Developments In Non-Linear Equalization

Graham W. Pulford

B.E., B.Sc. (UNSW)

February 1992

A thesis submitted for the degree of Doctor of Philosophy of the Australian National University

Department of Systems Engineering
Research School of Physical Sciences and Engineering
The Australian National University

Declaration

The contents of this thesis are the results of original research, and have not been submitted for a higher degree at any other university or institution.

A number of papers resulting from this work have been submitted to refereed journals:

[J1] R. A. Kennedy, G. W. Pulford, B. D. O. Anderson, and R. R. Bitmead, "When Has a Decision-Directed Equalizer Converged ?", *IEEE Transactions on Communications*,vol. 37, no. 8, pp. 879-884, Aug. 1989.

[J2] D. Williamson, R. A. Kennedy, and G. W. Pulford, "Block Decision Feedback Equalization", to appear in *IEEE Transactions on Communications*, Feb. 1992.

[J3] G. W. Pulford, R. A. Kennedy, and B. D. O. Anderson, "A Neural Network Structure for Decision Feedback Equalization", submitted to *IEEE Transactions on Information Theory*, July 1991.

[J4] G. W. Pulford, and R. A. Kennedy, "Maximum A Posteriori Decision Feedback Detection", submitted to *IEE Proceedings Part I Communications, Speech and Vision*, Nov. 1991.

A number of papers have been presented at conferences. Some of the material covered in these papers overlaps with that covered in the publications listed above:

[C1] G. W. Pulford, R. A. Kennedy, D. Williamson, and B. D. O. Anderson, "Error Analysis of the Block DFE using Finite State Markov Processes", *Proc. The Second International Symposium on Signal Processing and its Applications, ISSPA '90*, Vol. 2, pp. 687-690, Gold Coast, Australia, Aug. 1990.

[C2] D. Williamson, R. A. Kennedy, and G. W. Pulford, "A New Equalizer Structure Using Vector Quantization Based on the Decision Feedback Principle", *Proc. The Second International Symposium on Signal Processing and its Applications, ISSPA '90*, Vol. 2, pp. 683-686, Gold Coast Australia, Aug. 1990.

[C3] G. W. Pulford, and R. A. Kennedy, "Maximum A Posteriori Decision Feedback Detection", *Proc. International Symposium on Information Theory and its Applications, ISITA '90*, Waikiki, Hawaii, pp. 287-290, Nov. 1990.

[C4] G. W. Pulford, R. A. Kennedy, and B. D. O. Anderson, "A Neural Net Structure for Decision Feedback Equalisers", *Proc. Second Aust. Conf. Neural Networks ACNN'91*, pp. 223-226, Sydney, Australia, Feb. 1991.

[C5] G. W. Pulford, R. A. Kennedy, and B. D. O. Anderson, "Neural Net Structure for Emulating Decision Feedback Equalisers", *Proc. Int. Conf. Acoust. Speech Sig. Process. ICASSP'91*,Vol. 3,pp. 1517-1520, Toronto, Canada, May 1991.

[C6] D. M. Bauer, R. A. Kennedy, and G. W. Pulford, "Joint Estimation of Trellis-Coded Data and Channel Using the M-Algorithm", submitted to *Proc. Int. Conf. on Acoustics Speech and Sig. Proc. ICASSP'92*, San Fransisco, Calif., Feb. 1992, *john wested*[1].

The work described in this thesis has been carried out in collaboration with a number of people. They are: Prof. Brian D. Anderson and Dr. Rodney A. Kennedy, who were my supervisors, Prof. Darrell Williamson and Mr. Peter Do. Some work, not explicitly represented in this thesis, was carried out in conjunction with Dr. Diethelm Bauer. However, the majority of the work, approximately 75 %, is my own.

Graham Pulford

Graham Pulford.

February 6, 1992

[1] "It's the fish John West reject that make John West the best."

Acknowledgements

I would particularly like to thank my supervisors Professor Brian Anderson and Dr. Rodney Kennedy. Needless to say, much less would have been achieved without their expert help and guidance. I would like to thank Prof. Nils Holte from NTH Trondheim and Hans Guren of E. B. Technology, Norway for making available some simulation software. Mr Peter Doe is acknowledged for his help with the computing in [C4,C5]. Thanks also to Professor Darrell Williamson for his helpful collaboration during the initial stages of the work in chapter 3. I am indebted to the ANU centre for Information Science Research for their financial assistance over the last three years. I must also thank Dr. David Clements who supervised me during my undergraduate studies, and without whose encouragement I would not have embarked on this course of further study. Last, but definitely not least, a big thank you to my parents and to Catherine for their continued understanding and support.

Abstract

The focus of this thesis is the non-linear equalization of channels for digital communication. Throughout, we assume a baseband PAM[2] transmission system for uncoded data on a dispersive channel with additive noise. The emphasis is on theoretical development and analysis of new equalizer structures for the removal of intersymbol interference and the recovery of the transmitted data.

We present a multi-layer non-linear feedforward processor that emulates a decision feedback equalizer (DFE) on a finite impulse response channel. This feedforward emulator has close structural ties with multi-layer perceptron neural networks, but is more readily analysed. It derives from a non-adaptive decision feedback equalizer through a process of recursive unwrapping followed by truncation. We extend the finite-state Markov process techniques for the DFE to analyse this new structure, obtaining bounds on the noiseless error probability. We go on to develop training sequence adaptation rules using a stochastic gradient descent strategy and verify their convergence via numerical simulation.

We generalise conventional decision feedback equalization to block decision feedback equalization using a block processing channel model combined with a fixed-lag maximum *a posteriori* estimator and decision feedback. We consider various realisations of block DFEs, generating single decisions and blocks of decisions, and ascertain their performance under simulation. We investigate the extremes of performance of the block DFE obtainable by varying the dimension of the block processing and the decision device, as well as its behaviour for high signal-to-noise ratios. These extremes are: the conventional DFE; the Viterbi decoder; and the minimum bit error rate detector. We show how block decision feedback equal-

[2]Pulse Amplitude Modulated

ization can be applied to quadrature amplitude modulation signalling on infinite impulse response and non-linear channels with coloured noise. We derive minimum mean-square error and gradient descent adaptation rules for block DFEs with binary signalling on finite impulse response channels.

We provide performance analyses of the non-adaptive two-dimensional block DFE operating on low order channels. We give a direct calculation of the primary bit error rate. We treat the noiseless propagation of initial decision errors through decision feedback—firstly by deriving sufficient conditions on the channel impulse response parameters, then by modelling error propagation as a finite-state Markov process. The latter approach yields necessary and sufficient conditions on the channel which guarantee a bounded error recovery time and furthermore allows us to classify channels according to the statistics of their noiseless error recovery times. We also indicate how to include the effects of noise into the analysis.

Lastly, we derive an enhanced block DFE—the maximum *a posteriori* decision feedback detector—which generalises the block DFE through the incorporation of error event probabilities in its decision criterion. We show how this strategy relates to classical non-linear detection. We give numerical examples for realisations on low order channels, and investigate the decision regions arising from the new detection criterion.

Contents

List of Tables

List of Figures

Glossary

LE	Linear equalizer
DFE	Decision feedback equalizer
FFE	Feedforward emulator
FSMP	Finite-state Markov process
MAP	Maximum *a posteriori*
BER	Bit error rate
MLSE	Maximum likelihood sequence estimation/estimator
SNR	Signal-to-noise ratio
ISI	Intersymbol interference
ARMA	Auto-regressive moving average
k	Discrete time index
z^{-1}	Backward shift operator, $z^{-1}u_k = u_{k-1}$
$\mathbb{R}$	Real numbers
$\mathbb{C}$	Complex plane
$\mathbb{B}$	Binary symbol set: $\{-1, +1\}$
$\mathbb{E}$	Binary error set: $\{-2, 0, +2\}$
$\overline{\mathbb{E}}$	Extended binary error set: $\{-2, -1, 0, 1, 2\}$
L	Channel order
FIR(L)	Finite impulse response (filter with L+1 taps)
IIR	Infinite impulse response
iid	Independent and identically distributed
K	Transmitted message length
u_k	Channel input symbol at time k
y_k	Sampled received signal at time k
n_k	Additive noise sample at time k
h_i	i^{th} sampled channel impulse response coefficient
$H(z)$	z-Transform of discrete-time sequence $\{h_k\}$
$\hat{u}_k$	Input symbol estimate at time k, or decision
d_i	DFE feedback filter tap i
$d_i(k)$	Adaptive FFE weight i at time k

$\{A, b, c, d\}$	Single-input, single-output ARMA channel realisation
$\{F, G, H, D\}$	Block processing channel realisation
p	Block size, equal to decision delay -1
U_k, Y_k, N_k	p-vectors of successive u_k, y_k, n_k respectively
x_k	Channel state at time k, L-vector for FIR(L) channel
$\widehat{x}_k$	Channel state estimate at time k
E_k	State estimation error or error state at time k
e_{k-j}	Components of E_k, $j = L, \ldots, 1$
Z_k	Decision device input p-vector, components $z_{k,1}, \ldots, z_{k,p}$
$\widehat{U}_k$	p-vector estimate of U_k
$Pr(E)$	Probability of event E
$p_v(v_k)$	Probability density of *iid* random variables $\{v_k\}$
P	Transition probability matrix of finite state Markov process
p_{ij}	(i, j) element of transition probability matrix
π_k	State probability distribution vector of FSMP at time k
$w_{ij}(k)$	Branch weight connecting node i to node j (layer dependent)
$y_i^j(k)$	Output of node (i, j) at time k
Λ	Number of layers in feedforward emulator
$A(k)$	Aggregated state of FSMP $\{E_k, -E_k\}$, values $A(i)$
M	Number of aggregated states, $M = \frac{1}{2}(3^L + 1)$
R	Error recovery time
$\mathcal{E}\{\cdot\}$	Mathematical expectation

Chapter 1

Introduction

1.1 The Equalization Problem

1.1.1 Lead In

The equalization problem arises in the area of digital communication. It is desired to transmit a stream of discrete-time digital information through some physical medium, called the *channel*, to a receiver. Physical channels, having finite bandwidth, tend to introduce distortion of the transmitted data which manifests itself in the time domain as a spreading of the energy or duration of the individual data pulses. The continuous received waveform is sampled at the receiver, generating a train of pulses. For practical sampling rates, each received pulse contains contributions from more than one transmitted pulse. This dispersion of information is known as *intersymbol interference* (ISI). At high data rates or on highly dispersive channels ISI becomes the major factor hindering the reliable recovery of the transmitted signals. The part of the receiver which is responsible for the removal or mitigation of the effects of ISI is called an *equalizer*. The compensation process itself is referred to as *equalization*.

In many areas of high speed digital communication on band-limited channels, such as data modems for telephone lines, an equalizer is indispensable. Without equalization, the ISI introduced by the channel can significantly degrade system performance resulting in high error rates. There are essentially two ways of combating theses effects. The first is to use some form of coding prior to transmission

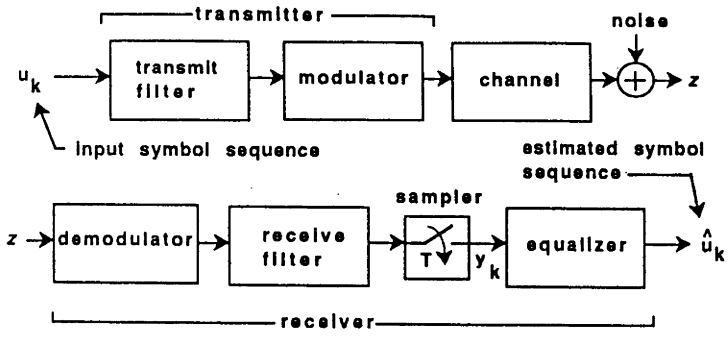

Figure 1-1: Communication system model.

followed by decoding at the receiver—although coding does not remove ISI, it introduces redundancy in the transmission so that more errors can be tolerated. The second is to develop new and better equalizer structures. In achieving transmission rates that approach the theoretical limits [1], both these approaches need to be contemplated. Practical developments, such as very high speed digital signal processing chips, mean that more and more numerically intensive processing algorithms can now be implemented, *e.g.*, trellis-coded modulation schemes [2]. We refer the reader to [3] for a review of coded modulation techniques.

While channel coding (as opposed to source coding) seeks to increase the rate at which information can be sent with a given reliability, equalization corrects the distortion introduced during transmission and allows still higher data rates. We mention that the two functions of decoding and equalization can often be combined. This is true in the case of trellis-coding and Viterbi decoding [4]. The same may be true of block-coding and block decision feedback equalization (chapter 3). We will, however, only be concerned with the design of equalizers for uncoded data, the incorporation of coding being a possible subject for future research.

In the following section we present the basic pulse amplitude modulation system model underlying the development of the various equalizers which form the basis for chapters 2-5. This model is a commonly adopted starting point for problems in channel equalization. We also review conventional equalization strategies which are important in understanding the developments that we will be describing.

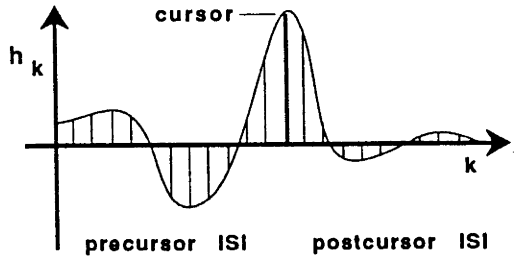

Figure 1-2: Sampled channel impulse response.

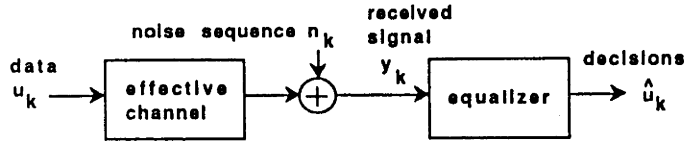

Figure 1-3: Baseband equivalent model.

1.1.2 Digital Communication System Model

We restrict the present discussion to *pulse amplitude modulation* [5] (PAM) over a linear channel. The system block diagram is given in Fig.1-1. The input symbols or data $\{u_k\}$ take values in a discrete set (for binary transmission the symbol set is $\{-1, +1\}$), and are indexed in time by the subscript k (representing the k^{th} sampling instant). The data are passed through a transmitter filter and modulated by a carrier signal before input to the channel (which is often assumed to be linear). Fig.1-2 shows the sampled impulse response of a representative communications channel and some associated terminology. Random fluctuations in the channel, modelled as additive noise, also corrupt the signal. The received signal (z in Fig.1-1) is demodulated (with correct carrier phase and frequency, *i.e.*, coherently), filtered and then sampled (with correct timing phase) at the symbol rate to produce the signal y_k, which, for equalization purposes, we will refer to as the received signal. This conversion from continuous to discrete time loses no information as long as we use a *matched filter* [3] before the sampler. The equalizer estimates the data sequence from the received signals and these estimates $\hat{u}_k$ are called *decisions*. The process resulting in y_k can equivalently and more conveniently be represented in the baseband equivalent form shown in Fig.1-3.

The combined effects of filtering and sampling in Fig.1-1 have been lumped into

one filter which we may view as the effective channel. The received signal can be expressed as the convolution of the channel impulse response (parametrised by h_i, $i = 0, \ldots, L$) and the input symbol sequence, with additive noise n_k *viz:*

$$\begin{aligned} y_k &= \sum_{i=0}^{L} h_i u_{k-i} + n_k \\ &= h_0 u_k + \sum_{i=1}^{L} h_i u_{k-i} + n_k, \end{aligned} \tag{1.1.1}$$

in which the channel order L may be infinite. We refer to the coefficient h_0 as the *cursor*, which, without loss of generality, we can (and usually do) take to be unity. In writing (1.1.1) in this form, we are assuming implicitly that there is no transmission delay (since y_k depends on u_k), and, in treating h_0 as the cursor (which is generally the dominant coefficient), we are assuming that some linear filtering in the receiver has cancelled the precursor ISI. The middle term in the last equation therefore represents the remaining (postcursor) ISI. The vector of the L most recent past channel inputs

$$x_k \triangleq [u_{k-1}, \cdots, u_{k-L}]' \tag{1.1.2}$$

is called the *state*. (We can also define a channel state in the IIR case.) This idealized model is a commonly adopted starting point [6] for problems in channel equalization of uncoded pulse amplitude modulated data.

