the relative levels for small shifts are much lower.

8.4 Soft Decision Generation

The methods of generating and the distribution of the soft decision metrics is an important part of the soft decision technique as demonstrated in Section 8.2.2. If the equaliser cannot detect whether its decisions are correct or erroneous with sufficient accuracy then the soft decision technique is seen to be significantly impaired.

The generation of the soft decisions for the Bayesian equaliser is straightforward as the decision metrics are estimates of the *a posteriori* symbol probability. From Bayes' rule,

$$P(x_i|r) = \frac{P(r|x_i)P(x_i)}{\sum_{j=0}^{M-1} P(r|x_j)P(x_j)} \tag{8.22}$$

for mutually exclusive transmitted symbols x_j. For equally probable x_j then

$$P(x_i|r) = \left[\frac{P(r|x_i)}{\sum_{j=0}^{M-1} P(r|x_j)}\right] \frac{P(x_i)}{P(x_i)} \tag{8.23}$$

$$= \frac{P(r|x_i)}{\sum_{j=0}^{M-1} P(r|x_j)} \tag{8.24}$$

$$= \frac{\lambda(x_i)}{\sum_{j=0}^{M-1} \lambda(x_j)} \tag{8.25}$$

This final equation may be compared to that of the metric weighted soft decision, Eqn. 8.1 to see that the Bayesian equaliser's decision metrics can be used easily to generate the soft decisions. The extension to the mean symbol soft decision or to the confidence value method is trivial.

However, simulations recording the distribution of the confidence values, α, show that in

practice the equaliser produces a very narrow spread of values. This occurs both for symbol estimates that are correct and incorrect and in effect the equaliser is always very confident in its decisions. The result indicates that the Bayesian equaliser does not produce suitable confidence values for the soft decision tracking methods discussed when operating on fading channels. Similar problems were observed in Morgul *et al* [77] in the erasure declaring part of the modified Viterbi algorithm.

8.5 Summary

The possibility of using confidence measures in the equaliser's symbol estimates to aid tracking of a time-varying channel has been investigated. The degradation in the quality of the channel estimate due to erroneous data being fed to the estimator during decision-directed tracking was observed in Chapter 5. The confidence information has been used to modify the LMS adaptation algorithm by a number of methods so that the change in the estimate is reduced during periods of low confidence. These methods either scale the symbol estimates by a real or complex number or modulate the LMS algorithm's stepsize parameter, μ.

Simulations of the resulting algorithms have shown that the technique can reduce the mean tap error of the channel estimator if the equaliser can generate good quality confidence / reliability measures. Thus the equaliser must be able to recognise when it is making an error and when it is not. Clearly, the actual symbol decision that the equaliser produces is the best estimate (based on some performance measure) but, by comparing the decision metrics for the other possible symbols, some measure of the confidence in the decision can be formed. The Bayesian equaliser's decision metrics are suited to the formation of a confidence value based on the *a posteriori* symbol probability but, since the equaliser operates on estimates of parameters such as the noise statistics and channel impulse response, the actual decision metrics are only *estimates* of the symbol probability. In practice these estimates are seen to vary significantly from the actual values as observed in the high symbol error rate compared to the error rate obtained using the actual channel impulse response and noise statistics. This difference produces inaccurate confidence information.

A second area which limits the effectiveness of the soft decision technique is the burst nature of the equaliser errors. Even if the equaliser produces accurate confidence values, the result of updating the channel estimate by small amounts leads to a rise in the mean tap error as the channel is time-varying. This situation is aggravated by rapidly changing channels and leads to further equaliser symbol errors.

The presence of burst error characteristics is the major limitation to the application of the soft decision technique. In situations where the equaliser forms burst of errors the soft decision technique will not have a significant positive effect on the symbol error rate. However,

this does not totally exclude the possibility that, in some situations, the technique prevents or delays the onset of error bursts. The potential for deriving improved confidence information from equalisers exists and, given success in this area and the presence of less dependent symbol errors, the soft decision technique will be able to reduce the channel estimation error. The result of this decrease will be to reduce the occurence of equalisation / detection errors.

Chapter 9

Conclusions and Further Work

9.1 Conclusions

This thesis has investigated the development and design of adaptive equalisers for use in radio links with rapidly time-varying and frequency selective channels. Rapid channel variations become more common as either the relative speed between mobile transmitters and receivers or the speed of reflectors and/or diffractors (which create paths within the channel) increases. Additionally, the use of higher carrier frequencies causes higher Doppler shifts and spreads for a given mobile or channel velocity, creating more rapid time-variations. The frequency selectivity of the channel becomes increasingly significant with higher bandwith signals (resulting from higher symbol rate transmissions) due to a larger amount of intersymbol interference. The use of such radio channels is spurred by the increasing demand for radio communication with higher available bandwidths and for use in harsher radio environments.

The functionality of the equaliser has been discussed in the context of additional techniques (e.g. channel coding and interleaving) designed to improve the overall robustness of the radio link to nonideal channel effects. Conventional equalisation techniques are described together with the more recent development of neural network techniques. These latter techniques have been the subject of significant research effort due to their pattern recognition capabilities, which is essentially the function of equalisation and detection.

A critical analysis of the use of a recurrent neural network to provide improved equalisation has been carried out. This work has developed a theoretical analysis of single node structures and shows the limitations in the types of nonminimum phase channels that may be equalised with the use of the hyperbolic tangent nonlinear node element. The structured design of multinode networks has been carried out rather than relying on the adaptation algorithm to form a suitable network. These designs have been based on the cancellation of both pre- and postcursors of the channel impulse response in a manner similar to the conventional

decision feedback equaliser. Such similarities were observed in the structures obtained by the application of the original network's adaptation algorithm, the real time recurrent learning algorithm. Simulations of the symbol error rate performance have shown that the networks offer little improvement over the decision feedback equaliser and are highly sensitive to errors in symbol estimates used to cancel precursors of the channel impulse response. These results have been obtained from time-invariant channels and, given the known improvement of maximum likelihood techniques over decision feedback methods for use with time-varying channels, the application of the recurrent neural networks has not been investigated further.

Simulations have shown the limitations of the maximum likelihood based equaliser together with a channel estimator adapted by the LMS algorithm for time-varying channels. The results have shown that the equaliser's performance is constrained by the quality of the channel estimate used to form the equaliser's decision metrics. The channel estimator's ability to track the time-varying channel accurately is shown to be dependent on the quality of the symbol estimates used in the estimation of the received signal and the delay with which the symbol estimates have been generated. The identification of these limiting mechanisms has prompted investigation and development of alternative techniques in an effort to overcome the performance restrictions.

The first technique studied was the Bayesian equaliser which may be implemented as a radial basis function neural network. This equaliser forms its symbol estimates on a different statistical basis than the maximum likelihood technique. Both theoretical studies and simulations have been carried out to compare the robustness of the two statistical techniques in the presence of channel estimation errors. The generation of an analytic solution to the probability of symbol error requires a knowledge of the statistics of the channel estimation error and for practical channels the derivation of the probability of error becomes complex, at least, leading to very limited results. The results of simulations show that the Bayesian equaliser is no more robust to errors in the channel estimate than the maximum likelihood equaliser.

The second method investigated was the use of multiple channel estimators applied to the maximum likelihood method. Rather than using a single channel estimator in the process of symbol estimation, a number of different estimators are employed and the algorithm allows previous symbol estimates used in the channel estimation procedure to be updated. Extensions to the existing algorithm have been developed which allow a trade-off between the performance and the algorithm complexity by reducing the number of channel estimators employed by the algorithm. The relative number of computations has been compared for different equalisers and configurations and indicates that the complexity increase for the multiple channel estimator technique, in order to obtain a performance gain, is in the order of 1.25 to 3 times more complex than the conventional maximum likelihood technique for typical channels and transmitted alphabets. The significance of this increase in computation

is application dependent but is of such an order that the R-MCE technique is of practical interest. The performance of the multiple channel estimator techniques show that as the number of channel estimators is reduced, the symbol error rate increases and eventually becomes worse than the conventional maximum likelihood algorithm. This threshold occurs when the number of estimators falls below 4 or 16, depending on the rapidity of the channel variations and is independent of the length of the channel impulse response. Simulations have been carried out on a number of different channels and the results demonstrate that the technique performs better when the channel variations are rapid ($f_d T_s > 0.002$) and when the length of the channel impulse response spans more than two symbol periods.

The final method investigated was the use of soft decisions or confidence measures of the equaliser's symbol estimates to aid decision-directed tracking of the channel estimator. The effects of two methods of using confidence measures to modify the channel estimator's LMS adaptation algorithm have been simulated and indicate that, provided accurate confidence measures can be produced, the mean tap error of the channel estimate can be reduced which should lead to a decrease in the symbol error. However, the use of soft decisions is limited additionally by the occurence of bursts of errors. During such events the tracking of the channel becomes difficult as the estimator has little or no reliable information with which to calculate the estimate. Coupled with a rapidly changing channel, the mean tap error increases quickly leading to further equaliser / detector errors.

9.2 Further Work

The application of neural networks to the equalisation of rapidly time-varying radio channels is considered limited unless the network structure is particularly suited to the equalisation problem and can be adapted quickly and accurately. Whilst some neural networks (e.g. radial basis functions) have the appropriate pattern recognition qualities required to separate the components of the received signal arising from intersymbol interference and additive noise, the adaptation algorithms used to train the networks must be capable of adapting rapidly to changes in the channel parameters. The patterns caused by intersymbol interference are relatively simple but the rate of change of the underlying mechanisms causing the interference is large and allows conventional equalisation techniques, which can be adapted more easily, to be used with increased performance over some neural network techniques (e.g. a general recurrent neural network). The development of faster adaptation algorithms for neural networks is a priority before such structures can be realistically used in the equalisation of the radio channels discussed.

The multiple channel estimator algorithms have shown an increased performance over the conventional maximum likelihood technique by means of providing symbol estimates for the channel estimators with zero decision delay and by allowing previous symbol estimates to be

updated. The zero decision delay property is more suited to rapidly changing channels and as the variations become slower the performance increase is reduced. Studies of the effects of introducing a decision delay into the sequence of symbol estimates fed to the channel estimators would provide an insight into the relative advantages of being able to update the symbol estimates versus the degradation due to estimating an old channel impulse response, especially for channels which do not vary quite as rapidly. The application of this additional method depends upon a knowledge of the rate of variations of the channel which will be known, in general, to some extent.

Further investigations into the soft decision techniques are required to ascertain the effects on alternative tracking algorithms such as the RLS algorithm. The soft decision methods discussed are not exhaustive of the manner in which confidence information could be used to modify adaptation algorithms. For instance, the use of a threshold on the confidence measure could be used to suspend adaptation of the channel estimator for a given period. The generation of reliable confidence information by the equaliser is imperative. Further research is required to develop confidence metrics which may be different from the decision metrics used to produce the symbol estimates. Such confidence metrics require less polarised values than those obtained, for instance, by the decision metrics of the Bayesian equaliser. However, the major barrier to the soft decision techniques discussed is the presence of bursts of errors. Under such conditions the decision-directed form of tracking is of limited use even when a burst of errors is recognised by the equaliser. There has been a growing interest in blind adaptation algorithms [31, 79, 80, 81, 82] and such methods may provide better channel tracking capabilities during bursts of errors. The soft decision techniques investigated in this thesis are not solely applicable to the conventional maximum likelihood plus channel estimator method. Any equalisation algorithm requiring a channel estimate could benefit from advances in improved channel estimation via soft decision methods. Such algorithms include versions of the decision feedback equaliser which use a channel estimate to derive the equaliser taps rather than direct adaptation of the taps, the multiple channel estimator techniques and the Bayesian equaliser.

In summary the problems of equalising radio channels in which the mechanisms causing intersymbol interference are time-variant are still present and are likely to become even more significant as the demand for higher data rates and mobile communications and computing increases. The solution to combat the higher bit error rates is likely to include improved equalisation techniques in conjunction with advanced error correcting codes.

Appendix A

Air-Radio Interfaces of Current Digital Mobile Radio Systems

This appendix details a number of functions of the air-radio interface aspects of some land based digital cellular mobile communication systems either in operation or the process of development. These digital cellular systems are called second generation networks, the first generation being analogue, and should be distinguished from satellite based systems such as INMARSAT-M, Odyssey, Globalstar and Iridium [83]. However, satellite and land mobile services are planned to be integrated in the third generation systems (Universal Mobile Telecommunication System, UMTS, sponsored by ETSI in Europe, and the ITU's International Mobile Telecommunications, IMT-2000, global standardisation).

The sections cover both 'cellular' systems and 'cordless' systems although there is a convergence in the functionality of the two classes. Cellular systems are designed for use in more widespread and diverse environments than cordless systems with the use of latter being normally constrained to within small 'picocells' such as homes, offices and small areas of streets. These cordless systems also cover 'telepoint' services. In addition the air-radio interface of the European standard for digital private mobile radio is described. For further details, see [4, 5, 49, 53, 84, 85, 86, 87, 88, 89, 90, 91, 92].