Further assumptions concern the statistics of the input and noise sequences (*e.g.*, correlated or independent) and the type of channel (*e.g.*, linear or non-linear). In the following chapters we will mostly consider the case in which:

1. The input to the channel is a sequence of independent and identically distributed (*iid*) multi-level random variables.

2. The channel is a linear finite impulse response filter.

3. The noise is a sequence of independent zero-mean Gaussian random variables.

The independence of the data teamed with assumption 2 guarantees that the baseband system, from the point of view of the receiver, can be modelled as a finite-state machine with noisy observations, or a finite-state Markov process [7, 8]. That is, we

can characterise the system in terms of its initial state and the transition probabilities between its various states.

For simplicity of presentation, we will mainly be concerned with binary signalling, although extensions to M-ary signalling and quadrature amplitude modulation will be covered in chapter 3. There, we will also deal with the equalization of infinite impulse response and non-linear channels. We do not consider how to modify the equalizer structures we develop for coded data.

1.1.3 Equalizer Design and Analysis

The equalization problem now reduces to the design of a system that reliably recovers the data $\{u_k\}$ from the received signals $\{y_k\}$. In practice, the following points need to be considered:

1. Lack of knowledge and time variation of the channel parameters.
2. Computational complexity and decoding delay of the equalizer.
3. The signal-to-noise ratio (SNR).
4. The required bit error rate (BER).

We make some general remarks concerning the above aspects. The channel is in general unknown by the receiver and may change with time due to fading (varying signal strength). It is therefore desirable for the equalizer to adapt its internal model of the channel to *track* the unknown physical channel and cope with any (slow) time variation. An equalizer which adjusts its parameters automatically is called an *adaptive equalizer*. Adaptive equalization divides into two distinct approaches. In the first, an initial *training sequence* generated at the transmitter and known to the receiver is used to to perform the adjustment. The second, called ***blind adaptation***, uses only signals available at the receiver. Blind adaptation is preferred in multi-receiver systems where it is impractical to retransmit training sequences. Blind adaptive schemes often adjust the equalizer taps via a stochastic gradient descent algorithm, derived from constraints imposed by the signal alphabet (modulus restoral [9], for instance). The signal-to-noise ratio is determined by the transmitter power (subject to fading and other disturbances). The processing delay of the equalizer affects the

maximum bit rate that can be transmitted. Broadly speaking, the lower the required bit error rate, the higher the complexity/delay of the equalizer.

The bulk of this thesis focuses on *non-adaptive* aspects of equalizer systems (with the exception of parts of chapters 2 and 3). Although, in practice, the adaptation of an equalizer is of crucial importance to its operation, the importance of understanding the underlying mechanisms which cause errors (incorrect decisions) in the tuned (correctly adapted) device cannot be overemphasized. Our main concern is the non-adaptive performance of non-linear equalizers and we give only a brief account of adaptation in the conventional equalizers which we review in the following sections (the reader is referred to [6] for a comprehensive coverage).

The term *non-linear equalizer* is understood to mean a system for the recovery of transmitted data whose operation, in the non-adaptive mode, cannot be represented by a linear filter. The non-adaptive performance of a non-linear equalizer, as measured by its *bit error rate* (BER), is related to the criterion (subject to practical constraints) used in its design. Some examples of design criteria are: maximum likelihood sequence estimation, minimum bit error rate detection, minimum mean square error and zero-forcing criteria. Two major practical constraints in the realisation of an equalizer are the computational complexity and the inherent delay in obtaining data estimates.

In most (but not all) non-linear equalizers, there is some kind of feedback of past decisions. The mechanism for this may be either direct, as in a decision feedback equalizer (DFE) [5, 6], or indirect, as in reduced-state sequence estimators (RSSEs) [10, 11, 12]. The presence of a feedback mechanism complicates the performance analysis. The problem of computing the output error probability, $Pr(\widehat{u}_k \neq u_k)$, usually a relatively straightforward calculation for a linear equalizer or feedforward equalizer, is made more arduous by the dependence of present outputs on past outputs via feedback (recursion). Nonetheless, we can distinguish two partial solutions to the problem of non-linear equalizer performance analysis:

1. The conventional (and usually more tractable) analysis of primary (noise-induced) errors in the system, assuming correct past decisions.

2. The analysis of errors produced by initial error states and propagated subsequently, in the absence of noise.

A complete understanding of error performance requires both of the above analyses. Many authors consider analysis 1 mandatory but 2 is often left out (for examples of 1, see [13, 14]). When there is feedback or use of past decisions, initial errors can produce further errors, enhancing the bit error rate due to noise alone, so that analysis 2 becomes a study of *error propagation*. One of the themes in our work is a study of error propagation in a generalised decision feedback equalizer called a *block decision feedback equalizer* [15]. The block DFE (in certain cases) is amenable to analyses of the kind applied to the conventional DFE [16, 17, 18]. The theory of finite-state Markov processs is an invaluable tool for modelling such systems.

1.2 Established Techniques

With the intention of setting the scene for the new techniques that we have developed, we now review three basic equalization strategies in increasing order of complexity and performance. These are the linear equalizer (LE), the decision feedback equalizer (DFE) and the maximum likelihood sequence estimator (MLSE) [19]. We also examine the reduced-state sequence estimator (RSSE) [11] which has close ties with the MLSE and the DFE. An understanding of the workings of these systems will be important in what follows. Certain detailed aspects of MLSEs and RSSEs have been relegated to appendix A.

1.2.1 Linear Equalization

The adaptive linear transversal equalizer (Fig.1-4) consists of a tapped delay line whose taps are adjusted to give a response approximating the *inverse* of the sampled channel transfer function. (Of course, disregarding noise, the exact equalization of a FIR channel would generally require a LE of infinite length.) In this configuration, the LE acts to minimise intersymbol interference, possibly *enhancing* the noise, and is called a zero-forcing equalizer (ZFE). The LE can also be adapted to minimise an ISI plus noise criterion yielding a linear mean-square equalizer (LMSE). The

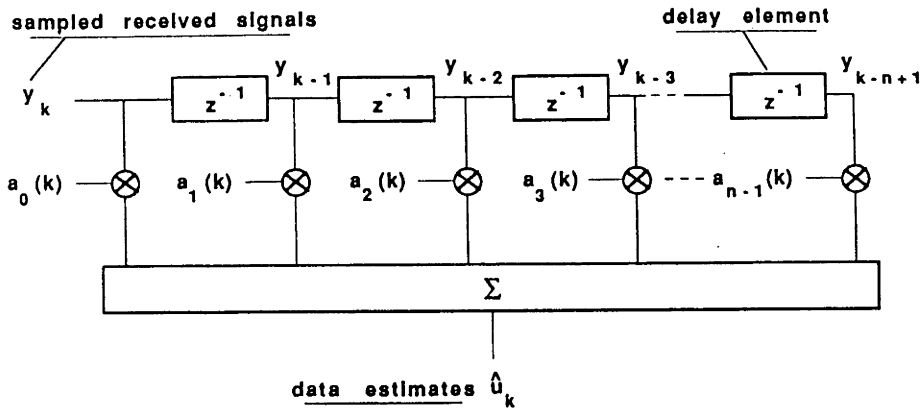

Figure 1-4: Adaptive linear transversal equalizer.

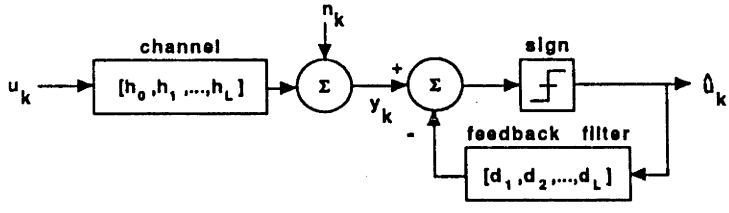

Figure 1-5: Non-adaptive decision feedback equalizer.

ZFE is typically less robust than the LMSE and can only be used on eye-open[1] ISI channels [20]. A decision device—a hard limiter in the case of binary signals, or vector quantizer for M-ary signalling—can be added at the output of the linear equalizer to improve its noise immunity. This configuration is sometimes called a decision directed equalizer (DDE), although this terminology is more often reserved for the description of equalizer adaptation. Blind adaptation of DDE's has been studied in [21], and [6] provides a general reference on training sequence adaptation of linear equalizers.

1.2.2 Decision Feedback Equalization

For more severe channel distortion (*e.g.*, non-minimum phase channels whose inverse is unstable) a decision feedback equalizer (Fig.1-5) may be required. The DFE is a non-linear equalizer with a linear feedforward filter designed to handle *precursor*

[1] Channels whose sampled output produces an open eye diagram.

ISI, and linear feedback filter and non-linear decision device handling *postcursor* ISI and noise. For argument's sake, we assume the feedforward part is lumped with the channel response and consider only the operation of the feedback part (treating the dominant impulse response parameter h_0 as the *cursor*). Ideally the DFE adapts or tunes its feedback taps d_i to cancel the intersymbol interference ($d_i = h_i$, $\widehat{u}_{k-i} = u_{k-i}$, $i = 1, \ldots, L$) and passes the ISI-free signal ($h_0 u_k + n_k$) to a quantizer (slicer) which makes a decision on the transmitted signal. In symbols this reads

$$\widehat{u}_k = \operatorname{sgn}(\underbrace{h_0 u_k + \sum_{i=1}^{L} h_i u_{k-i}}_{y_k} - \sum_{i=1}^{L} d_i \widehat{u}_{k-i} + n_k). \tag{1.2.1}$$

The decision feedback equalizer does not utilise the energy contribution from the intersymbol interference in decoding each symbol, it merely tries to cancel it. If the DFE makes an initial incorrect decision $\widehat{u}_k \neq u_k$ (be it due to noise, incorrect initialization or mistuning) it may fail to cancel the ISI and this occasionally leads to bursts of errors. The phenomenon of an initial error causing subsequent errors is known as *error propagation*. A more detailed description of the operation of the decision feedback equalizer, covering training sequence adaptation, is set out in [22]. Error propagation in DFEs has been analysed in [23, 24] and blind adaptation in [25].

1.2.3 Maximum Likelihood Sequence Estimation

The third technique—maximum likelihood sequence estimation (MLSE)—is perceived as the *optimal* system for equalization. MLSEs can give performance approaching that of ISI-free transmission [19], and substantial bit error rate improvements over the DFE are obtainable (*e.g.*, 6dB for simulations on a second order partial response channel). We discuss the technique in some detail as it relates to work described in chapter 3. The MLSE is a *sequence estimator*, as opposed to the LE and DFE which output one decision per sample and can be seen as symbol-by-symbol detectors. Other optimal non-linear detectors have been devised [26, 27, 28, 29] which maximise various probability criteria, but these do not seem to have been adopted as the basis for practical systems. A summary of some of these

techniques is given in section 3.4.3.

Using the notation introduced in section 1.1.2, the MLSE operates on finite impulse response channels and determines the length K state sequence $\{x_k\}$ (or equivalently the input sequence $\{u_k\}$) which maximises the *conditional probability density* $p(\{y_k\}|\{u_k\})$. A closely related problem, that of maximum *a posteriori* (MAP) estimation, seeks to maximise the *a posteriori* probability $Pr(\{u_k\}|\{y_k\})$. The two criteria are equivalent when the input sequence is *iid* and equiprobable [26, 15] and can be interpreted as finding the input sequence which best represents the measurement sequence in a mean square sense. The MAP criterion minimises the probability of incorrectly decoding the whole state sequence. In chapters 3 and 5, we consider non-linear equalizers based on fixed-delay MAP criteria incorporating decision feedback.

For binary inputs to a finite impulse response channel of order L there are 2^L possible states (since each component is binary and the state consists of L of these). The MLSE searches over all admissible state sequences and selects the one which minimises a sum-of-squares cost function—arising from the assumption of white Gaussian noise. The delay involved in this brute force computation would be unacceptable for long channels and many authors have considered various related criteria which result in simpler (suboptimal) non-linear detectors [27, 30, 31]. However, the landmark work on the MLSE problem was achieved by Forney [19] with an application of the *Viterbi algorithm* (VA) [32]—a forward-time version of dynamic programming [33]. This recursive solution of the optimization problem, *Viterbi decoding*, makes the MLSE computationally practicable. However, since its complexity grows exponentially with the channel length, the MLSE can only be used on channels having a short impulse response. Appendix A.1 contains a summary of the Viterbi algorithm.

1.2.4 Review of Other Techniques

As pointed out in section 1.2.3, there exists a large discrepancy in performance and in complexity between the decision feedback equalizer and the maximum likelihood sequence estimator. Seeking to bridge this gap, many attempts have been made to design hybrid equalizers using the three basic equalizers as components

[12, 34, 35, 36, 37, 38, 39]. These composite systems try to take advantage of the simplicity of the LE and/or DFE to preprocess and remove a portion of the inter-symbol interference from the received signal, thus presenting an effectively shorter channel and reducing the complexity of the Viterbi decoder required for the remaining task. Although these hybrid equalizers can work well, their success is limited in general by the performance of the LE or DFE they incorporate.

Since the introduction of the Viterbi decoder, an equalizer with performance akin to a MLSE, but needing substantially less computation was the object of intensive research. By the mid 1980's, systematic attempts were being made at developing such equalizers, or moreover classes of equalizers, with the attributes of relative simplicity and performance ranging between the extremes of the DFE and MLSE. We now mention some of these. One technique is based on reduced complexity Viterbi decoding teamed with internal decision feedback. This was introduced independently in [10], under the name *delayed decision feedback sequence estimation* (DFSE) and by [11], under the name *reduced-state sequence estimation* (RSSE), who built upon the preceding work of [10]. Another technique, called the M-algorithm, simply truncates the search used in the Viterbi decoder. A large part of our work relates to the development and evaluation of a device (the *block decision feedback equalizer*) which also satisfies these requirements. We defer its discussion until chapter 3. A description of delayed decision feedback sequence estimation and reduced-state sequence estimation is available in appendix A.2. Both schemes reduce the dimension of the Viterbi trellis in a natural, structured way, retaining the essential features of the Viterbi decoder. In [10] the channel may be recursive (IIR), whereas in [11] it is assumed to have finite impulse response.

1.3 Outline of Thesis

1.3.1 Summary and Contributions

We now proceed with a chapter-by-chapter description of the thesis, pointing out the major contributions.

In chapter 2 we study a non-linear feedforward processor with hard limiting nodes, akin to an multi-layer perceptron neural network [40] (see Fig.2-4), that

emulates a decision feedback equalizer. The so-called feedforward emulator (FFE) arises through a recursive *unwrapping* of the DFE, followed by truncation (cutting off the feedback path). This unwrapping procedure is analogous to a Markov expansion [41] of a linear (finite dimensional) IIR system. The DFE with feedback is replaced by a multilayer feedforward processor and generates data estimates with delay corresponding to the number of layers.

In section 2.3 we obtain an upper bound on the noiseless error probability (due to non-zero initial conditions) by generalising the existing theory for the DFE [23]. The method entails modelling exactly the feedforward emulator as a DFE that has been initialized at each time instant in a *non-standard* error state. Then, modelling the DFE by a finite-state Markov process with a large number of states, aggregation is performed by choosing a worst case channel (specialised to the FFE case) and an exponential upper bound for the FFE's error probability is obtained in terms of the number of *layers*. The bound is realized by worst case channels but seems to be conservative for most practical (decaying) channels. The importance of this work is that it brings hard analysis to bear on the non-adaptive performance of a neural network-like structure which may have more general application. The norm in most work on neural networks is the recognition that a multi-layer perceptron neural network can perform a certain task adequately (*i.e.*, non-linear mapping), but the justification often rests solely on the experimental or simulated performance.

The structure of the feedforward emulator is constrained by the requirement that it should act as an equalizer. This manifests itself in the number of nodes per layer, the connectivity between nodes and the *interdependence* of the weights. It turns out that, for a FFE with Λ layers, only $\Lambda - 1$ of the $\frac{1}{2}\Lambda(\Lambda + 1)$ weights are independent (as the matrix of weights is Toeplitz). Thus only $\Lambda - 1$ quantities need be adapted during training (or tracking). Using *back propagation* ideas [40, 42], recursive gradient descent algorithms have been developed and tested for a FFE consisting of (1) sign nodes and (2) sigmoid nodes.