A.1 GSM / DCS1800

The GSM (Global System for Mobile communications) and DCS1800 (Digital Cellular System) are cellular based systems operating in many countries in Europe, Africa and Asia and are very similar to each other. The DCS1800 is a PCN (Personal Communication Network) system operating at a higher frequency. In the UK Cellnet and Vodaphone operate GSM systems and the Hutchinson 'Orange' and Mercury 'One-To-One' systems operate using

DCS1800. In the USA, PCN is refered to as PCS (Personal Communication Services) and a variant on GSM operating at 1.9GHz has been adopted as the standard for the licensed band.

The system operates with multiple channels, each using a TDMA format with 8 timeslots of 0.5769ms duration and 4.615ms separation. Base and mobile stations transmit / receive on separate 200kHz channels (i.e. frequency division duplexing, FDD) with a 45MHz separation between the duplex carriers for GSM (95MHz for DCS1800). In GSM, the carrier frequencies are in 25MHz bands from 890 - 915MHz (mobile to base / reverse link) and 935 - 960MHz (base to mobile / forward link). In DCS1800, carrier frequencies are in 75MHz bands from 1710 - 1785MHz on the reverse link and 1805 - 1880MHz on the forward link. The system provides a total of 124 duplex RF channels for GSM (374 for DCS1800). Speech codecs are RPE-LPC with a full-rate of 13kbit/s with a half-rate codec to be specified for future systems. Data transmission is supported upto 9.6kbits/s. Channel coding is via a rate $\frac{1}{2}$ convolutional coder with interleaving over 8 bursts plus error detection. GMSK modulation with a bandwidth-time product of $BT = 0.3$ is used with a bit rate of 270.833kbits/s (symbol period = $3.692\mu s$).

The system is capable of operating with a channel multipath delay spread of upto $20\mu s$ and thus requires an equaliser for such channels. The 200kHz wide channels results in multipath delay spread resolution which introduces ISI and diversity in some propagation environments.

A.2 IS-54

The IS-54 (or D-AMPS) cellular systems operate mainly in the USA. Multiple channels with a TDMA format are used with 6 timeslots, each of 40ms duration, but in current systems two timeslots are allocated to each user allowing only three users to use the same channel simultaneously. The base and mobile stations transmit / receive on separate 30kHz channels (FDD) with 45MHz spacing between the duplex channels and with carrier frequencies in bands of 25MHz from 824 - 849MHz for the reverse link and 869 - 894MHz for the forward link. This structure provides 832 duplex RF channels. Speech codecs are VSELP (Vector Sum Excited Linear Prediction) at a full-rate of 7.95kbits/s (13kbits/s with CRC error correction) and data transmission upto 9.6kbits/s is provided. Channel coding uses a rate $\frac{1}{2}$ convolutional coder with a constraint length of 6. The system uses $\pi/4$-DQPSK modulation with a roll-off of 0.35 and bit rate of 48.6kbits/s (symbol rate of 24.3kBaud).

The systems are designed to operate with a channel multipath delay spread of upto $60\mu s$ but, due to the on-air bit rate of 24.3kBaud, equalisation is not required in most situations.

A.3 PDC

The Japanese Personal Digital Cellular (PDC) is very similar to IS-54. Multiple channels using TDMA with three timeslots, each of duration 6.67ms, is used and the base and mobile stations use separate 25kHz channels for the FDD functionality with carrier frequencies around 900MHz (810 - 826MHz reverse link, 940 - 956MHz forward link) and 1.5GHz (1.429 - 1.453GHz reverse link, 1.477 - 1.501GHz forward link). The bands provide 640 duplex RF channels at the lower frequency and 960 at the higher. Speech codecs are a full-rate, 6.7kbits/s (11.2kbits/s with error correction) VSELP and recently a half-rate, 5.6kbits/s psi-CELP codec has been specified. Channel coding is via a rate $\frac{9}{17}$ convolutional coder (constraint length = 5) plus 2 slot interleaving and CRC error detection. Modulation is $\pi/4$-DQPSK with a root Nyquist filter (roll-off of 0.5), at 42kbits/s.

Channel multipath delay spread is not specified and equalisation is not required.

A.4 IS-95

IS-95 is a USA cellular standard for CDMA systems and the first implementation within the USA is presently functioning.

Base and mobile stations transmit / receive on separate 1250kHz channels with carrier frequencies around 900MHz and the band provides 20 duplex RF channels. The forward-link (base to mobile) and reverse-link operate in different manners. The basic data rate is 9.6kbits/s which is spread upto a chip rate of 1.2288Mchips/s. On the forward-link data is coded by a rate $\frac{1}{2}$ convolutional coder, interleaved and spread by 1 of 64 orthogonal spreading sequences of length 64 chips (Walsh functions). Prior to QPSK modulation, all spread sequences are scrambled by an identical PRBS generator of length 2^{15} chips in order to achieve a wideband spectrum. On the reverse link (mobile to base) the data is rate $\frac{1}{3}$ convolutional coded and each block of 6 encoded bits is mapped to 1 of 64 Walsh functions. The signal is then further spread by a factor of 4 by a mobile specific code with period $2^{42}-1$ chips to give a bit rate of 1.2288Mbits/s prior to QPSK modulation. Speech coding is via a variable rate CELP codec and data transmission is provided at 9.6kbits/s.

Receivers at both base and mobile stations use RAKE systems to combine and resolve multipath effects to reduce fading.

A.5 CT2 Common Air Interface

The UK CT2 'cordless' standard offers speech and data services between fixed and portable stations. The link operates in a TDD format with frames of 2ms duration. Carrier frequencies are from 864 - 868MHz providing 40 FDMA channels with each channel being 100kHz wide. Speech is coded via a 32kbits/s ADPCM codec and, after time compression and the addition of control bits, is Gaussian-FSK modulated at 72kbits/s. Data transmission is provided up to 2.4kbits/s. The relatively low baud rate results in equalisation being unnecessary.

A.6 DECT

The European DECT 'cordless' standard operates cordless telephone links in a TDMA and TDD format with 12 duplex slots per channel. Carrier frequencies are from 1.880 - 1.900GHz providing 10 FDMA channels with 1728kHz spacing and, in addition, a frequency hopped multiple access mode is provided. Speech coding is via a 32kbits/s ADPCM codec and modulation is Gaussian-FSK at 1152kbits/s in bursts of 5ms. Due to the high on-air symbol rate, either antenna diversity (fixed stations) or equalisation (portable stations) is required in more dispersive picocells.

A.7 PHS

The Japanese PHS 'cordless' system employs TDMA and TDD with four duplex slots per channel with each slot spanning 5ms. 77 channels are provided at carrier frequencies from 1.895 - 1.918GHz with 300kHz channel spacing. Speech coding is via a 32kbits/s ADPCM codec and channel coding via CRC detection (no correction). The modulation used is $\pi/4$-DQPSK at 384kbits/s.

A.8 PACS

The Personal Communications System (PACS) is a third generation system which aims to provide voice, video and data services for microcell or indoor use within the U.S.A. The air-radio interface uses 8 or 4 timeslot TDMA depending on whether the system uses either frequency or time division duplexing. The carriers are in bands from 1.850 - 1.910GHz and 1.930 - 1.990GHz providing 400 channels with a 300kHz spacing. Speech coding is via a 16-bit ADPCM codec operating at 32kbits/s with CRC and the modulator uses $\pi/4$-DQPSK at 384kbits/s.

A.9 TETRA — European Digital Private Mobile Radio

The Trans European Trunked Radio (TETRA) standard aims to provide moderately different voice and data services than the GSM system for non-public users such as the emergency services. Only the key air-radio interface characteristics are summarised.

The system operates in a variable TDMA format with a maximum of four timeslots per channel. The emergency services in Europe have been allocated the 380 - 400MHz band with another 20MHz band to be provided for commercial users. Speech coding is via a 4.56kbits/s CELP codec and with channel coding the rate rises to 7.2kbits/s. The four timeslot structure and overheads gives a final bit rate of 36kbits/s. The data is modulated using $\pi/4$-DQPSK with a root raised cosine filter and roll-off factor = 0.35, resulting in a 25kHz channel spacing.

The multipath delay spreads of typical channels are, in general, much less than the transmitted symbol period (55.5μs) and so in most cases equalisation is not required.

Appendix B

Mathematical Overview of the LMS and RLS Adaptation Algorithms

This appendix describes the mathematical functions required to be calculated in order to implement the LMS and RLS adaptation algorithms. Both these algorithms are designed to adjust the filter coefficients or parameters of a linear combination of the input signal such that the output of this filter matches the desired signal in an 'optimum' manner. The definition of 'optimum' differs between the algorithms and results in alternative ways of adapting the parameters of the same filter structure. The linear combiner / filter is shown in Fig. B.1 together with the signal notation.

B.1 LMS Algorithm

The LMS algorithm is designed to adjust the parameters of a linear combination of samples of the received signal. The output of the filter producing the linear combination is given by:

$$y(k) = \sum_{i=0}^{m-1} b_i(k) r(k-i) \tag{B.1}$$

$$= \underline{\mathbf{b}}^T(k)\underline{\mathbf{r}}(k) \tag{B.2}$$

where m is the order of the filter, $b_i(k)$ are the parameters and $r(k-i)$ are the received signal samples — both terms may be real or complex. Eqn. B.2 gives the vector notation,

$$\underline{\mathbf{b}}(k) = [b_0(k), \ldots, b_{m-1}(k)]^T,$$

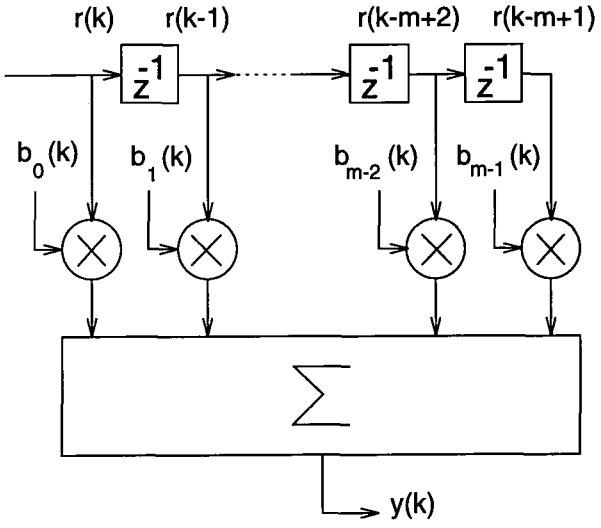

Figure B.1: Schematic of a general adaptive linear combiner / filter. In general all terms are complex valued.

$$\underline{\mathbf{r}}(k) = [r(k), \ldots, r(k-m+1)]^T$$

and $[\cdot]^T$ indicates vector transposition.

An 'optimum' filter, known as the Weiner filter, minimises the mean square error between the desired output and the filter output:

$$J_w = \frac{1}{2}E[|e(k)|^2] = \frac{1}{2}E[e(k)e^*(k)] \tag{B.3}$$

where

$$e(k) = d(k) - y(k). \tag{B.4}$$

$d(k)$ is the desired output of the filter and $*$ indicates complex conjugation. The optimum Weiner settings of the filter parameters require a knowledge of the correlation of the received signal $r(k)$ and the cross-correlation between the received signal, $r(k)$, and the desired signal, $d(k)$ [47]. Since the channel is unknown, the statistics of $r(k)$ are not computable and the Weiner parameter settings are unobtainable.

The LMS algorithm is a gradient descent based technique which replaces the error function, J_w, in Eqn. B.3 by the instantaneous error $e(k)$ (Eqn. B.4). The gradient of the new error surface, J_{lms}, is then given by:

$$\frac{\partial J_{lms}}{\partial \underline{\mathbf{b}}(k)} = -\underline{\mathbf{r}}(k)e^*(k) \tag{B.5}$$

$$= -\underline{\mathbf{r}}(k)\left(d^*(k) - \underline{\mathbf{b}}^H(k)\underline{\mathbf{r}}^*(k)\right) \tag{B.6}$$

where $(\cdot)^H$ indicates the complex conjugate transposed vector (Hermitian). The filter pa-

rameters are updated in the direction of the negative of the conjugate of the error gradient:

$$\underline{\mathbf{b}}(k+1) = \underline{\mathbf{b}}(k) - \mu \left(\frac{\partial J_{lms}}{\partial \underline{\mathbf{b}}(k)} \right)^* \tag{B.7}$$

$$= \underline{\mathbf{b}}(k) + \mu \underline{\mathbf{r}}^*(k) e(k) \tag{B.8}$$

where μ is the stepsize which governs the rate of descent (and final mean square error).