In chapter 3 we develop a family of equalizers, called block decision feedback equalizers, whose performance/complexity tradeoff can be varied *parametrically* to bridge the gap separating the DFE and the maximum *a posteriori* (MAP) detector (or equivalently, in the case of *iid* equiprobable symbols, the maximum likelihood se-

quence estimator). In some respects our contribution, the block decision feedback equalizer, is complementary to reduced-state sequence estimation (see appendix A.2). Whereas the latter approach is based on reducing the complexity of an MLSE using the idea of a subset state, our approach seeks to improve the performance of the DFE by generalizing it to the vector case. In a manner of speaking, the RSSE is a top-down approach (the MLSE being perceived as the "top") and the block DFE a bottom-up approach. Precursors to the block DFE were studied in [13, 14, 43, 44], although the block DFE was developed independently.

The block DFE (Fig.3-1) is a natural generalization of the conventional DFE. It is based on a *block processing* [45, 46, 47] channel model connected in feedback with a *vector quantizer* [15] operating under a maximum *a posteriori* criterion. The block DFE, or (p, q)-DFE, is indexed by two parameters: the block length p and the number of decisions q produced at each (block) iteration. The block length is independent of the channel length. It can be made to replicate the DFE when $p = q = 1$, the MLSE in the limit as $p = q \rightarrow \infty$ and the maximum *a posteriori* symbol-by-symbol detector [30, 31] when $q = 1$, $p \rightarrow \infty$. The best performance (for fixed p) is achieved in the latter mode with $q = 1$, p large, where the $(p, 1)$-DFE functions as a minimum bit error rate detector. We investigate these connections in section 3.3.5

Intuitively, the block DFE acts as a non-linear fixed-lag smoother [48] with lag p. A block of channel outputs $y_k, \ldots, y_{k+p-1}$ is used to estimate a block of channel inputs $u_k, \ldots, u_{k+q-1}$. The computational load increases exponentially with p, but fortunately even small values of p result in greatly improved performance over the DFE. Despite the fact that the $(p, 1)$-DFE has superior performance to the (p, p)-DFE, the latter device may still find useful application in the equalization of block codes in which each codeword has a fixed length [49]. The decision criterion for the block DFE reduces to a minimum Euclidean distance calculation for high signal-to-noise ratios [15]. The use of a minimum distance metric in a block DFE gives rise to a device called a *high SNR block DFE*, which has similar performance, but is computationally much simpler. These results, presented in section 3.2, are of practical importance for the implementation of block decision feedback equalizers. Simulated performance and complexity issues are investigated in section 3.3. Exam-

ples of two and three-dimensional block DFE realisations and their vector quantizers are also given. Some other interesting examples of two-dimensional decision devices (and their corresponding decision boundaries) which optimize various criteria (including maximum likelihood decision boundaries for non-linear FIR channels) may be found in [50, 51, 52]. The block DFE can operate on linear infinite impulse response (ARMA) channels and also non-linear channels having a finite-dimensional state space realization. Generalization to M-ary signalling, coloured noise and adaptation of block DFE parameters are considered in section 3.5.

Block DFE performance analysis forms the substance of chapter 4. All analyses of the block decision feedback equalizer so far concentrate on the 2-input ($p = 2$) case. The analysis is complicated by the non-linearity of the decision device and its dependence on the channel parameters. The *decision boundary* is the curve separating the different decision regions in the decision device. This boundary is curved for a (2,1)-DFE but becomes piecewise linear for high signal-to-noise ratios. (The (2,2)-DFE decision boundary is always piecewise linear.) Simple geometrical considerations yield an explicit formulae for the decision boundary, and are the starting point of performance/stability analyses of the (2,1)-DFE and (2,2)-DFE.

We give a representative example calculation in section 4.3 of the (2,1)-DFE's primary error probability on a first order channel, *i.e.*, the bit error rate assuming that there have been no past decision errors. In section 4.4 we derive *eye conditions* for finite error recovery of the (2,1)-DFE on an arbitrary second order channel.

The block DFE is a non-linear equalizer with feedback and therefore can suffer from error propagation. In particular, we are interested in describing the class of channels on which the block decision feedback equalizer has a finite error recovery time (in the sense that it can recover from an initial error state in a finite time regardless of the input sequence). On such channels we say the block DFE is *stable* [16]. Of course, other statistical measures of the error recovery time could be used to define stability, but the present definition is more convenient for our purposes. The theory of finite-state Markov processs [7, 8] gives *necessary and sufficient conditions* defining the class of stable channels, using the idea of *pathological* input sequences [16, 23]. This classification of second order channels has been carried out for the (2,1)-DFE and the (2,2)-DFE in the noiseless case [53] and is presented in section

4.5. Example calculations of the mean and variance of error recovery times are also presented. These preliminary results show that the block decision feedback equalizer is stable on a broader class of channels and is therefore more *robust* than the decision feedback equalizer.

As is common in decision feedback equalization, the block DFE's criterion uses the assumption of correct block ISI cancellation (or channel state estimation) in the design phase. In chapter 5 we consider a different criterion, related to fixed-delay, symbol-by-symbol MAP detection, which does not rely on this assumption and subsequently generalises the block DFE, although it only operates on finite impulse response channels. The resulting non-linear equalizer is called a maximum *a posteriori* decision feedback detector [54], and incorporates knowledge of certain error probabilities, giving improved performance. The design of the detector is covered in chapter 5 in which we also present a recursive procedure for its realization (this is necessary since the decision criterion cannot be expressed in closed form). We provide simulated performance comparisons for the new detector on first and second order channels, showing the improvement in bit error rate over the block DFE, and give examples of the decision regions that are thus formed.

Chapter 6 concludes the thesis and contains a discussion of topics for future research. In particular, we mention the need for further work regarding simplified decision devices for, and the incorporation of coding into block decision feedback equalizers.

Chapter 2

Feedforward Emulation of the Decision Feedback Equalizer

2.1 Introduction and Motivation

The decision feedback equalizer is a simple but effective non-linear equalizer that has enjoyed widespread application in digital communication systems [6]. Its operation has been studied by various authors [16, 17, 36, 55] and is reasonably well understood. The simplest realisation of a DFE is the non-linear recursive structure shown in Fig.1-5. On the other hand, the multi-layer perceptron (MLP) neural network [40], which also has been applied to the equalization problem in [52, 56], is a relatively poorly understood system. In this chapter, we consider an intermediate structure for equalization, the *feedforward emulator* (FFE) [57, 58, 59], which derives from the DFE and is closely related to the feedforward neural network, and which may be analysed in much the same way as the DFE.

The feedforward emulator has a structure akin to a systolic array for parallel processing. In the absence of noise, the the emulator can be represented as a standard MLP neural network [59] with hard-limiting nodes, but it has a non-standard structure in general. Unlike MLP neural networks whose weights may be chosen freely, the parameters here are constrained so that only some may vary independently. If we can identify the communication channel, then we know how to assign the weights of the FFE system to make it act as an equalizer. This is an enhance-

ment of standard MLP neural network equalizer techniques because there is a direct link between the weights and the parameters characterising the channel.

The chapter is organised as follows. In section 2.2, we briefly review the non-adaptive decision feedback equalizer and detail an unwrapping procedure followed by truncation that results in a recursive multilayer processor with hard limiting nodes. We obtain the feedforward emulator by disconnecting the feedback of decisions made in the distant past. We give some low order illustrations and show how the structure generalises to an arbitrary number of layers.

In section 2.3 we re-introduce the finite state Markov process description of the tuned noiseless decision feedback equalizer found in [17]. We extend the model to embrace both the DFE and the feedforward emulator by enlarging the state space. In section 2.3.2 we introduce the idea of a worst case channel, *i.e.*, a FIR channel guaranteeing the worst bit error rate performance for any channel of the same order. We subsequently apply FSMP theory to upper bound the noiseless error probability of the DFE (in section 2.3.3) and then the FFE (in section 2.3.4), obtaining a bound in terms of the number of layers for the latter. We present numerical examples for the non-adaptive feedforward emulator in section 2.4 and also examine conditions on the channel under which the representation is exact (in the sense of producing the same sequence of outputs in the absence of noise).

Having established the possibility of representing a DFE by a specific feedforward emulator, we consider, in section 2.5, algorithms for FFE adaptation using a training sequence—as in supervised learning of neural networks. Because of the FFE's non-standard structure, the usual back propagation learning rule [40] is not applicable. Instead, we develop two novel training algorithms. The first of these assumes the hard limiters are replaced by sigmoid processing elements and implements a gradient descent strategy subject to the constraints imposed on the weights. The second algorithm, applicable to an FFE with *sign* nodes, uses a different error measure from the more usual sum-of-squares output error. Numerical evaluation of these training algorithms are presented, and comparisons with the DFE are made. Section 2.6 contains concluding remarks and discussion of some open problems.

2.2 Unwrapping the Decision Feedback Equalizer

Conventional multi-layer perceptron neural networks, which can be configured to act as non-linear equalizers [52], are highly interconnected non-linear systems whose analysis is typically very difficult. In contrast to this, we derive a new non-linear feed-forward processor designed to emulate the well-studied decision feedback equalizer. The new structure is more amenable to analysis and it is possible to obtain bounds on its performance in the absence of noise—which we assume for simplicity only. We therefore consider a non-adaptive binary decision feedback equalizer operating on a finite impulse response channel corrupted by additive zero-mean white Gaussian noise n_k. At the output of the channel, the sampled received signal is

$$y_k = \sum_{i=0}^{L} h_i u_{k-i} + n_k, \tag{2.2.1}$$

where $\{h_i\}$ are the impulse response coefficients and $\{u_k\}$ is a sequence of equiprobable *iid* binary inputs which we cannot measure directly. The DFE (Fig.1-5) generates an estimate of the input signal, based on its own past decisions, given by[1]

$$\widehat{u}_k = \mathrm{sgn}(y_k - \sum_{j=1}^{L} d_j(k)\widehat{u}_{k-j}) \stackrel{\Delta}{=} f_1^L(y_k; \widehat{u}_{k-1}, \ldots, \widehat{u}_{k-L}). \tag{2.2.2}$$

The feedback tap gains $d_j(k)$ are adapted to cancel the intersymbol interference introduced by the channel. Initially, we will be presenting an analysis of the non-adaptive system in which the $d_j(k) = d_j$ are constant. We assume, with no loss of generality, that $h_0 = 1$.

We develop a self-similar recursive representation of (2.2.2) by "unwrapping" the DFE, and, in so doing, introducing a delay in the computation of the decisions. As we mentioned in chapter 1, this procedure is analogous to the Markov expansion of an ARMA filter. At the first step we write

$$\widehat{u}_{k-1} = \mathrm{sgn}(y_{k-1} - \sum_{j=1}^{L} d_j \widehat{u}_{k-j-1}),$$

[1] $\mathrm{sgn}(x) = 1$ if $x \geq 0$ and -1 if $x < 0$

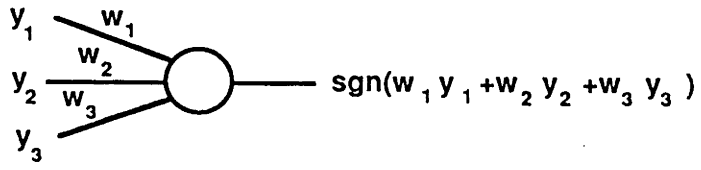

Figure 2-1: Processing element or node.

and substitute for $\hat{u}_{k-1}$ in (2.2.2), thus obtaining

$$\begin{aligned}\hat{u}_k &= \operatorname{sgn}(y_k - d_1 \operatorname{sgn}(y_{k-1} - \sum_{j=1}^{L} d_j \hat{u}_{k-j-1}) - \sum_{j=2}^{L} d_j \hat{u}_{k-j}) \\ &\triangleq f_2^L(y_k, y_{k-1}; \hat{u}_{k-2}, \ldots, \hat{u}_{k-L-1}). \end{aligned} \tag{2.2.3}$$

At the next step, we eliminate $\hat{u}_{k-2}$ (which appears twice in (2.2.3)) to get

$$\begin{aligned}\hat{u}_k &= \operatorname{sgn}\{y_k - d_1 \operatorname{sgn}(y_{k-1} - d_1 \operatorname{sgn}(y_{k-2} - \sum_{j=1}^{L} d_j \hat{u}_{k-j-2}) - \sum_{j=2}^{L} d_j \hat{u}_{k-j-1}) \\ &\qquad - d_2 \operatorname{sgn}(y_{k-2} - \sum_{j=1}^{L} d_j \hat{u}_{k-j-2}) - \sum_{j=3}^{L} d_j \hat{u}_{k-j}\} \\ &\triangleq f_3^L(y_k, y_{k-1}, y_{k-2}; \hat{u}_{k-3}, \ldots, \hat{u}_{k-L-2}). \end{aligned} \tag{2.2.4}$$

After $\Lambda - 1$ such steps we obtain a highly nested composition of *sign* functions whose functional form can be written as

$$\hat{u}_k = f_\Lambda^L(y_k, y_{k-1}, \ldots, y_{k-\Lambda+1}; \hat{u}_{k-\Lambda}, \ldots, \hat{u}_{k-L-\Lambda+1}). \tag{2.2.5}$$

There are in fact Λ degrees of nesting in this expression and we can interpret these as the layers in a recursive multilayer processor whose external inputs are the $\{y_{k-\Lambda+1}, \ldots, y_k\}$, whose feedback inputs are the $\{\hat{u}_{k-\Lambda}, \ldots, \hat{u}_{k-\Lambda-L+1}\}$, and whose output is $\hat{u}_k$. The processing elements, or nodes, compute the sign of the weighted sum of their inputs. Fig.2-1 depicts a typical node.

Intuitively, it is reasonable to expect that the dependence of the output $\hat{u}_k$ on the past decisions $\hat{u}_{k-\Lambda+1-j}$ decreases the larger we make Λ (at least for $\Lambda > L$).

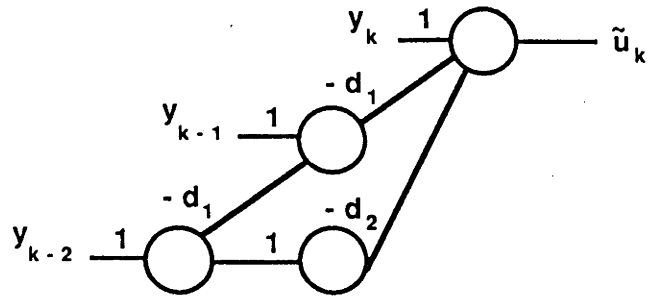

Figure 2-2: Three-layer feedforward processor.

This is equivalent to the effect of earlier decisions in a DFE ceasing to influence later decisions, given a large enough time delay between the two. We give substance to this notion in the following sections.

Supposing, then, that we can ignore decisions made in the distant past, we disconnect the feedback implied in (2.2.5) by setting the arguments involving past decisions to zero, obtaining

$$\tilde{u}_k = f_\Lambda^L(y_k, y_{k-1}, \ldots, y_{k-\Lambda+1}; 0, \ldots, 0). \tag{2.2.6}$$

If our intuition is correct, then with high probability and given a large number of layers (Λ large) we would have $\hat{u}_k = \tilde{u}_k$. It is instructive to visualise (2.2.6) as a multilayer *feedforward* processor with *sign* nodes. We illustrate in Fig.2-2 the corresponding system in the $\Lambda = 3$ case first (setting $\hat{u}_{k-j} = 0$, $j \geq 3$ in (2.2.4)).

Before proceeding with the general case embodied by (2.2.6), we make a short digression to examine the noiseless feedforward version of (2.2.4) with $\Lambda = 3$, which may be expressed as

$$\begin{aligned}
\tilde{u}_k &= f_3^L(y_k, y_{k-1}, y_{k-2}; 0, \ldots, 0)\,|_{n_k = n_{k-1} = n_{k-2} = 0} \\
&= \operatorname{sgn}\{u_k + \sum_{i=1}^{L} h_i u_{k-i} - d_2 \operatorname{sgn}(u_{k-2} + \sum_{i=1}^{L} h_i u_{k-i-2}) \\
&\quad -d_1 \operatorname{sgn}(u_{k-1} + \sum_{i=1}^{L} h_i u_{k-i-1} - d_1 \operatorname{sgn}(u_{k-2} + \sum_{i=1}^{L} h_i u_{k-i-2}))\}.
\end{aligned}$$

This is depicted in Fig.2-3 for a channel length $L = 2$ and arbitrary h_0. Of course,

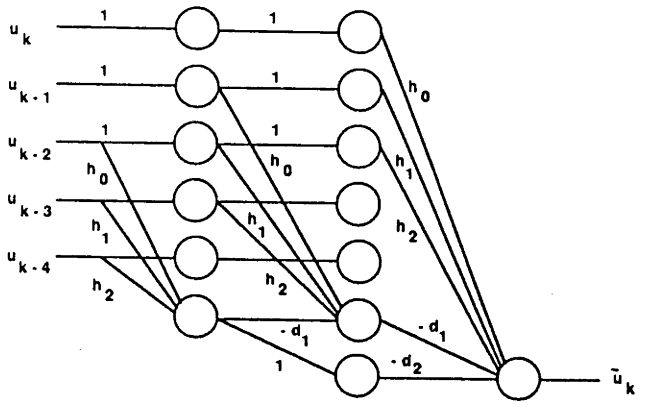

Figure 2-3: Noiseless MLP realisation.

in practice, the input sequence $\{u_k\}$ and the channel parameters are unknown, but the purpose of Fig.2-3 is to show how Fig.2-2 can be captured in a standard MLP neural network framework.