The algorithm is not a true 'gradient descent' technique due the approximation of the mean square error surface by the instantaneous square error surface. This leads to slower convergence times and the 'converged' parameters are noisy versions of the Weiner parameters [47]. The LMS algorithm's behaviour is well documented and the trade-off between fast convergence (large μ) / low mean square error (small μ), stability (upper bound on μ) and behaviour in time-varying channels is largely understood [47, 59].

Variations exist including the normalised LMS which modulate the stepsize, μ, according to the power of the input signal $\underline{\mathbf{r}}(k)$. If this input signal is large the algorithm suffers from gradient noise amplification due to the approximation of the mean square error by the instantaneous square error. The stepsize, μ, is replaced by:

$$\mu(k) = \frac{\mu}{\|\underline{\mathbf{r}}(k)\|^2} \tag{B.9}$$

Block LMS algorithms replace the statistical expectation operator in Eqn. B.3 by a short-term time average of the instantaneous square error.

B.2 RLS Algorithm

The RLS algorithm is designed to adapt the parameters of an FIR filter but uses a different error criterion to the LMS algorithm. The RLS algorithm is a special case of the more general Kalman filter and uses input signal information since the algorithm was first initialised to update its filter parameters [47, 93, 94]. This results in a much faster rate of convergence than the LMS algorithm but involves significantly more computation. The least square error criterion does not use any statistical measures, as does the Weiner filter (ensemble averages) or its LMS approximation, but bases its error function on time averages which, in the case of recursive least squares, can be continually updated.

The output of the filter is again given by (see Section B.1):

$$y(k) = \sum_{i=0}^{m-1} b_i(k) r(k-i) \tag{B.10}$$

$$= \underline{\mathbf{b}}^T(k) \underline{\mathbf{r}}(k) \tag{B.11}$$

and the error by:

$$e(k) = d(k) - y(k). \tag{B.12}$$

The error function, $J_{rls}(k)$, is now:

$$J_{rls}(k) = \sum_{t=0}^{k} \beta(k,t)|e(t)|^2 \tag{B.13}$$

where $\beta(k,t)$ is a weighting factor to allow time-variations in the channel being equalised. This function is typically an exponentional function defined by:

$$\beta(k,t) = \lambda^{k-t} \tag{B.14}$$

where λ is called the forgetting factor.

For this error function, the optimum settings (in the least square error sense) of the filter parameters, $\underline{\mathbf{b}}_{opt}(k)$, are given (in conjugate form) by:

$$\underline{\mathbf{b}}^*_{opt}(k) = \underline{\Phi}(k)^{-1}\underline{\Theta}(k) \tag{B.15}$$

where the m-by-m correlation matrix, $\underline{\Phi}(k)$, and the m-by-1 cross-correlation matrix, $\underline{\Theta}(k)$, are given by:

$$\underline{\Phi}(k) = \sum_{t=0}^{k} \lambda^{k-t}\underline{\mathbf{r}}(t)\underline{\mathbf{r}}^H(t) \tag{B.16}$$

$$\underline{\Theta}(k) = \sum_{t=0}^{k} \lambda^{k-t}\underline{\mathbf{r}}(t)d^*(t). \tag{B.17}$$

The correlation and cross-correlation matricies may be updated recursively (to save on computation and storage) by using the equations:

$$\underline{\Phi}(k) = \lambda\underline{\Phi}(k-1) + \underline{\mathbf{r}}(k)\underline{\mathbf{r}}^H(k) \tag{B.18}$$

$$\underline{\Theta}(k) = \underline{\Theta}(k-1) + \underline{\mathbf{r}}(k)d^*(k). \tag{B.19}$$

Using the matrix inversion lemma [47] a recursive relation for the inverse correlation matrix may be expressed as:

$$\underline{\Phi}^{-1}(k) = \lambda^{-1}\underline{\Phi}^{-1}(k-1) - \lambda^{-1}\underline{\mathbf{k}}(k)\underline{\mathbf{r}}^H(k)\underline{\Phi}^{-1}(k-1) \tag{B.20}$$

where the Kalman gain, $\underline{\mathbf{k}}(k)$ is given by:

$$\underline{\mathbf{k}}(k) = \frac{\lambda^{-1}\underline{\Phi}^{-1}(k-1)\underline{\mathbf{r}}(k)}{1 + \lambda^{-1}\underline{\mathbf{r}}^H(k)\underline{\Phi}^{-1}(k-1)\underline{\mathbf{r}}(k)} \tag{B.21}$$

Finally, a recursive relation for the parameter update may be used which removes the need to calculate the cross-correlation matrix, $\underline{\Theta}(k)$. This involves calculating the 'innovation', $\alpha(k)$,

$$\alpha(k) = d(k) - \underline{\mathbf{b}}^T(k-1)\underline{\mathbf{r}}(k). \tag{B.22}$$

The second term in this equation is a prediction of the desired signal and the difference between the actual and prediction, the innovation, represents new information since it has not been predicted. The final expression for the parameter update equation is:

$$\underline{\mathbf{b}}(k) = \underline{\mathbf{b}}(k-1) + \underline{\mathbf{k}}^*(k)\alpha(k) \tag{B.23}$$

Thus the innovation is multiplied by the conjugate Kalman gain to provide a correction term for the filter parameters. Equations B.21, B.22, B.23 and B.20 give the final set of calculations required at each sample period and are clearly more complex than the LMS update equations.

Variants of this standard form of RLS exist which compute the Kalman gain in by different, more rapid, method. Further details may be found in [47].

Appendix C

True CIR Model Calculation

The combined impulse response of transmit filter, time-varying channel and receive filter cannot be obtained by a normal convolution of the three impulse responses. The receive filter delay line contains energy from symbols arriving with similar propagation delays but with varying amplitudes and phase shifts. This appendix details the calculation of the true instantaneous channel impulse response — i.e. the amplitude and phase of each transmitted symbol component of the received signal.

The transmitted signal, $x(k)$, (sampled at a rate $f_s = \frac{1}{T_s}$) is passed through a transmit filter with impulse response $f_T(m)$ (operating at a rate qf_s) to give the transmitted signal $u(m)$ (see Fig. C.1). The higher sample rate is an approximation to continuous time. The transmitted signal passes through the time-varying channel filter, $h_i(m)$, and filtered by the receive filter, $f_R(m)$, and downsampled by a factor q to give the received signal, $r(k)$.

The transmitted signal may be written:

$$u(m) = u(qk+p) = \sum_{i=0}^{\frac{n_1-1}{q}} f_T(qi+p)x(k-i) \tag{C.1}$$

where p is chosen ($p = 0, 1, \cdots, q-1$) such that ($m = qk+p$). The length of the transmit filter impulse response is n_1, spaced with delays of $\left(\frac{1}{qf_s}\right)$.

The signal passes through the channel filter to give a signal:

$$v(m) = \sum_{i=0}^{L} h_i(m)u(m-i) \tag{C.2}$$

where $(L+1)$ is the length of the channel filter impulse response, spaced with delays of $\left(\frac{1}{qf_s}\right)$.

Finally $v(m)$ is filtered by the receive filter and downsampled by a factor q to give the

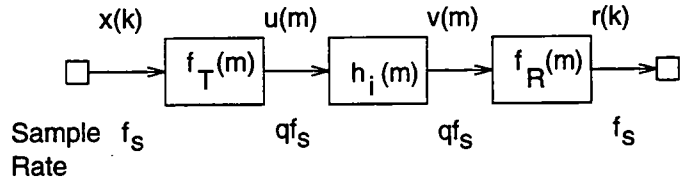

Figure C.1: Representation of transmit, channel and receive filters for combined impulse response calculation.

received signal:

$$r(k) = r\left(\frac{m}{q}\right) = \sum_{i=0}^{n_1-1} f_R(i)v(m-i) \tag{C.3}$$

n_1 is the length of the receive filter impulse response, spaced with delays of $\left(\frac{1}{qf_s}\right)$ and an integer multiple of q.

Each received sample contains information from $\left(\frac{2n_1+L-1}{q}\right)$ transmitted symbols and, assuming a causal channel model, the combined delay of the transmit filter, channel filter and receive filter is $\left(\frac{n_1-1}{q}\right)$ symbol periods.

The combined impulse response may be calculated by substituting Eqn. C.1 into Eqn. C.2 and rearranging $v(m)$ in terms of $x(k), \cdots, x(k-\frac{n_1-1}{q})$,

$$v(m) = \sum_{i=0}^{L} h_i(m) \sum_{j=0}^{\frac{n_1-1}{q}} f_T(qj+p-i)x(k-j) \tag{C.4}$$

where p is chosen $(p = 0, 1, \cdots, q-1)$ such that $(m = qk+p)$.

The resulting sequence of coefficients of $x(k), \cdots, x(k-\frac{n_1-1}{q})$ are stored and a new coefficient sequence is formed from the stored coefficients corresponding to a particular symbol, $x(k-d)$. This sequence is then convolved with the receive filter impulse response, f_R, to give the value of the combined impulse response value at time $k-d$. This final convolution is repeated for all sequences of coefficients of $x(k-d)$, where $d = 0, \cdots, \left(\frac{2n_1+L-1}{q}\right)$.

Downsampling is achieved by calculating these final set of convolutions once every q^{th} sample, taking into account the relative shift of the receive filter impulse response and the symbol coefficient sequence.

Appendix D

Publications

1. M.J.BRADLEY and P.MARS. A critical assessment of recurrent artificial neural networks as adaptive equalizers in digital communications. In *IEE Colloquium on Applications of neural networks to signal processing*, pages 11/1 — 11/4. London, UK. 15 December 1994. Digest No. 1994/248.

2. M.J.BRADLEY and P.MARS. Application of recurrent neural networks to communication channel equalization. In *International Conference on Acoustics Speech and Signal Processing (ICASSP-95)*, pages 3399 — 3402. Detroit, USA. 8 — 12 December 1995.

3. M.J.BRADLEY and P.MARS. Analysis of recurrent neural networks as digital communication channel equalizers. In *Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications 2*, pages 1 — 8. Stockholm, Sweden. 22 — 24 May 1995.

4. M.J.BRADLEY and P.MARS. Application of multiple channel estimators in MLSE detectors for fast time-varying and frequency selective channels. *Electronics Letters*, pages 620 — 621, vol. 32, no. 7, 28 March 1996.

5. M.J.BRADLEY and P.MARS. Application of multiple channel estimators in MLSE detectors for fast time-varying and frequency selective channels. In *International Conference on Universal Personal Communications (ICUPC-96)*, Cambridge, Mass. USA. 29 September — 2 October 1996.

References

[1] A.S.Spanias. Speech coding: A tutorial review. *Proc. IEEE*, 82(10):1541–1582, 1994.

[2] J.G.Proakis. *Digital Communications.* McGraw Hill, 1989.

[3] K.Feher. *Digital Communications: Satellite / Earth Station Engineering.* Prentice Hall, 1983.

[4] T.S.Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall, 1996.

[5] R.Steele. *Mobile Radio Communications.* Pentech Press, 1992.

[6] G.Ungerboeck. Trellis-coded modulation with redundant signal sets. Part 1: An introduction. *IEEE Communications Magazine*, 25(2):5–11, 1987.

[7] D.D.Falconer et al. Time division multiple access methods for wireless personal communications. *IEEE Communications Magazine*, 33(1):50–57, January 1995.

[8] A.H.Aghvami. Digital modulation techniques for mobile and personal communication systems. *IEE Electronics and Communications Engineering Journal*, 5(3):125–132, June 1993.

[9] L.E.Franks. Carrier and bit synchronisation in data communication - A tutorial review. *IEEE Trans. on Communications*, 28(8):1107–1120, 1980.

[10] G.A.Halls. HIPERLAN: the high performance radio local area network standard. *IEE Electronics and Communication Engineering Journal*, pages 289–296, December 1994.

[11] C.A.Siller. Multipath propagation. *IEEE Communications Magazine*, 22(2):6–15, 1984.

[12] J.B.Andersen et al. Propagation measurements and models for wireless communications channels. *IEEE Communications Magazine*, 33(1):42–49, January 1995.

[13] C.C.E.Baden et al. Planning tools for mobile networks. *IEE Electronics and Communication Engineering Journal*, 5(5):309–314, October 1993.

[14] CCIR. *Fixed services at frequencies below about 30MHz: Recommendations of the CCIR, vol.3*, 1990.

[15] ETSI. GSM Recommendation 05.05. *Radio transmission and reception for GSM 900 and DCS 1800.*

[16] H.G.Brierley. *Telecommunications Engineering.* Arnold, 1986.

[17] G.Roda. *Troposcatter Radio Links.* Artech House, 1988.

[18] P.Monsen. Theoretical and measured performance of a DFE modem on a fading multipath channel. *IEEE Trans. on Communications*, 25(10):1144–1154, 1977.