Returning to the general Λ-layer structure described before in (2.2.6), it is fairly easy to generalise the low order cases to arrive at the diagram in Fig.2-4. This feedforward parallel processing structure, which implements (2.2.6), will be referred to in the sequel as a Λ-layer FFE. We have drawn Fig.2-4 to accentuate its Toeplitz structure—the weight of the branch connecting node i in layer k to node $i+j$ in layer $k+1$ is $-d_j$, independent of i. We alert the reader to the following important differences between Fig.2-4 and a standard MLP neural network.

- Each diagonal node has one external input, this being a noisy channel output.
- All horizontal connections have fixed weight one.
- There are only $\Lambda - 1$ distinct weights.

The FFE is clearly self-similar. The Λ-layer FFE is embedded in the $(\Lambda-1)$-layer FFE. This embedding property has interesting consequences in terms of monotonicity of performance (see simulations in section 2.4) and the simplicity of the training rule required for adaptation (section 2.5).

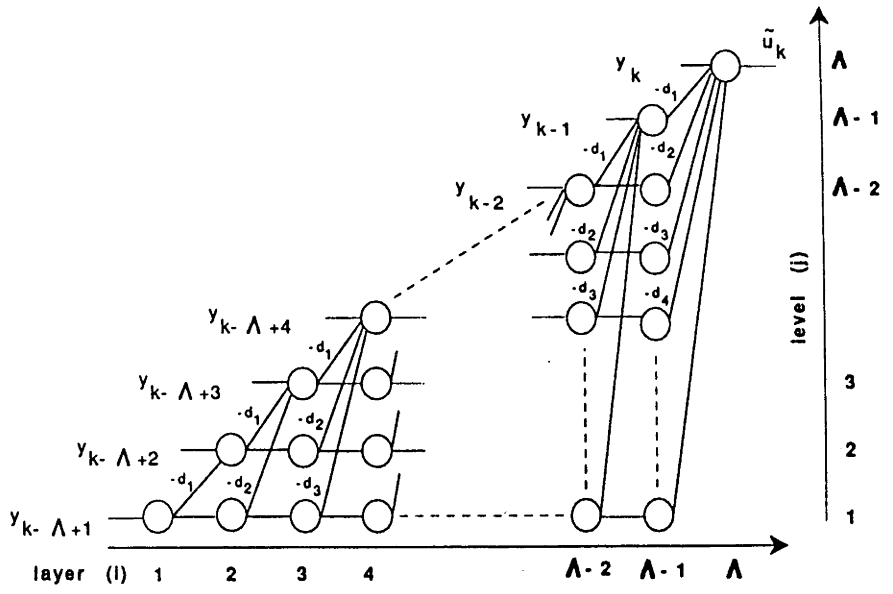

Figure 2-4: Feedforward emulator for the DFE.
Each node computes the sign of the weighted sum of its inputs. The weights are marked on the branches and horizontal connections have weight 1.

2.3 Analysis of Noiseless Error Probability

2.3.1 Finite State Markov Process Description

We can accurately model the stochastic dynamics of the DFE using the theory of finite state Markov processes [8] as long as the input sequence to the channel is independent. Referring to (2.2.1) and (2.2.2), the input is u_k and we choose

$$X_k = [u_{k-L}, \ldots, u_{k-1}, \hat{u}_{k-L}, \ldots, \hat{u}_{k-1}]'$$

as the state vector. There are 4^L states in total if all elements are binary. Since the DFE is assumed to be tuned (in the sense that $d_j = h_j$, $j \in \{1, \ldots, L\}$), we can reduce the number of states to 3^L by defining an error state

$$E_k \triangleq [e_{k-L}, \ldots, e_{k-1}]' \tag{2.3.1}$$

where each component $e_{k-j} = u_{k-j} - \hat{u}_{k-j}$ takes values in the set $\mathbb{E} \triangleq \{-2, 0, +2\}$. We denote by $\mathbb{E}^L$ the set of all possible E_k states. Consequently, the DFE output

(2.2.2) (with $h_0 = 1$) can be re-expressed as

$$\widehat{u}_k = \mathrm{sgn}(u_k + \sum_{i=1}^{L} h_i e_{k-i} + n_k). \tag{2.3.2}$$

In order to simplify the analysis, as is standard in the analysis of error propagation of DFEs [16, 23], we consider only the noise-free case ($n_k = 0\ \forall k$) so that the unique absorbing state of the FSMP is $E_k = \underline{0}$ (the zero vector). To see this, note that the error vector or state satisfies the simple recursion

$$E_{k+1} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} E_k + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ u_k - \mathrm{sgn}(u_k + \underline{h}' E_k) \end{bmatrix}, \tag{2.3.3}$$

where $\underline{h} \triangleq [h_1, \ldots, h_L]'$. If $E_k = \underline{0}$ then for all inputs we have $\mathrm{sgn}(u_k + \underline{h}' E_k) = u_k$ and the DFE remains in the zero-error state regardless of the input. However, from an E_k state having a non-zero entry, there is a non-zero probability of reaching the absorbing state in M steps for any $M \geq L$ (since a sequence of L correct decisions will cause transition to the absorbing state). Also, the probability of ultimately reaching the absorbing state is 1 [16]. When noise is present, only a noise-induced decision error can cause a transition from the zero-error state.

Returning to the recursive representation of the DFE described in the last section, we see that the output, in the absence of noise, can be viewed as depending on the sequence of inputs $u_{k-\Lambda+1}, \ldots, u_k$ and the initial state $X_{k-\Lambda+1}$ (or $E_{k-\Lambda+1}$). For convenience, we introduce the notation (with reference to (2.2.5))

$$\begin{aligned} & f_\Lambda^L(y_k, \ldots, y_{k-\Lambda+1}; \widehat{u}_{k-\Lambda}, \ldots, \widehat{u}_{k-\Lambda-L-1})|_{n_k=\cdots=n_{k-\Lambda+1}=0} \\ & \quad \triangleq \ g_\Lambda^L(u_k, \ldots, u_{k-\Lambda-L-1}; \widehat{u}_{k-\Lambda}, \ldots, \widehat{u}_{k-\Lambda-L-1}). \end{aligned} \tag{2.3.4}$$

We aim to determine an upper bound on the probability of an incorrect decision at time k from an arbitrary non-zero initial error state. This measure of performance will be seen to be central to the analysis of the feedforward emulator to follow.

With this in mind, we define an extended state $\overline{X}_k$ ($\overline{E}_k$ in the tuned case) having the same form as X_k but in which the decisions which appear in the initial state may take the additional value zero. Each element $\overline{E}_k$ of the extended error state $\overline{E}_k$ takes values in $\overline{I\!E} \triangleq \{0, \pm 1, \pm 2\}$, so there are a total of 5^L $\overline{E}_k$ states comprising the set $\overline{I\!E}^L$. Of course, a DFE with an initial state in $\overline{I\!E}^L$ reverts to a DFE with state in $I\!E^L$ after L time units because the binary decisions that are fed back will displace the initial conditions. The concept of an extended state, together with the error probability bound, will allow us to gauge the effect of omitting the recursive part (*i.e.*, old decisions) in the representation for the DFE (2.2.5), thus obtaining a *feedforward* structure generated by

$$\widetilde{u}_k = g_\Lambda^L(u_k, \ldots, u_{k-\Lambda+1}, u_{k-\Lambda}, \ldots, u_{k-\Lambda-L-1}; \widetilde{u}_{k-\Lambda} := 0, \ldots, \widetilde{u}_{k-\Lambda-L-1} := 0), \tag{2.3.5}$$

(or by (2.2.6) in the noisy case) where $\widetilde{u}_k$ is binary, but, by the notation $\widetilde{u}_{k-\Lambda+1-j} := 0$, we mean that any feedback paths in the recursive processor (2.3.4) have been deleted. Note that the same $\widetilde{u}_k$ as (2.3.5) would be generated by a standard DFE, started in an abnormal initial state $\overline{E}_{k-\Lambda+1}$ with fictitious past decisions $\widehat{u}_{k-\Lambda} = 0, \ldots, \widehat{u}_{k-\Lambda-L-1} = 0$ and fed with the sequence of inputs $u_{k-\Lambda+1}, \ldots, u_k$. Thus, the feedforward structure in (2.3.5) is effectively a "sliding-window" version of the DFE which resets its initial conditions at each time instant. We shall have more to say about this in section 2.3.4.

2.3.2 Worst Case Channels

Extending the development found in [23], we now determine a class of channels on which the DFE has the worst possible performance (in terms of error propagation) from an arbitrary initial condition in the extended error state space $\overline{I\!E}^L$. This will allow us to bound the DFE's noise-free performance, and subsequently the feedforward emulator's performance, on an arbitrary FIR channel.

Any channel

$$\mathbf{h} = [h_0 = 1, \underbrace{h_1, \ldots, h_L}_{\underline{h}'}]'$$

satisfying

$$\min_{\overline{E}_k \neq 0} |\underline{h}' \overline{E}_k| > 1 \tag{2.3.6}$$

has the property that $Pr(\hat{u}_k \neq u_k) = \frac{1}{2}$ for any non-zero extended error state $\overline{E}_k$. This follows since the inputs are equiprobable and therefore u_k has a probability of $\frac{1}{2}$ of having the same sign as $\underline{h}' \overline{E}_k$ (recall that $\hat{u}_k = \text{sgn}(u_k + \underline{h}' \overline{E}_k)$). Channels satisfying (2.3.6) will be termed "worst case" channels. We claim that the expected error recovery time[2] is maximised for such channels, which form a subset of the worst case class in [16] since we are allowing E_k to have the additional values ± 1. Any channel (with $h_0 = 1$) whose parameters belong to the set

$$\{\underline{h} \in I\!R^L \mid h_1 > 1\} \bigcap_{j=2}^{L} \{\underline{h} \in I\!R^L \mid h_j > 1 + 2 \sum_{k=1}^{j-1} h_k\},$$

will fall into the worst case category. This is because the h_j have been spaced so far apart that no linear combination with coefficients in the set $\{0, \pm 1, \pm 2\}$ has magnitude less than 1. (The same is true of any channel that can be obtained from this set by changes of sign and/or permutation of parameters.)

As an illustration, consider the $L = 2$ case. We may take $h_1 = 1.2$ and $h_2 = 3.5$ so that

$$\min_{\overline{E}_k \neq 0} |[1.2, 3.5]\, \overline{E}_k| = 3.5 - 2 \times 1.2 = 1.1 > 1,$$

demonstrating that $[1, 1.2, 3.5]$ is a worst case channel. That is, no second order FIR channel may have an expected recovery time which exceeds that of the above channel.

2.3.3 Bound for the Decision Feedback Equalizer

In what follows we may assume the DFE is operating on a noiseless worst case channel so as to obtain a tight performance bound. The order of the FSMP model can further be reduced by aggregating states, provided that states to be aggregated have identical transition probabilities [60]. Here, we are able to impose a structured aggregation of the $5^L - 1$ non-zero extended states $\overline{E}_k$ while retaining the Markov property between the aggregated states. We choose to aggregate the $\overline{E}_k$ states

[2] Expected time to reach the absorbing state.

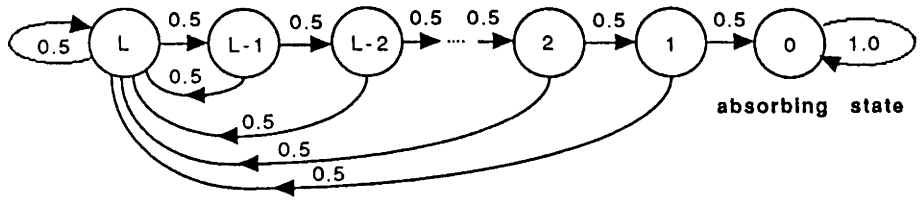

Figure 2-5: Aggregated FSMP for a worst case FIR(L) channel.

according to the following rule [18].

Definition 2.3.1 (Aggregation Rule) *The extended error state $\overline{E}_k$ at time k, defined by (2.3.1) with components in $\overline{\mathbb{E}}$, is in aggregated state $\epsilon_k = q$ if there exists a binary input sequence $\{u_j\}_{j=k}^{k+q-1}$ such that the absorbing state $E_{k+q} = \underline{0}$ is reached in q steps (while it cannot be reached in fewer).*

From the shift register property (2.3.3) applied to $\overline{E}_k$ (by allowing $\hat{u}_k = 0$), it is clear that at most L successive correct decisions are needed to force an arbitrary non-zero $\overline{E}_k$ state to the absorbing state. Hence there are $L+1$ aggregated states ϵ_k in the new FSMP. From a given state $\epsilon_k = q$ ($q \neq 0$) there is a probability of $\frac{1}{2}$ (for equiprobable inputs) of transiting either to state $\epsilon_{k+1} = q - 1$ (with a correct decision) or to state $\epsilon_{k+1} = L$ (with an incorrect decision) at the next time instant (see Fig.2-5). Subsequently, the transition probability matrix can be partitioned as

$$P \triangleq (p_{ij}) = \left[\begin{array}{c|c} Q & 0_L \\ \hline r' & 1 \end{array}\right] \in \mathbb{R}^{(L+1)\times(L+1)} \tag{2.3.7}$$

where

$$p_{ij} = Pr(\epsilon_{k+1} = L + 1 - i \mid \epsilon_k = L + 1 - j)$$

and

$$Q = \left[\begin{array}{ccc|c} \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2}I_{L-1} & & 0_{L-1} \end{array}\right] \in \mathbb{R}^{L\times L} \tag{2.3.8}$$

$$r' = [0, \ldots, 0, 1/2] \in \mathbb{R}^L.$$

Since $\epsilon_k = 0$ is the unique absorbing class (containing only $E_k = \underline{0}$), the eigenvalues of Q are less than one in magnitude [8, 60]. Let π_k be the $(L+1)$-vector whose ith component $\pi_{k,i}$ is the probability of the aggregated system state at time k being

$L+1-i$, or

$$\pi_{k,i} = Pr(\epsilon_k = L+1-i),\ i = 1,\ldots,L+1,$$

Then this state distribution vector evolves according to

$$\pi_{k+1} = P\pi_k. \tag{2.3.9}$$

Now suppose the initial extended error state $\overline{E}_0 \in \overline{\mathbb{E}}^L$ induces the distribution π_0 at time $k = 0$. We can compute the probability of the system failing to reach the absorbing state ϵ_{k+1} at time $k+1$, while operating on a worst case channel, as

$$p_k(\pi_0) \triangleq Pr(\epsilon_{k+1} \neq 0 \mid \pi_0) = \sum_{i=1}^{L} Pr(\epsilon_{k+1} = i \mid \pi_0). \tag{2.3.10}$$

In other words, $p_k(\pi_0)$ is the sum of the first L components of the vector π_k. If we partition π_k conformably with (2.3.7) as

$$\pi_k = \begin{bmatrix} \overline{\pi}_k \\ \rho \end{bmatrix},$$

and make repeated use of (2.3.9), we have

$$p_k(\pi_0) = [\underbrace{1\ \cdots\ 1\ 0}_{L+1}]\pi_k = [\underbrace{1\ \cdots\ 1}_{L}]Q^k\overline{\pi}_0,$$

in which $\overline{\pi}_0$ is the initial distribution across non-zero aggregated error states ϵ_k. Applying the power method to the matrix Q, we obtain the upper bound stated in the following theorem (this result is a mild generalisation of the analogous result in [23] concerning the DFE).