[19] P.A.Bello. A troposcatter channel model. *IEEE Trans. on Communications*, 17(2):130–137, 1969.

[20] S.Stein. Fading channel issues in system engineering. *IEEE Trans. on Selected Areas in Communications*, 5(2):68–89, 1987.

[21] C.M.Rush. Ionospheric radio propagation models and predictions - A mini review. *IEEE Trans. on Antennas and Propagation*, 34(9):1163–1170, 1986.

[22] Schwartz, Bennett, and Stein. *Communication Systems and Techniques.* McGraw Hill, 1966.

[23] C.Watterson, J.R.Juroshek, and W.D.Bensema. Experimental confirmation of an HF channel model. *IEEE Trans. on Communications*, 18(6):792–803, 1970.

[24] A.P.Clark and F.McVerry. Channel estimation for an HF radio link. *IEE Proc. Pt. F*, 128(1):33–48, 1981.

[25] CCIR report 322-3. *Characteristics and applications of atmospheric radio noise data*, 1988.

[26] Alta Group of Cadence Design Systems Inc. *Signal Processing WorkSystem. V3.2*, 1995.

[27] A.P.Clark. *Adaptive Detectors for Digital Modems.* Pentech Press, 1989.

[28] I.Crohn and E.Bonek. Modelling of intersymbol-interference in a Rayleigh fast fading channel with typical delay power profiles. *IEEE Trans. on Vehicular Technology*, 41(4):438–446, 1992.

[29] C.Loo and N.Secord. Computer models for fading channels with applications to digital transmission. *IEEE Trans. on Vehicular Technology*, 40(4):700–706, 1991.

[30] S.U.H.Qureshi. Adaptive equalization. *Proc. IEEE*, 73(9):1349–1387, 1985.

[31] S.Haykin. *Blind Deconvolution.* Prentice Hall, 1994.

[32] G.D.Forney. Jr. The Viterbi algorithm. *Proc. IEEE*, 61(3):268–278, 1973.

[33] G.D.Forney. Jr. Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference. *IEEE Trans. on Information Theory*, 18(3):363–378, 1972.

[34] A.Duel-Hallen and C.Heegard. Delayed decision-feedback sequence estimation. *IEEE Trans. on Communications*, 37(5):428–436, 1989.

[35] M.V.Eyuboglu and S.U.H.Qureshi. Reduced-state sequence estimation with set partitioning and decision feedback. *IEEE Trans. on Communications*, 36(1):13–20, 1988.

[36] D.Williamson, R.A.Kennedy, and G.W.Pulford. Block decision feedback equalization. *IEEE Trans. on Communications*, 40(2):255–264, 1992.

[37] K.Abend and B.D.Fritchman. Statistical detection for communication channels with intersymbol interference. *Proc. IEEE*, 58(5):779–785, 1990.

[38] R.P.Lippmann. An introduction to computing with neural networks. *IEEE ASSP magazine*, pages 4–22, April 1987.

[39] S.Haykin. *Neural Networks: A Comprehensive Foundation.* Macmillan, 1994.

[40] G.J.Gibson, S.Siu, and C.F.N.Cowan. Multilayer perceptron structures applied to adaptive equalisers for data communications. In *ICASSP-89*, pages 1183–1186, 1989.

[41] M.Meyer and G.Pfeiffer. Multilayer perceptron based decision feedback equalisers for channels with intersymbol interference. *IEE Proc. Pt. I*, 140(6):420–424, 1993.

[42] G.J.Gibson and C.F.N.Cowan. On the decision regions of multilayer perceptrons. *Proc. IEEE*, 78(8):1590–1594, 1990.

[43] S.Chen, B.Mulgrew, and S.McLaughlin. Adaptive Bayesian equalizer with decision feedback. *IEEE Trans. on Signal Processing*, 41(9):2918–2927, 1993.

[44] R.J.Williams and D.Zisper. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1:270–280, 1989.

[45] G.Kechriotis, E.Zervas, and E.S.Manolakos. Using recurrent neural networks for adaptive communication channel equalization. *IEEE Trans. on Neural Networks*, 5(2):267–278, 1994.

[46] J.Salz. Optimum mean-square decision feedback equalization. *Bell System Technical Journal*, pages 1341–1373, October 1973.

[47] S.Haykin. *Adaptive Filter Theory, 2nd Edition.* Prentice Hall, 1991.

[48] E.Kreyszig. *Advanced Engineering Mathematics, 6th Edition.* Wiley, 1988.

[49] W.T.Webb and R.D.Shenton. Pan-European railway communications: where PMR and cellular meet. *IEE Electronics and Communication Engineering Journal*, 6(4):195–202, August 1994.

[50] S.A.Fechtel and H.Meyer. An investigation of channel estimation and equalization techniques for moderately rapid fading HF channels. In *ICC-91*, pages 25.2.1–25.2.5, 1991.

[51] M.Stojanovic, J.G.Proakis, and J.A.Catpovic. Analysis of the impact of channel estimation errors on the performance of a decision feedback equalizer in fading multipath channels. *IEEE Trans. on Communications*, 43(2/3/4):877–885, 1995.

[52] R.A.Ziegler and J.M.Cioffi. Estimation of time-varying digital radio channels. *IEEE Trans. on Vehicular Technology*, 41(2):134–151, 1992.

[53] M.R.L.Hodges. The GSM radio interface. *British Telecom Tech. Journal*, 8(1), 1990.

[54] F.R.Magee and J.G.Proakis. Adaptive maximum-likelihood sequence estimation for digital signaling in the presence of intersymbol interference. *IEEE Trans. on Information Theory*, pages 120–124, January 1973.

[55] S.A.Fechtel and H.Meyer. Optimal parametric feedforward estimation of frequency selective fading radio channels. *IEEE Trans. on Communications*, 42(2/3/4):1639–1650, 1994.

[56] G.F.M.Beenker, T.A.C.M.Classen, and P.W.C.Hermens. Binary sequences with a maximally flat amplitude spectrum. *Phillips Journal of Research*, 40(5):289–304, 1985.

[57] L.Ljung and T.Soderstrom. *Theory and Practice of Recursive Identification*. MIT Press, 1983.

[58] A.P.Clark and S.G.Jayasinghe. Channel estimation for land mobile radio systems. *IEE Proc., Pt.F,*, 134(4):383–393, 1987.

[59] B.Widrow, J.M.McCool amd M.G.Larimore, and C.R.Johnson. Stationary and nonstationary learning characteristics of the LMS adaptive filter. *Proc. IEEE*, 64(8):1151–1162, 1976.