Theorem 2.3.1 (Noiseless Error Bound - DFE) *Consider the tuned DFE with output given by (2.3.2) with $n_k = 0$, operating on a noiseless worst case FIR channel of order L, and initially in non-zero extended error state $\overline{E}_0 \in \overline{\mathbb{E}}^L$ at time 0. If $\overline{E}_0$ induces the aggregated state distribution π_0, the probability of not reaching the absorbing state $\epsilon_k = 0$ at time k is given by*

$$p_k(\pi_0) \triangleq Pr(\epsilon_k \neq 0 \mid \pi_0) = \alpha_1\lambda_1^k(1 + o(2^{-k})),$$

L	2	3	4	5	6
λ_1	0.8090	0.9196	0.9638	0.9830	0.9918

Table 2.1: Dominant eigenvalue of Q.

where $\alpha_1 = \widehat{w}_1' \overline{\pi}_0 \in (0, 1]$,

$$\lambda_1 \stackrel{\Delta}{=} \max_{1 \le i \le L} \lambda_i(Q) \in (0, 1),$$

is the unique dominant eigenvalue of Q *defined in (2.3.8), and* $\widehat{w}_1 = w_1/|w_1| \in I\!R^L$ *where* w_1 *is the eigenvector of* Q *corresponding to* λ_1, *given by*

$$w_1 = [1, \sum_{j=1}^{L-1} \mu^j, \sum_{j=1}^{L-2} \mu^j, \ldots, \mu + \mu^2, \mu]',$$

with $\mu = (2\lambda_1)^{-1}$.

Since the calculations assumed a worst case channel, the above bound indicates the highest degree, on average, to which an initial error can influence subsequent decisions by error propagation alone on any noiseless channel. The bound may be conservative in that $\lambda_1 \approx 1$ for worst case channels. We list λ_1 for various channel lengths (L) in Table 2.1. For practical channels (*e.g.*, with decaying impulse responses) the exponential form of the bound is still valid (with correspondingly smaller λ_1), but in general it is not possible to aggregate the FSMP model to obtain the L^{th} order description used above, so the full order 5^L non-aggregated FSMP would need to be used.

2.3.4 Bound for the Feedforward Emulator

In the last section we saw that the probability of an incorrect decision in the DFE, due to some initial decision error in the absence of noise, becomes exponentially less likely with time. Equivalently, we can say that after a sufficient time, the effect of an initial error is negligibly small. Because of the close link between the DFE and the recursive processor (2.3.4) (or (2.2.5) in the noisy case), this "settling time" will be shown to be an indicator of the number of layers Λ required to produce a feedforward

emulator (2.3.5) (or (2.2.6) in the noisy case) that is a good approximation to the original DFE (in an error probability sense).

At each time instant k, the noise-free feedforward emulator (2.3.5) is equivalent (in the sense of producing the same output from a given sequence of inputs) to a (tuned) noiseless DFE that has been initialised in the non-zero error state

$$\overline{E}_{k-\Lambda+1} = (u_{k-\Lambda}, \ldots, u_{k-\Lambda-L-1})' \tag{2.3.11}$$

at time $k - \Lambda + 1$. Recall that we have forced $\widetilde{u}_{k-\Lambda+1-j} = 0$ for $j = 1, \ldots, L$.

We can reformulate (2.3.5) in a way which reflects more lucidly the internal structure of the feedforward emulator. We denote by γ_k^i $(1 \leq i \leq \Lambda)$ the internal binary decision generated at the i^{th} layer, used in computing the eventual output $\widetilde{u}_k$ at time k. (In other words, γ_k^i is the output of the i^{th} diagonal node in the processor of Fig.2-4 when $d_i = h_i$ $\forall i$ and there is no noise.) These preliminary decisions are obtained iteratively as follows:

$$\gamma_k^{i+1} = \begin{cases} \operatorname{sgn}(u_{k-\Lambda+i+1} + H'\overline{E}_{k-\Lambda+i+1}), & \text{if } 0 \leq i \leq \Lambda - 1 \\ 0, & \text{otherwise} \end{cases}$$

$$\overline{E}_{k-\Lambda+i+1} = (u_{k-\Lambda+i} - \gamma_k^i, \ldots, u_{k-\Lambda+i-L+1} - \gamma_k^{i+1-L})' \in \overline{I\!E}^L, \tag{2.3.12}$$

and $\widetilde{u}_k = \gamma_k^\Lambda$. Note that we have assigned $\gamma_k^i = 0$ for $i < 1$ to match the initial conditions and produce the same $\widetilde{u}_k$ as in (2.3.5). Thus $\widetilde{u}_k$ is the output of a FSMP with initial state $\overline{E}_{k-\Lambda+1}$ driven by the input sequence $\{u_{k-\Lambda+1}, \ldots, u_k\}$, passing through the sequence of states $\overline{E}_{k-\Lambda+i+1}$ $(i = 0, \ldots, \Lambda - 1)$. In what follows, we take $[h_0 = 1, \underline{h}']$ to be a worst case channel and aggregate the states according to Definition 2.3.1.

We can calculate the probability that the feedforward emulator decision is correct at time k, supposing an "initial" state distribution $\pi_{k-\Lambda+1}$. We use Bayes' rule to condition on the aggregated state ϵ_k corresponding to the extended state $\overline{E}_k$ (defined by (2.3.12)) at the output of the Λ^{th} layer of the feedforward emulator. Thus

$$Pr(\widetilde{u}_k = u_k | \pi_{k-\Lambda+1}) = Pr(\widetilde{u}_k = u_k | \pi_{k-\Lambda+1}, \epsilon_k \neq 0) Pr(\epsilon_k \neq 0 | \pi_{k-\Lambda+1})$$

$$+ \quad Pr(\tilde{u}_k = u_k | \pi_{k-\Lambda+1}, \epsilon_k = 0) Pr(\epsilon_k = 0 | \pi_{k-\Lambda+1}).$$

Now, the ϵ_k states have the transition diagram (Fig.2-5) corresponding to the choice of a worst case channel. Hence in the absorbing state $\epsilon_k = 0$ the probability of a correct decision is 1, whereas for all states ($\epsilon_k \neq 0$) outside the absorbing state this probability is $\frac{1}{2}$. We now have

$$\begin{aligned} Pr(\tilde{u}_k = u_k | \pi_{k-\Lambda+1}) &= \frac{1}{2} p_k(\pi_{k-\Lambda+1}) + 1 - p_k(\pi_{k-\Lambda+1}) \\ &= 1 - \frac{1}{2} p_k(\pi_{k-\Lambda+1}), \end{aligned} \tag{2.3.13}$$

where, like (2.3.10), we have defined

$$p_k(\pi_l) \stackrel{\Delta}{=} Pr(\epsilon_{k+1} \neq 0 \mid \pi_l).$$

However, unlike the DFE, the "initial" distribution $\pi_{k-\Lambda+1}$ (for each k) is not arbitrary. In fact all "initial" error states $\overline{E}_{k-\Lambda+1}$ for the feedforward emulator (2.3.5) belong to the aggregated state $\epsilon_{k-\Lambda+1} = L$. To see this, recall the shift register property for the extended states (2.3.12). Clearly we would need at least a sequence of L correct decisions $\gamma_k^{i+1} = u_{k-\Lambda+i+1}$ $(i = 0, \ldots, L-1)$ to attain the absorbing state $\overline{E}_{k-\Lambda+L} = \underline{0}$. Therefore, the particular "initial" distribution we seek is

$$\pi^*_{k-\Lambda+1} = [\underbrace{1, 0, \ldots, 0}_{L+1}]', \ \forall k$$

corresponding to $\epsilon_{k-\Lambda+1} = L$.

We now drop the conditioning on the left hand side of (2.3.13) since only one $\pi_{k-\Lambda+1}$ is possible in the feedforward emulator case. Applying Theorem 2.3.1 to evaluate $p_k(\pi_0)$ with $\pi_0 \stackrel{\Delta}{=} \pi^*_{k-\Lambda+1}$, yields the asymptotic formula for the error probability which we state below.

Theorem 2.3.2 (Noiseless Error Bound - FFE) *The noiseless decision error probability $Pr(\tilde{u}_k \neq u_k)$ for the Λ-layer feedforward emulator generated by (2.3.5),*

with worst case channel parameters, is given by

$$Pr(\widetilde{u}_k \neq u_k) = \frac{1}{2}p_k(\pi^*_{k-\Lambda+1}) = \frac{1}{2}\alpha_1^*\lambda_1^{\Lambda-1}(1+o(2^{-\Lambda+1})),$$

where

$$\alpha_1^* = \left(1+\sum_{j=1}^{L-1}(\sum_{i=1}^{j}\mu^i)^2\right)^{-\frac{1}{2}},$$

where $\mu \triangleq (2\lambda_1)^{-1}$ *and* λ_1 *is the (unique) dominant eigenvalue of* Q *(2.3.8).*

The reasoning carries over to the feedforward emulator because of the definition of the extended error state $\overline{E}_k$, which allows for the disconnection of the feedback part in the recursive representation (2.3.4). The above bound is *tight* in the sense that certain channels that realise the bound exist, but on practical channels, fewer layers would be required to obtain the same noiseless error probability. The reasoning leading to Theorem 2.3.2 implies that a Λ-layer feedforward emulator with sign nodes, operating on a general FIR channel of order L, has a noiseless error probability which is upper-bounded by

$$Pr(\widetilde{u}_k \neq u_k) \leq \frac{1}{2}[\underbrace{1,\ldots,1}_{L}]\, Q^{\Lambda-1}\, [\underbrace{1,0,\ldots,0}_{L}]' \qquad (2.3.14)$$

where Q is given by (2.3.8).

2.4 Evaluation of the Non-Adaptive System

2.4.1 Tuned Noiseless Performance

Here, we give an example of the simulated performance of the tuned feedforward emulator on a second order channel. The weights d_i, $i \leq L$ are equal to the corresponding channel impulse response coefficients h_i and we assign d_i, $i > L$ to be zero.

In Fig.2-6 we have plotted the upper bound resulting from the aggregated FSMP realised by a worst case second order channel $[1.0, 1.2, 3.5]$. On the same graph, the bit error probability (simulated over 10^6 points) for a feedforward emulator with sign nodes operating on the second order partial response channel $[1, 2, 1]$, as a function

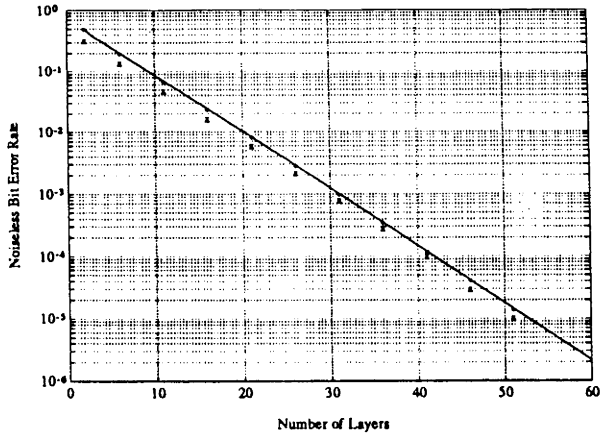

Figure 2-6: Noiseless bit error rate of tuned FFE.
FFE implemented with sign nodes on (i) $[1, 2, 1]$ partial response channel ("x"); and (ii) $[1, 1.2, 3.5]$ worst case channel—realising the upper bound (solid line).

of the number of layers Λ, is also displayed. The simulation clearly shows the agreement of the theoretical noiseless error probability bound for the FFE (2.3.14) and the simulated noiseless performance on a worst case channel.

2.4.2 Exact Noise-free Representation

The possibility arises of exact representation of a DFE by the feedforward emulator structure in the absence of noise. This is indeed the case when the error propagation events for the DFE have a guaranteed finite length. For certain classes of channels $[h_0 = 1, \underline{h}']$ (which define a stability class [16]), for example, those which satisfy the following frequency-domain sufficient condition:

$$1/2 + \sum_{k=1}^{L} h_k \cos(k\theta) > 0, \ \forall \theta \in [0, 2\pi) \tag{2.4.1}$$

it is known that the DFE has a deterministic, bounded error recovery time from a non-zero error state [24]. If we call this time R, then the feedforward emulator derived from the DFE for this stable channel will reproduce the output sequence $\{\widehat{u}_k\}$ exactly (from the same sequence of inputs) if the number of layers is chosen to be greater than or equal to $R + L$. The reason for this is that the decision $\widehat{u}_k$ is independent of $\widehat{u}_{k-\Lambda-j+1}$ for $j > 0$ due to the finite extent of error propagation on stable channels. Hence (including a further L layers to regain the $\mathbb{E}^L$ state space) a feedforward emulator with $R + L$ layers or more loses no information relative to

Λ	2	3	4	5
BER (i)	0.04129	0.02003	0.01287	0.00000
BER (ii)	0.01070	0.00100	0.00000	0.00000

Table 2.2: Noiseless bit error rate of FFE.

deciding $\widehat{u}_k$ by neglecting these past decisions, bearing in mind that we are only treating the noiseless case. In some cases, fewer than $R + L$ layers will secure this property.

We have tabulated in Table 2.2 the noiseless bit error rates (simulated over 10^6 points) for the feedforward emulator as a function of the number of layers Λ while operating on the following positive real FIR channels[3]:

1. An exponential impulse response channel

$$h_k = (0.8)^k \cos(k\pi/6),\ k = 0, \ldots, 10,\ h_k = 0,\ k > 10.$$

2. A 10^{th} order channel with coefficients

$$[1.0, -0.27, -0.18, -0.31, 0.27, 0.09, -0.05, 0.06, -0.08, 0.084, 0.01].$$

It is clear, in this example, that the representation becomes exact (exactly reproduces the input sequence in the absence of noise) when the number of layers is chosen to exceed the maximum duration of DFE error events, which is known to be finite on both the above channels. In the following subsection we look at the performance of the feedforward emulator in the presence of noise and give a numerical example demonstrating the monotonicity of bit error rate with respect to the number of layers.

2.4.3 Non-Adaptive Performance in the Presence of Noise

We have seen that the output of the feedforward emulator of Fig.2-4 with sign nodes becomes indistinguishable from that of a DFE (for operation on a noiseless finite

[3] *i.e.*, channels whose parameters satisfy (2.4.1).

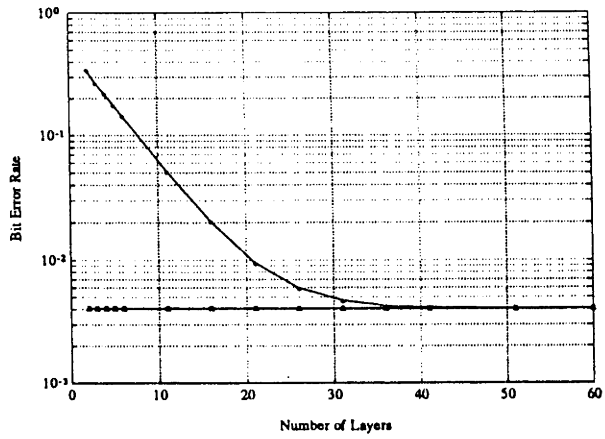

Figure 2-7: Performance of tuned FFE and DFE.
The bit error rate of the feedforward emulator (upper curve) is a function of the number of layers. The DFE's error probability is the straight line. The signal-to-noise ratio is 10dB.

error recovery channel) if the number of layers is sufficiently large. It will be apparent from the next example that the same structure can attain performance equaling that of a DFE when operating on a general FIR channel in the presence of noise. In Fig.2-7, the bit error rate of a DFE on the $[1, 2, 1]$ binary partial response channel has been simulated (over 10^5 points) and plotted for a signal-to-noise ratio[4] of 10dB. On the same graph we show the simulated error probability versus the number of layers of a feedforward emulator with sign nodes whose weights have been tuned to the channel coefficients (see section 2.4.1). Clearly, the two systems perform with the same probability of error when the number of layers in the feedforward emulator is 50 or greater, and the bit error rate of the FFE decreases monotonically with the number of layers Λ.