[60] W.A.Gardener. Nonstationary learning characteristics of the LMS algorithm. *IEEE Trans. on Circuits and Systems*, 34(10):1199–1207, 1987.

[61] S.Jaggi and A.B.Martinez. Upper and lower bounds on the misadjustment in the LMS algorithm. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 38(1):164–166, 1990.

[62] E.Eleftheriou and D.D.Falconer. Tracking properties and steady-state performance of RLS adaptive filter algorithms. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 34(5):1097–1109, 1986.

[63] S.Chen, S.McLaughlin, B.Mulgrew, and P.M.Grant. Adaptive Bayesian decision feedback equalizer for dispersive mobile radio channels. *IEEE Trans. on Communications*, 43(5):1937–1946, 1995.

[64] D.D.Falconer S.N.Crozier and S.A.Mahmoud. Least sum of squared errors (LSSE) channel estimation. *IEE Proc. Pt. F*, 138(4):371–378, 1991.

[65] J.C.S.Cheung and R.Steele. Soft-decision feedback equalizer for continuous phase modulated signals in wideband mobile radio channels. *IEEE Trans. on Communications*, 42:1628–1638, 1994.

[66] J.G.Proakis and D.G.Manolakis. *Introduction to Digital Signal Processing.* Macmillan, 1988.

[67] J.Lin, J.G.Proakis, F.Ling, and H.Lev-Ari. Optimal tracking of time-varying channels: A frequency domain approach for known and new algorithms. *IEEE Trans. on Selected Areas in Communications*, 13(1):141–153, 1995.

[68] J.Lin, F.Ling, and J.G.Proakis. Fading channel tracking properties of several adaptive algorithms for North American digital cellular system. In *Proc. IEEE 43rd Vehicle Technol. Conf.*, pages 273–276, 1993.

[69] S.Chen, S.McLaughlin, and B.Mulgrew. Complex-valued radial basis function network. Part 2: Application to digital communications channel equalisation. *Signal Processing*, 36(2):175–188, February 1994.

[70] D.Dzung. Error probability of MLSE equalizers using imperfect channel measurements. In *ICC-91*, pages 19.4.1–19.4.5, 1991.

[71] H.Kubo, K.Murakami, and T.Fujino. An adaptive maximum-likelihood sequence estimator for fast time-varying intersymbol interference channels. *IEEE Trans. on Communications*, 42(2):1872–1880, 1994.

[72] R.Raheli, A.Polydoros, and C.Tzou. Per-survivor processing: A general approach to MLSE in uncertain enviroments. *IEEE Trans. on Communications*, 43(2):354–364, 1995.

[73] R.Raheli, G.Marino, and P.Castoldi. Per-survivor processing and tentative decisions: What is in between? *IEEE Trans. on Communications*, 44(2):127–129, 1996.

[74] B.Lundborg, R.Skartlien, V.Jodalen, M.Bröms, P.S.Cannon, M.J.Angling, N.C.Davies, and K.W.Moreland. Scandanavian DAMSON measurements of multipath and Doppler characteristics over high latitude HF paths. In *Nordic Shortwave Conf. (HF'95)*, pages 1.1.1–1.1.10, August 1995.

[75] P.S.Cannon, M.J.Angling, T.Willink, K.W.Moreland, N.C.Davies, B.Lundburg, V.Jodalen, and M.Bröms. Requirements on channel probes for automatic channel selection. In *IEE Colloquium on Frequency Selection and Management Techniques for HF Communications*, pages 24/1–24/6, 1996.

[76] ETSI. I-ETSI 300 382-2:1995. *TETRA standard*, 1995.

[77] A.Morgul and D.Dzung. Decision directed channel parameter estimation and tracking using erroneous detectors. *Signal Processing*, 25:307–318, 1991.

[78] J.Hagenauer and P.Hoeher. A Viterbi algorithm with soft-decision outputs and its applications. In *Globecom '89*, pages 47.1.1 - 47.1.7, 1989.

[79] R.A.Iltis, J.J.Shynk, and K.Giridhar. Bayesian algorithms for blind equalization using parallel adaptive filtering. *IEEE Trans. on Communications*, 42(2/3/4):1017–1032, 1994.

[80] J.K.Tugnait. Blind estimation of digital communication channel impulse response. *IEEE Trans. on Communications*, 42(2/3/4):1606–1616, 1994.

[81] Y.Sato. A blind sequence detection and its application to digital mobile communications. *IEEE Trans. on Selected Areas in Communications*, 13(1):49–58, 1995.

[82] F.Chen, S.McLaughlin, and B.Mulgrew. Blind equalization of nonminimum phase channels: Higher order cimulant based algorithm. *IEEE Trans. on Signal Processing*, 41(2):681–691, 1993.

[83] W.W.Wu, E.F.Miller, W.L.Pritchard, and R.L.Pickholtz. Mobile satellite communications. *Proc. IEEE*, 82(9):1431–1448, 1994.

[84] *ITU-R report 8/BL/42-E Annex 1*, 1994.

[85] J.Padgett, C.G.Gunther, and T.Hattori. Overview of wireless personal communications. *IEEE Communications Magazine*, 33(1):28–41, January 1995.

[86] J.Cosmas et al. Overview of the mobile communications programme of RACE II. *IEE Electronics and Communications Engineering Journal*, 7(4):155–167, August 1995.

[87] P.A.Ramsdale. The development of personnal communications. *IEE Electronics and Communication Engineering Journal*, 8(3):143–151, June 1996.

[88] P.Whitehead. The other communications revolution. *IEE Review*, 42(4):167–170, 1996.

[89] P.Jung P.W.Baier and A.Klein. Taking the challenge of multiple access for third-generation cellular mobile radio systems — A European view. *IEEE Communications Magazine*, 34(2):82–89, 1996.

[90] R.Steele. The evolution of personal communications. *IEEE Personal Communications Magazine*, 1(2):6–11, 1994.

[91] W.C.Y.Lee. Overview of cellular CDMA. *IEEE Trans. on Vehicular Technology*, 40(2):291–302, 1991.

[92] W.Webb. TETRA in perspective: Complimentary and competetive systems. WWW document: www.smithsys.co.uk/smithsys/techp/tetra2/tetra2.html.

[93] C.F.N.Cowan and P.M.Grant. *Adaptive Filters.* Prentice Hall, 1985.

[94] B.Mulgrew and C.F.N.Cowan. *Adaptive Filters and Equalisers.* Kluwer Academic, 1988.