2.5 Adaptive Aspects

In the preceding sections, we have developed a feedforward emulator structure with sign nodes (2.2.6) which approximates a non-adaptive DFE when the branch weights are correctly assigned. In this section we address the problem of how to adapt the weights in the emulator, using a training sequence, so that they converge to a setting in which the system acts as an equalizer. We will derive two algorithms

[4]The signal-to-noise ratio is defined as $10 \log_{10} \sigma^2$ where σ^2 is the variance of the white noise.

designed to achieve this, the first applies to FFEs with nodes having a sigmoidal characteristic (which we define below)—as is the norm in work involving training of MLP neural networks. The second and less conventional algorithm applies to FFEs with sign nodes. Both are based on a stochastic gradient descent strategy [61] whereby a criterion is chosen that reflects the error between the actual and the desired performance as a function of the weights. The weights are then adjusted recursively so as to minimise the error.

Typically, for a standard artificial neural network (with the multilayer perceptron architecture), the algorithm which is used in the adaptation or training is called back propagation [40]. In the present case, the training problem differs fundamentally from the conventional problem to which back propagation is applied, in that the weights in the FFE are not independent, but are constrained in a Toeplitz manner. Any training algorithm, regardless of the activation functions used in the nodes, should take this special structure into account. In the following sections we derive two such training algorithms and evaluate their performance under simulation.

2.5.1 Training with Sigmoid Nodes

In keeping with conventional neural network training, we first consider a gradient descent scheme, motivated by back propagation ideas, for adapting the weights in the feedforward emulator assuming that the nodes have the following sigmoidal characteristic:

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} = \tanh(x/2). \tag{2.5.1}$$

We prefer to use bipolar activation functions in developing the training scheme since they closely reflect the desire to force the node outputs to take the values ± 1 (in minimising the gradient of the objective function). In fact, $f(\alpha x) \rightarrow \mathrm{sgn}(x)$ as $\alpha \rightarrow +\infty$, so we might expect FFEs with sigmoid nodes to have similar attributes to their counterparts with sign nodes as α is increased from unity (this is born out in the simulations of section 2.5.3). We also note that the sigmoid function satisfies the first order differential equation

$$\frac{df(\alpha x)}{dx} = \frac{2\alpha e^{-\alpha x}}{(1 + e^{-\alpha x})^2} = \frac{\alpha}{2}[1 - f(\alpha x)][1 + f(\alpha x)] \triangleq \alpha\phi(f(\alpha x)), \tag{2.5.2}$$

where $\phi(x) \triangleq \frac{1}{2}(1-x^2)$ and α is a real constant.

Suppose at time k the input pattern $\{y_{k-\Lambda+1}, \ldots, y_k\}$ (where y_j is a noisy channel output generated by (2.2.1)) has been applied to the feedforward emulator of Fig.2-4. If the output of the FFE is $\tilde{u}_k$, we define the instantaneous output error as

$$J_k = \frac{1}{2}(\tilde{u}_k - u_k)^2, \tag{2.5.3}$$

in which u_k is the current channel input in equation (2.2.1). Clearly $J_k = 0\ \forall k$ in the case of correct equalization. Unlike blind equalization, we assume that a known training sequence $\{u_k\}$ is available for the adaptation.

Regarding Fig.2-4, we make the following definitions.

Definition 2.5.1 *At time k, for any layer in the Λ-layer FFE of Fig.2-4, and for $1 \le i, j \le \Lambda$, we define $w_{ij}(k)$ as the weight of the branch connecting the node at level i in that layer to the node at level j in the next layer. The output of the node in layer i at level j is $y_j^i(k)$. The external input to the diagonal node whose output is $y_i^i(k)$ is $y_{k-\Lambda+i}$, where y_l is defined in (2.3.12).*

For the diagonal nodes we can drop the subscript, defining $y^j \triangleq y_j^j$. The Toeplitz structure of the FFE imposes the following constraints on the weights w_{ij} (for all k):

$$w_{ij}(k) = \begin{cases} -d_{j-i}(k), & i < j \\ 1, & i = j \\ 0, & i > j, \end{cases} \tag{2.5.4}$$

where $d_j(k)$ is the jth tap gain in the corresponding adaptive DFE (2.2.2). The outputs of the individual nodes can subsequently be expressed as

$$y_j^i(k) = \begin{cases} f(\sum_{l=1}^{j-1} w_{lj}(k) y_l^{j-i}(k) + y_{k-\Lambda+j}), & i = j \\ f(y_j^{i-1}(k)), & i > j \\ 0, & i < j. \end{cases} \tag{2.5.5}$$

Here, $y_j^j(k)$ is to be compared with γ_k^j in (2.3.12) in the case of a tuned, noiseless FFE with sign nodes. In particular, $y^\Lambda(k) = \tilde{u}_k$ is the output of the FFE at time k. It is understood that the weights w_{ij} for $i \ge j$ are to remain fixed (as indicated in (2.5.4))

during and after the training phase. The equality constraints (2.5.4) acting on the $w_{ij}(k)$ indicate that only $\Lambda - 1$ distinct parameters (the $d_j(k)$, $j = 1, \ldots, \Lambda - 1$) need be varied independently to force the feedforward emulator to act as a channel equalizer. The objective of the training algorithm is to adapt the $d_j(k)$ so that the instantaneous output error J_k is minimised. This may be achieved iteratively via the following stochastic gradient algorithm:

$$d_j(k+1) = d_j(k) - \eta \frac{\partial J_k}{\partial d_j(k)}, \; j = 1, \ldots, \Lambda - 1. \tag{2.5.6}$$

The step size $\eta > 0$ (or learning rate, borrowing a term from the literature on neural networks) is adjusted to trade off speed versus accuracy of convergence to the minimum of the mean error, while maintaining the algorithm's stability.

The training algorithm is specified once we have found a (recursive) rule for evaluating the derivatives in (2.5.6). We now turn our attention to this task. Two applications of the chain rule yield

$$\frac{\partial J_k}{\partial d_j(k)} = (\tilde{u}_k - u_k) \frac{\partial \tilde{u}_k}{\partial d_j(k)}, \; j = 1, \ldots, \Lambda - 1, \tag{2.5.7}$$

and

$$\frac{\partial \tilde{u}_k}{\partial d_j(k)} = \sum_{i=1}^{\Lambda - j} \frac{\partial \tilde{u}_k}{\partial w_{i,i+j}(k)} \frac{\partial w_{i,i+j}(k)}{\partial d_j(k)}, \; j = 1, \ldots, \Lambda - 1. \tag{2.5.8}$$

Equation (2.5.4) implies that the rightmost term of each product in the above sum equals $-1 \; \forall k$. We therefore require an expression for the derivatives of the output with respect to the weights. This is furnished below (an inductive proof may be found in appendix B.1).

Lemma 2.5.1 *Consider the system described by the equations (2.5.1), (2.5.4) and (2.5.5), which has the structure shown in Fig.2-4 with sigmoid nodes replacing the sign nodes. The branch weights which give a non-zero contribution to the derivative of the output* $\tilde{u}_k = y^{\Lambda}(k)$ *with respect to* $d_j(k)$, $1 \leq j \leq \Lambda - 1$ *(2.5.8) are* $w_{i,i+j}(k)$, $1 \leq i \leq \Lambda - j$, *and the derivatives with respect to these weights are given by*

$$\frac{\partial \tilde{u}_k}{\partial w_{i,i+j}(k)} = \left\{ \prod_{l=i+j}^{\Lambda - 1} \phi(y^{l+1}(k)) w_{l,l+1}(k) \right\} \frac{\partial y^{i+j}(k)}{\partial w_{i,i+j}(k)}, \tag{2.5.9}$$

where $\phi(\cdot)$ is given in (2.5.2), and (as before) we have set $y^j(k) = y_j^j(k)$.

Furthermore, the derivative on the right hand side of 2.5.9 may be evaluated directly from (2.5.5) as

$$\begin{aligned}\frac{\partial y^{i+j}(k)}{\partial w_{i,i+j}(k)} &= \frac{\partial}{\partial w_{i,i+j}(k)} f\Big(\sum_{l=1}^{i+j-1} w_{l,i+j}(k) y_l^{i+j-1}(k) + y_{k-\Lambda+i+j}\Big) \\ &= \phi(y^{i+j}(k)) \sum_{l=1}^{i+j-1} y_l^{i+j-1}(k)\delta_{il} \\ &= \phi(y^{i+j}(k)) y_i^{i+j-1}, \end{aligned} \tag{2.5.10}$$

where δ_{il} is the Kronecker delta and we have used the fact that the $y_l^{i+j-1}(k)$, for $1 \le l \le i+j-1$, do not depend on $w_{i,i+j}(k)$.

Consolidating lemma 2.5.1 with (2.5.8) and (2.5.10), we now have (suppressing the time index k),

$$\begin{aligned}\frac{\partial \tilde{u}}{\partial d_j} &= -\sum_{i=1}^{\Lambda-j} \frac{\partial y^\Lambda}{\partial w_{i,i+j}}, \quad j = 1, \ldots, \Lambda - 1 \\ &= -\{\phi(y^\Lambda) y_{\Lambda-j}^{\Lambda-1} + \phi(y^{\Lambda-1}) w_{\Lambda-1,\Lambda} y_{\Lambda-j-1}^{\Lambda-2} + \cdots \\ &\quad + \phi(y^\Lambda) w_{\Lambda-1,\Lambda} \phi(y^{\Lambda-1}) w_{\Lambda-2,\Lambda-1} \cdots \phi(y^{j+1}) y_1^j\} \qquad (2.5.11) \\ &= -\phi(y^\Lambda)\{y_{\Lambda-j}^{\Lambda-1} - d_1 \phi(y^{\Lambda-1})\{y_{\Lambda-j-1}^{\Lambda-2} - \cdots \\ &\quad - d_1 \phi(y^{j+2})\{y_2^{j+1} - d_1 \phi(y^{j+1}) y_1^j\} \cdots\}\}, \qquad (2.5.12)\end{aligned}$$

where the (2.5.12) follows from (2.5.11) by virtue of the fact that $w_{l,l+1} = -d_1$ after nesting terms with the first term of (2.5.11) occupying the innermost bracket. This nested form lends itself easily to rewriting in the recursive form

$$\frac{\partial \tilde{u}_k}{\partial d_j(k)} = -\delta_j^{\Lambda-j}(k), \; j = 1, \ldots, \Lambda - 1, \tag{2.5.13}$$

where

$$\delta_j^1(k) \triangleq \phi(y^{j+1}(k)) y_1^j(k) \tag{2.5.14}$$

and

$$\delta_j^q(k) = \phi(y^{j+q}(k))\{y_q^{j+q-1}(k) - d_1(k)\delta_j^{q-1}(k)\},\ q = 2,\ldots,\Lambda - j. \qquad (2.5.15)$$

This last expression shows how the self-similarity between successive layers of the feedforward emulator contributes to the calculation of the output derivative. Put into words, the derivative of the output with respect to d_j is the sum of two terms: the contribution from the branch with weight d_j to the output node; and: d_1 times the derivative of the output of the previous diagonal node with respect to d_j (and so on). We can now state the recursive training rule, which we will refer to as the *sigmoid algorithm.*

Theorem 2.5.1 (Sigmoid Algorithm) *The stochastic gradient descent algorithm arising from the minimisation of the criterion (2.5.3), for adaptation of the weights $d_j(k)$ of the Λ-layer feedforward emulator with output $\tilde{u}_k$ and sigmoid nodes replacing the sign nodes, takes the form*

$$d_j(k+1) = d_j(k) + \eta(\tilde{u}_k - u_k)\delta_j^{\Lambda-j}(k),\ j = 1,\ldots,\Lambda - 1, \qquad (2.5.16)$$

where δ_j^i is computed recursively from (2.5.14) and (2.5.15), u_k is the known training input sequence to the channel and $\eta > 0$ is a small step size parameter.

The reader is referred to section 2.5.3 for an example of adaptation during training.

2.5.2 Training with Sign Nodes

Returning to the original feedforward emulator with sign nodes (Fig.2-4), we consider a recursive training procedure for the adaptation of the weights $d_j(k)$. Retaining the notation introduced in the last subsection, we redefine $f(\cdot)$ as $sgn(\cdot)$ and compute the node outputs y_j^i as in (2.5.5). The error measure used in the sigmoid algorithm is unsuitable in this case since it is insensitive to changes in the weights. Just as in FSMP analysis of error propagation in the DFE [23], the sign of a weighted sum of (fixed) inputs will be the same for a range of weight values. We therefore define a new measure of error which more closely reflects the effect of changes in weights during adaptation

$$J_k \doteq \frac{1}{2}(z_k - u_k)^2, \tag{2.5.17}$$

where

$$z_k = y_k - \sum_{i=1}^{\Lambda-1} d_i(k) y_{\Lambda-i}^{\Lambda-1}(k) \tag{2.5.18}$$

is the weighted sum of inputs to the output node of the feedforward emulator, *i.e.*, $\widetilde{u}_k = \mathrm{sgn}(z_k) \in \{-1, +1\}$. The training rule follows easily from the calculation of the gradient of J_k with respect to $d_j(k)$. Explicitly,

$$\frac{\partial J_k}{\partial d_j(k)} = (z_k - u_k)\frac{\partial z_k}{\partial d_j(k)}.$$

But

$$\frac{\partial z_k}{\partial d_j(k)} = -y_{\Lambda-j}^{\Lambda-1}(k) - \sum_{\substack{i=1 \\ i \neq j}}^{\Lambda-1} d_i(k)\frac{\partial y_{\Lambda-i}^{\Lambda-1}(k)}{\partial d_j(k)}.$$

The gradient of the sign function is zero everywhere except at the origin. Since the occurrence of a zero argument in any sign function in a node of the FFE is a probability zero event in the presence of noise, we assign the derivative of $\mathrm{sgn}(\cdot)$ to be zero everywhere. We thus obtain the following recursive rule stated in the theorem below. We will refer to this as the *sign algorithm.*

Theorem 2.5.2 (Sign Algorithm) *The stochastic gradient descent algorithm arising from the minimisation of the criterion (2.5.17), for adaptation of the weights $d_j(k)$ of the Λ-layer feedforward emulator with sign nodes, takes the form*

$$d_j(k+1) = d_j(k) + \eta(z_k - u_k)y_{\Lambda-j}^{\Lambda-1}, \; j = 1, \ldots, \Lambda - 1, \tag{2.5.19}$$

where u_k is the known training input sequence to the channel, z_k is given by (2.5.18) and η is the step size.

Whereas the sigmoid algorithm has contributions to the derivative from several layers of the FFE (in general), the only contribution in the sign algorithm is from the output of the preceding layer. This accounts for the algorithm's relative simplicity. The resemblance of the sign algorithm to the *perceptron algorithm* of Rosenblatt [62] is more than coincidental as both use nodes with hard-limiting activation functions.

Examples of the performance of the respective algorithms are to be found in the next section.

2.5.3 Simulation Examples of Adaptation with Noise

As announced, we now proceed with an evaluation of the two adaptation algorithms. We chose a second order "closed-eye" channel $[h_0 = 1.0, h_1 = 0.8, h_2 = -0.3]$, with a SNR of 10dB, and simulated both the sigmoid and sign training algorithms with a step size η of 0.0001 over 10^5 points. In both cases, we observed the evolution of the weights $d_1, \ldots, d_{10}$ in an 11-layer feedforward emulator corresponding to Fig.2-4. In this case, satisfactory operation as an equalizer (without decision delay) requires the weights to converge to a neighbourhood of the point in parameter space given by

$$[d_1 = 0.8, d_2 = -0.3, d_3 = d_4 = \cdots = d_{10} = 0.0].$$

Convergence of the weights using the sigmoid algorithm (Theorem 2.5.1) was observed to require considerably more than 10^5 training points. The speed of convergence was greatly increased (along with an increase in jaggedness of the weight trajectories) of by using an "acceleration factor" α greater than unity, so that x in equation (2.5.1) is replaced by αx and $\phi(x)$ (equation (2.5.2)) is replaced with $\alpha\phi(\alpha x)$ in the sigmoid algorithm. Fig.2-8 shows the time evolution of the weights in the sigmoid algorithm from random initial values with an acceleration factor of $\alpha = 10$. As can be seen in Fig.2-8, the weight trajectories are converging to the expected equilibrium settings for equalization, although more slowly than the corresponding trajectories for the sign algorithm (theorem 2.5.2) in Fig.2-9.

As a further example of the adaptive feedforward emulator's operation, we have included Fig.2-10. This shows the weight-space trajectories ($d_1(k)$ versus $d_2(k)$) of a three-layer FFE during sign algorithm adaptation on the $[1, 2, 1]$ channel at an SNR of 10dB with a step size $\eta = 0.001$ over 10^5 iterations. Each parameter trajectory has been decimated to 100 points. All trajectories were initialised on the boundary of the rectangle $[-10 \leq d_1(k) \leq 10] \times [-10 \leq d_2(k) \leq 10]$. The equilibrium point[5] of the adaptive algorithm is $[d_1 \approx 1.1, d_2 \approx 0.10]$ in this case.

[5] *i.e.*, the point of convergence of the weights for noiseless adaptation with small step size.

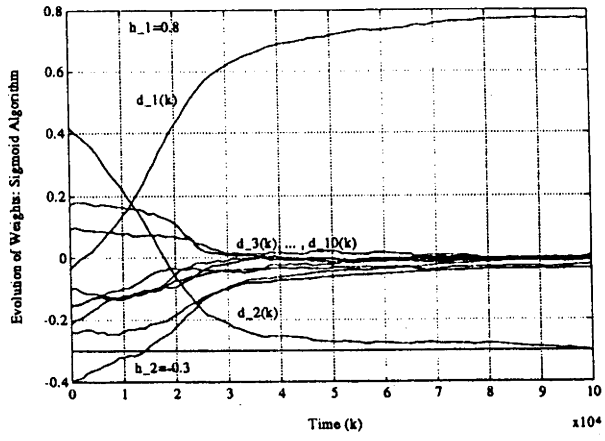

Figure 2-8: Accelerated sigmoid algorithm training of FFE.

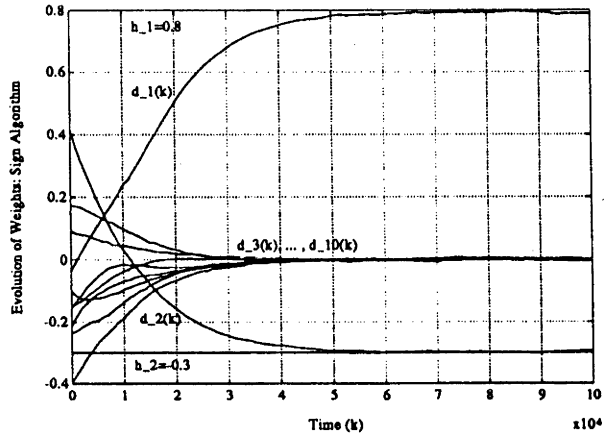

Figure 2-9: Sign algorithm training of FFE.
The simulation was performed on the $[1, 0.8, -0.3]$ channel at a SNR = 10dB for a feedforward emulator with 11 layers and *sign* nodes.

Some of the trajectories have not reached the vicinity of this point by the end of the simulation run, becoming hung for long periods of time at one of various points on the branches of the central "Y" in the figure. This behaviour indicates the presence of a one-dimensional manifold of local equilibria for the adaptive algorithm in this example. The trajectories display an interesting symmetry, belying the shape of the mean error surface which the gradient algorithm is descending.

2.6 Summary and Conclusions

We described a non-linear feedforward processing structure derived from a recursive non-linear decision feedback equalizer. This new system, called a feedforward

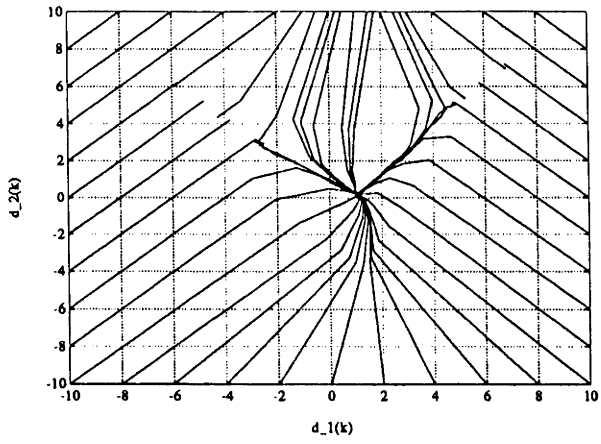

Figure 2-10: Parameter trajectories during adaptation.
Sign algorithm training with step size 0.001, SNR 10dB, three-layer FFE on the $[1, 2, 1]$ channel with $d_1(k)$ on the x-axis and $d_2(k)$ on the y-axis.

emulator, has close algebraic links with the decision feedback equalizer and close structural links with multi-layer perceptron neural networks. The connection with the DFE allowed theoretical techniques, already devised for that equalizer, to be applied with only minor modification to obtain a (noiseless) error probability bound for the FFE in terms of the number of layers in its realisation. The tightness of this theoretical upper bound was verified by simulation on a noiseless second order channel (a worst case channel). Previous results on finite error recovery channels for the DFE can be harnessed to determine classes of noiseless channels on which the representation of a tuned DFE by a tuned FFE is exact.

The training adaptation of the FFE's equality-constrained weights was considered and two stochastic gradient descent algorithms were developed. The first algorithm (the sigmoid algorithm) is applicable to FFEs with nodes having a sigmoidal activation function, and was derived using an extension of back propagation which allows for the interdependence of the weights. The second (the sign algorithm) is specifically tailored for FFEs, derived from the DFE, having hard-limiting (sign) nodes. The sign algorithm gives generally much quicker convergence of the FFE's weights for a given channel, SNR and step size of the gradient algorithm. The convergence of the sigmoid algorithm can be accelerated, although this may lead to oscillatory behaviour of the parameter trajectories.

Whereas the decision feedback equalizer can suffer from error propagation due

to the feedback of incorrect decisions, the feedforward emulator suffers from another kind of error propagation due to the truncation of the Markov expansion for the DFE from which it derives. This effectively limits the FFE's performance to that of a conventional DFE at best, despite the added delay involved in obtaining decisions. At any rate, replacing the DFE by a neural network-like device was not the motivating objective. Rather, the importance of these results lies in their solid theoretical base and the possibility of their broader application to the relatively new area of artificial neural networks. It is recognised that theoretical tools are needed which permit the designer of systems incorporating artificial neural networks to specify (or at least bound) architectural parameters such as the number of layers needed for a particular task. Analyses of the type presented in this chapter could provide a starting point for answering these questions.

Chapter 3

Block Decision Feedback Equalization

3.1 Introduction

In chapter 2, we studied a non-linear feedforward structure—the FFE, resembling a multi-layer neural network. The emphasis in that part was on the analysis of the new system which was facilitated by its close links with the decision feedback equalizer. We continue here with another development of the conventional non-linear DFE. This time, however, the emphasis is shifted from analysis to design and performance, and instead of dissecting the DFE, we build upon it.

As mentioned in section 1.2.4, the large discrepancy in performance/complexity separating the decision feedback equalizer and the Viterbi decoder has led to many attempts at designing an equalizer structure that bridges this gap. The reduced-state sequence estimator [11] and the delayed decision feedback sequence estimator [10] are two structured approaches to the problem of simplifying the Viterbi decoder. In this chapter we introduce another equalizer structure, called the *block decision feedback equalizer* (block DFE), which complements the work of [10, 11] by generalising the feedback and decision mechanism of the conventional DFE[1]. The block DFE is an *optimal* generalization of a modified DFE proposed by Clark, Lee and Marshall [43], although it was developed independently. The central idea is to make the de-

[1]We are therefore assuming that a linear prefilter has already cancelled the precursor intersymbol interference.

tection process more effective by using groups of instead of single received signals to estimate one data symbol with a fixed delay. Proakis and Khazen-Terezia [14] and more recently Moon and Carley [13] have also proposed detection algorithms based on a finite length tree search which makes use of a fixed number of previous decisions to cancel the ISI on present and future symbols. We show that these algorithms are in fact equivalent to that of [43] and demonstrate their connection to the block DFE.

In contrast to most of the preceding trellis-based ideas, the block DFE is motivated by a systems approach using block processing [45, 46, 47]. The equalizer we present can give performance arbitrarily close to the MLSE as the block length p and the number of decisions made per iteration q tend to infinity. At the other extreme ($p = q = 1$) the conventional DFE is recovered. When $q = 1$ and p is arbitrary, the block DFE corresponds to the marriage of decision feedback with the symbol-by-symbol MAP fixed-delay optimal detector [26]. This property holds for arbitrary signal to noise ratios (SNRs), whereas the schemes proposed in [14, 13, 43] are only optimal in the same sense in the high SNR limit.

We mention that the proposed system cannot be classified as *ad hoc*. The derivation mirrors in a mechanical fashion, but in a block-processing environment, the conventional DFE philosophy, *i.e.*, committed past decisions are used as if they were correct, and an *optimal* noise rejection system is then derived. Thus we build a theoretical framework in which the past suboptimal schemes [43, 14, 13] may be embedded and interpreted.

In the following sections we develop the block DFE which is composed of two essentially independent components: (i) a feedback filter, derived in section 3.2.2, based on a block processing model of the communication channel developed in section 3.2.1; and (ii) a p-input, q-output ($1 \leq q \leq p$) memoryless decision device which has various potential realisations according to the criterion of interest. We will be concerned principally with two realisations of block DFEs: the *full-blocking* block DFE or (p, p)-DFE, producing p data estimates per block iteration, and the *sliding-window* block DFE or $(p, 1)$-DFE, producing a single decision per block iteration. The latter device has an optimal realisation and a lower complexity realisation for high signal-to-noise ratios (called the *high SNR block DFE*). Section 3.3 considers low order realisations of block DFEs and presents simulations of the performance of

the device based on an example which has appeared in the literature [11]. Computational issues are addressed in section 3.3.6. We demonstrate the close connection between the block DFE (using a maximum *a posteriori* decision device), the conventional MLSE [19], and the optimal MAP detector [26, 27, 30, 31] in section 3.4, at the same time surveying the literature on minimum bit error rate detectors.

The block DFE, like the delayed decision feedback sequence estimator [10], can equalize recursive channels—the approach being to use state estimation feedback, which reduces to decision feedback in the case of FIR channels. Nonlinear channels with finite-dimensional state-space realisations are also amenable to this technique, and we give a description in section 3.5.1. Most of the treatment assumes binary signalling. The extension to M-ary signalling and QAM is the subject of section 3.5.2, noise colouration is treated summarily in section 3.5.3. A discussion of adaptation of block DFE parameters is given in section 3.5.4. We look first at deriving minimum mean-square error tap settings for the multivariable feedback filter and decision device, then obtain a stochastic gradient descent algorithm for training sequence adaptation of the block DFE. We leave consideration of performance analysis and stability to chapter 4, in which we present various analyses of the two-input block DFE (which we treat on account of the difficulty in analysing the more general p-input block DFE).

3.2 Block Decision Feedback Equalizer Development

3.2.1 Block Processing

The starting point for the generalisation of decision feedback equalization is the concept of a *block processing* communication channel model. A single-input single-output linear (IIR) channel has a transfer function $H(z)$ given by

$$H(z) = d + c(zI - A)^{-1}b = d + \sum_{j=0}^{\infty} cA^j bz^{-(j+1)}$$

where I is the $n \times n$ identity matrix, $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^{n\times 1}$, $c \in \mathbb{R}^{1\times n}$, and $d \in \mathbb{R}$. This can be realised in the time-domain by a state space system of the form [41]

$$\begin{aligned} x_{k+1} &= Ax_k + bu_k, \quad x_k \in \mathbb{R}^n \\ y_k &= cx_k + du_k + n_k, \quad k = 0, 1, 2, \ldots, \end{aligned} \tag{3.2.1}$$

where u_k, x_k and y_k denote respectively the (*iid* zero-mean binary) channel input, state and output at time k. We assume the output y_k is corrupted by zero mean white Gaussian channel noise[2] n_k with variance σ^2. Following chapter 2, we define the signal-to-noise ratio as $\text{SNR} \triangleq 10\log_{10}\sigma^2$ independently of the channel[3].

A block processing [45, 46, 47] version of (3.2.1) of *block length* p can be derived by observing that

$$\begin{aligned} x_{k+2} &= A^2x_k + Abu_k + bu_{k+1} \\ y_{k+1} &= cAx_k + cbu_k + du_{k+1} + n_{k+1} \end{aligned}$$

and

$$\begin{aligned} x_{k+3} &= A^3x_k + A^2bu_k + Abu_{k+1} + bu_{k+2} \\ y_{k+2} &= cA^2x_k + cAbu_k + cbu_{k+1} + du_{k+2} + n_{k+2} \end{aligned}$$

etc., leading to

$$\begin{aligned} x_{k+p} &= Fx_k + GU_k, \quad k = 0,\ p,\ 2p, \ldots, \\ Y_k &= Hx_k + DU_k + N_k, \end{aligned} \tag{3.2.2}$$

where

$$F \in \mathbb{R}^{n\times n} \ ; \ G \in \mathbb{R}^{n\times p} \ ; \ H \in \mathbb{R}^{p\times n} \ ; \ D \in \mathbb{R}^{p\times p}$$

are given by

$$F \triangleq A^p \ ; \ G \triangleq [A^{p-1}b \ \ A^{p-2}b \ \cdots \ Ab \ b]$$

[2]We may also denote this by $n_k \sim N(0, \sigma^2)$.

[3]Other definitions exist, for example, SNR= $\text{Var}(u_k)\sum_{i=0}^{L} h_i^2/\text{Var}(n_k)$ on a finite impulse response channel with taps h_i [19].

$$H \triangleq \begin{bmatrix} c \\ cA \\ \vdots \\ cA^{p-1} \end{bmatrix} \; ; \; D \triangleq \begin{bmatrix} d & 0 & 0 & \cdots & 0 \\ cb & d & 0 & \cdots & 0 \\ cAb & cb & d & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ cA^{p-2}b & \cdots & cAb & cb & d \end{bmatrix} \tag{3.2.3}$$

and the vectors U_k, Y_k, and N_k in (3.2.2) are given by[4]

$$\begin{aligned} U_k &\triangleq [u_k, u_{k+1}, \ldots, u_{k+p-1}]' \in \mathbb{R}^p \\ Y_k &\triangleq [y_k, y_{k+1}, \ldots, y_{k+p-1}]' \in \mathbb{R}^p \\ N_k &\triangleq [n_k, n_{k+1}, \ldots, n_{k+p-1}]' \in \mathbb{R}^p. \end{aligned} \tag{3.2.4}$$

3.2.2 Decision Feedback Structure

Continuing with the development, we now propose a decision feedback equalizer structure for the block processing realisation of the channel (3.2.2)–(3.2.4). The term DU_k in (3.2.2) is the direct feed-through term of the vector of channel outputs Y_k in (3.2.2) and can be considered as the "cursor". The term Hx_k in (3.2.2) summarizes the effect of past inputs, and therefore acts as the "tail". The term N_k in (3.2.2) is the vector of channel noise components.

We assume some arbitrary decisions have been made corresponding to a choice of $\hat{x}_k$ (a channel state estimate) via some decision procedure. We then attempt to cancel the "tail" Hx_k in (3.2.2) (as in a conventional DFE) by generating a vector $Z_k \in \mathbb{R}^p$ for $k = 0, p, 2p, \ldots$, defined by

$$\begin{aligned} Z_k &\triangleq Y_k - H\hat{x}_k \\ &= DU_k + HE_k + N_k \end{aligned} \tag{3.2.5}$$

where

$$E_k \triangleq x_k - \hat{x}_k \tag{3.2.6}$$

denotes the state estimation error (which ideally should be zero). An estimate $\hat{U}_k$ of U_k is generated by passing the signal Z_k through a memoryless decision function

[4] Recall, x' denotes the transpose of x.

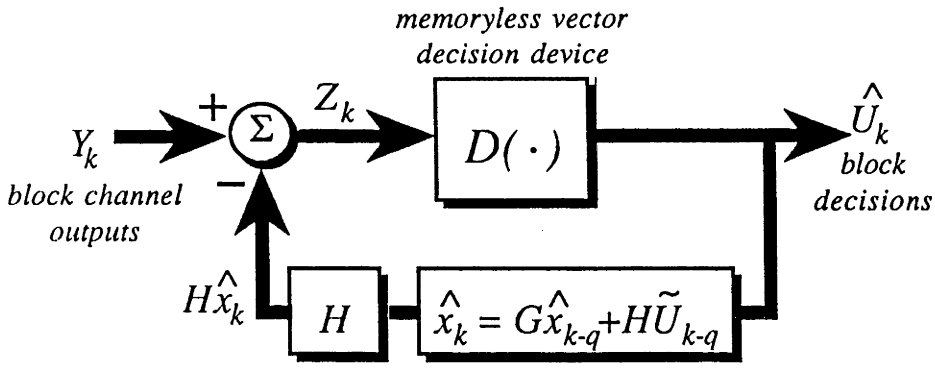

Figure 3-1: Block processing DFE structure.

$\mathcal{D}(\cdot)$ according to

$$\hat{U}_k = \mathcal{D}(Z_k) \tag{3.2.7}$$

as illustrated in Fig.3-1. Note that any reasonable decision function $\mathcal{D}(\cdot)$ may be used (*e.g.*, optimising various decision criteria, as we will see later) but the decision feedback structure is completely independent of the choice of $\mathcal{D}(\cdot)$.

3.2.3 Finite Impulse Response Channels

For the remainder of this chapter we will largely constrain the discussion to the study of channels with finite impulse responses. We remark that this is not a necessary restriction of the technique but leads to a slightly simpler and more familiar presentation. An outline of the ARMA channel case will be given at the end of section 3.2.5.

If $H(z)$ is the transfer function of a finite impulse response channel, then

$$H(z) = h_0 + \sum_{j=1}^{L} h_j z^{-j} \;\; ; \; h_L \neq 0, \tag{3.2.8}$$

so that

$$y_k = h_0 u_k + h_1 u_{k-1} + \cdots + h_L u_{k-L} + n_k, \tag{3.2.9}$$

then we have in (3.2.1) that $n = L$, and may define the state as

$$x_k = [u_{k-L}, u_{k-L+1}, \ldots, u_{k-2}, u_{k-1}]'. \tag{3.2.10}$$

We lose no generality in assuming that $h_0 = 1$. A realisation of $H(z)$ is

$$A = \begin{bmatrix} 0 & I_{L-1} \\ 0 & 0 \end{bmatrix} \; ; \; b = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \; ; \; c = [h_L, \ldots, h_2, h_1]' \; ; \; d = 1$$

where I_m denotes the $m \times m$ identity matrix and 0 denotes an appropriately dimensioned matrix (or vector) of zeros.

Hence in (3.2.2) and (3.2.3) with the state x_k given by (3.2.10), we have

$$F = \begin{bmatrix} 0 & I_{L-p} \\ 0 & 0 \end{bmatrix} \; ; \; G = \begin{bmatrix} 0 \\ I_p \end{bmatrix} \quad ; \quad \text{for } p \leq L-1$$

or

$$F = 0 \; ; \; G = [0 \;\; I_L] \quad ; \quad \text{for } p \geq L$$

while

$$H = \begin{bmatrix} h_L & h_{L-1} & \cdots & \cdots & h_2 & h_1 \\ 0 & h_L & \cdots & \cdots & h_3 & h_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & h_L & \cdots & h_p \end{bmatrix} \in \mathbb{R}^{p \times L} \tag{3.2.11}$$

and

$$D = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ h_1 & 1 & 0 & \cdots & 0 \\ h_2 & h_1 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ h_{p-1} & \cdots & h_2 & h_1 & 1 \end{bmatrix}. \tag{3.2.12}$$

Note that the channel length L and block length p are *independent.* (For convenience, when $p > L$ we define $h_k = 0$ for $L+1 \leq k \leq p$.)

In the special case of a finite impulse response channel (3.2.8) with state (3.2.10), the state estimate of x_k in (3.2.5) is simply the vector of L past decisions

$$\hat{x}_k \triangleq [\hat{u}_{k-L}, \; \hat{u}_{k-L+1}, \; \ldots, \; \hat{u}_{k-1}]'. \tag{3.2.13}$$

Writing Z_k as

$$Z_k \triangleq [z_{k,1}, \; z_{k,2}, \; \ldots, \; z_{k,p}]', \tag{3.2.14}$$

we have from (3.2.5) and (3.2.6) that the components $z_{k,i}$ are given for $p \leq L$ by (recall that $h_0 = 1$)

$$\begin{aligned}
z_{k,1} &= u_k + h_L e_{k-L} + \cdots + h_1 e_{k-1} + n_k \\
z_{k,2} &= h_1 u_k + u_{k+1} + h_L e_{k,L-1} + \cdots + h_2 e_{k-1} + n_{k+1} \\
&\vdots \quad\quad \vdots \\
z_{k,p} &= h_{p-1} u_k + h_{p-2} u_{k+1} + \cdots + u_{k+p-1} + h_L e_{k-L+p-1} + \cdots \\
&\quad + h_p e_{k-1} + n_{k+p-1}
\end{aligned}$$

(a similar set of expressions for $p > L$ can also be derived), where, for any j,

$$e_j \triangleq u_j - \hat{u}_j,$$

and from (3.2.10) and (3.2.13), E_k (3.2.5) is given for FIR channels by

$$E_k = [e_{k-L}, e_{k-L+1}, \ldots, e_{k-2}, e_{k-1}]'. \tag{3.2.15}$$

That is, for finite impulse response channels the components of E_k are just past decision errors. Equivalently, the components of Z_k can be expressed as

$$\begin{aligned}
z_{k,1} &= y_k - \sum_{j=1}^{L} h_j \hat{u}_{k-j} \\
z_{k,2} &= y_{k+1} - \sum_{j=2}^{L} h_j \hat{u}_{k-j+1} \\
&\vdots \quad\quad \vdots \\
z_{k,p} &= y_{k+p-1} - \sum_{j=p}^{L} h_j \hat{u}_{k-j+p-1}.
\end{aligned} \tag{3.2.16}$$

This expression more clearly reveals the use of decision feedback. The set of equations (3.2.16) actually appears in [43] in relation to non-linear equalizers. This treatment, however, departs from that of [34, 43, 63] by developing an optimal decision procedure which does not require the assumption of high signal-to-noise ratios, and is able to cope with ARMA channel models.

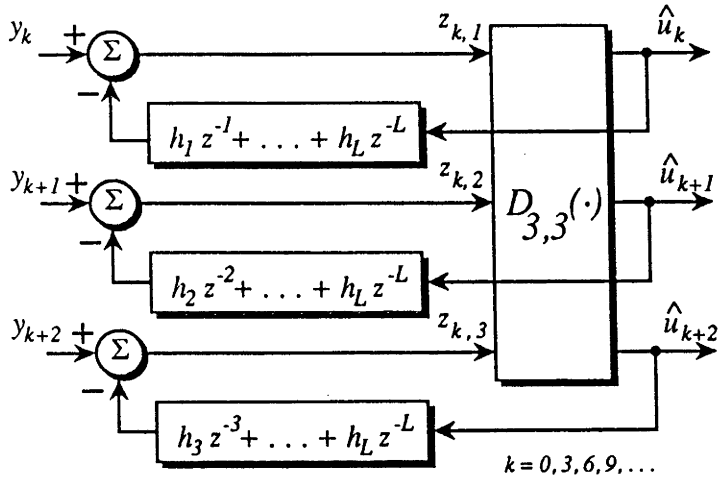

Figure 3-2: Three-input Block DFE.

The structure of a three-input ($p = 3$) block DFE for the FIR model (3.2.8) with a 3-input, 3-output decision device $\mathcal{D}_{3,3}(\cdot)$ whose inputs are the $z_{k,j}$ is illustrated in Fig.3-2.

3.2.4 Full-Blocking Maximum A Posteriori Decisions

We now use a maximum *a posteriori* (MAP), or equivalently, since we assume equiprobable *iid* input symbols [49], a maximum likelihood (ML) criterion, to develop an optimum memoryless decision function $\mathcal{D}_{p,p}(\cdot)$ with p inputs and p decision outputs for the block DFE. A MAP criterion is generally preferable because it minimises the mean error probability [6]. We stress that this is *finite subsequence* (length p) MAP detection and *not* the semi-infinite sequence estimation used in, say, the Viterbi decoder.

The vector Z_k in (3.2.14) is the decision device input which incorporates feedback of past decisions. In designing the decision device we are free to assume that there have been no past decision errors, *i.e.*, $E_k \equiv 0$ in (3.2.6), and therefore that only the noise process N_k corrupts the "block cursor" DU_k. We develop an optimal detection procedure subject to these assumptions.

The probability density of the white Gaussian noise process $N_k \in \mathbb{R}^p$ is

$$p_N(N_k) \triangleq \frac{1}{(2\pi)^{p/2}\sigma^p} e^{-\frac{1}{2\sigma^2} N_k' N_k} \tag{3.2.17}$$

where σ^2 is the variance. (Other noise distributions can be assumed provided they are independent [26].)

The MAP decision criterion demands we take as decisions those p data estimates

$$\widehat{U}_k \triangleq [\widehat{u}_k,\ \widehat{u}_{k+1},\ \ldots,\ \widehat{u}_{k+p-1}]'$$

which associate with the given input vector Z_k (3.2.5) the most probable candidate noise vector. In forming the decisions, we make the assumption of the absence of past decision errors ($E_k = 0$), *i.e.*,

$$\begin{aligned}
\widehat{U}_k &\triangleq \arg\max_{U \in \mathbb{B}^p} Pr(U_k = U |\ Z_k,\ E_k = 0) \\
&= \arg\max_{U \in \mathbb{B}^p} Pr(Z_k |\ E_k = 0,\ U_k = U) \frac{Pr(U_k = U |\ E_k = 0)}{Pr(Z_k |\ E_k = 0)} \\
&= \underbrace{\arg\max_{U \in \mathbb{B}^p} p_N(Z_k - DU)}_{\text{Maximum Likelihood}} \underbrace{Pr(U_k = U)}_{\text{constant}}
\end{aligned} \tag{3.2.18}$$

where $\mathbb{B} \triangleq \{-1, +1\}$. We use $\mathbb{B}^p$ to denote the set of vectors of dimension p with binary components. The last line in (3.2.18) follows from the independence of U_k and E_k, and the observation that $Pr(Z_k | E_k = 0)$ is constant in the maximisation. Since the input data are uniformly distributed, this MAP criterion reduces to a ML criterion. This leads to our first theorem.

Theorem 3.2.1 ((p,p)-DFE) *The memoryless decision function*

$$\mathcal{D}_{p,p}(\cdot) : Z_k \in \mathbb{R}^p \mapsto \widehat{U}_k \in \mathbb{B}^p$$

which optimises the maximum a posteriori *criterion (3.2.18) under the assumption that there has been no state estimation error (no past decision errors in the FIR case) is given for* $k = 0, p, 2p, \ldots$, *by*

$$\begin{aligned}
\widehat{U}_k &\triangleq \mathcal{D}_{p,p}(Z_k) \\
&= \arg\min_{U \in \mathbb{B}^p} \left\{ \|Z_k - DU\|^2 \right\}
\end{aligned} \tag{3.2.19}$$

where $\|v\|^2$ *denotes the* l_2*-norm* $v'v$, D *is defined in* (3.2.3).

Equation (3.2.19) is optimum (in a maximum *a posteriori* probability sense) when a *block of* p decisions is to be made. A block DFE, operating on an ARMA channel, using (3.2.19) as its decision criterion will be referred to as a *full-blocking block DFE* or (p,p)-DFE. In the next part we examine what form the decision device takes if we wish to generate q $(1 \leq q \leq p)$ decisions and advance the block processing in units of size q. (For example, decoding data in groups of size q is useful for block codes [49].) This will allow us to compare the technique with a related structure, "System 1" of [43].

3.2.5 Sliding-Window Maximum A Posteriori Decisions

We now determine the maximum *a posteriori* block-by-block decision procedure which generates $q \leq p$ decisions from p inputs for the block DFE[5]. This is to be distinguished from the MAP detectors of [26, 27] which estimate sequences of data. Denote a block of q decisions as follows:

$$\tilde{U}_k \triangleq [\tilde{u}_k,\ \tilde{u}_{k+1},\ \ldots,\ \tilde{u}_{k+q-1}]';\ k = 0,\ q,\ 2q,\ \ldots$$

The following theorem now applies.

Theorem 3.2.2 ((p,q)-DFE) *The memoryless decision function*

$$\mathcal{D}_{p,q}(\cdot):\ Z_k \in I\!R^p \mapsto \tilde{U}_k \in I\!B^q$$

for $q \leq p$ which optimises the maximum a posteriori *criterion, under the assumption that there has been no state estimation error (no past decision errors in the FIR case), is given, for $k = 0,\ q,\ 2q, \ldots$, by*

$$\begin{aligned} \tilde{U}_k &\triangleq \mathcal{D}_{p,q}(Z_k) \\ &= \underset{U \in I\!B^q}{\arg\max} \left\{ \sum_{V \in I\!B^{p-q}} e^{-\frac{1}{2\sigma^2}\left\| Z_k - D\left[\begin{smallmatrix} U \\ V \end{smallmatrix}\right] \right\|^2} \right\}. \end{aligned} \tag{3.2.20}$$

where D is defined in (3.2.3).

[5]When $q = 1$, the block DFE becomes a symbol-by-symbol detector.

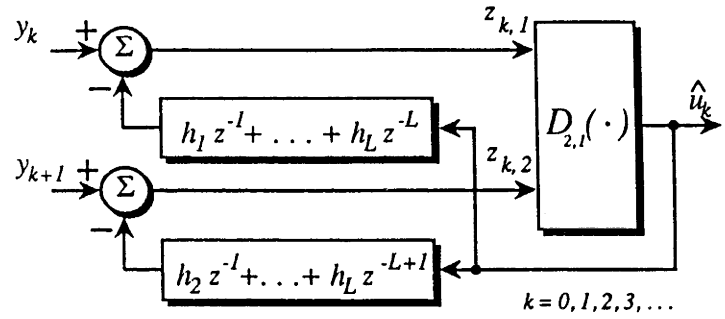

Figure 3-3: Block diagram of a (2,1)-DFE.

This theorem is proved in appendix C.1. We have assumed, for design purposes, that E_k is independent of both U_k and N_k. Note that the above criterion involves a search over 2^q candidates U, each of which involves a sum of 2^{p-q} terms. We make some observations regarding this result.

Remarks

1. If $q = p$ we have Theorem 3.2.1, as the notation suggests. Unlike Theorem 3.2.1, (3.2.20) is not equivalent to minimising an l_2-norm (Euclidean distance) for $q < p$. Computationally, (3.2.20) is less attractive than (3.2.19).

2. Equation (3.2.20) is not invariant to scaling of the noise amplitude (except when $p = q$), *i.e.*, the optimal detector is a function of the SNR.

3. We can interpret (3.2.20) as a structure equivalent to some classical optimal symbol-by-symbol detectors coupled with decision feedback (which introduces error propagation), see section 3.4.3.

4. We refer to the block DFE using the decisions (3.2.20) as the (p, q)-DFE, and in particular, when $q = 1$, we have the *sliding-window block DFE* or $(p, 1)$-DFE.

Fig.3-3 shows the structure of a (2,1)-DFE.

We indicate briefly the form of the block DFE when *recursive* (ARMA) channels, captured by (3.2.1), are to be equalized. The optimal decision function (also called a vector quantizer) $\mathcal{D}_{p,q}(\cdot)$ given in (3.2.20) remains the same, except we need to use the general D matrix in (3.2.3) rather than the special case (3.2.12). Strictly

speaking, what we have is state estimation feedback rather than decision feedback. Hence, in an implementation, we need to take the decisions $\{\widehat{u}_k\}$ and reconstruct the state estimate $\widehat{x}_k$. From (3.2.1) this is easily accomplished recursively by

$$\widehat{x}_{k+1} = A\widehat{x}_k + b\widehat{u}_k \tag{3.2.21}$$

which covers the case $q = 1$. A block processing version of (3.2.21) may be used for $q > 1$ to generate $\widehat{x}_{k+q}$; then the next vector quantizer input is given by (see Fig.3-1)

$$\begin{aligned} Z_{k+q} &= Y_{k+q} - H\widehat{x}_{k+q} \\ &= Y_{k+q} - HF_q\widehat{x}_k - HG_q\widetilde{U}_k \end{aligned} \tag{3.2.22}$$

from (3.2.5) where $\widetilde{U}_k$ is the block of q decisions, $\widehat{x}_k$ is the previous state estimate, and F_q and G_q are the q (rather than p) dimensional analogues of the block processing matrices (3.2.3). This style of generalization of decision feedback to state estimation feedback can also be found in [10] and [44].

3.2.6 High Signal-to-Noise Ratio Behaviour

The next result examines the *asymptotic* behavior of the decision function $\mathcal{D}_{p,q}(\cdot)$ in the case of high signal-to-noise ratios. We refer to this structure as the high SNR (p,q)-DFE, or the p-input high SNR block DFE when $q = 1$.

Theorem 3.2.3 (High SNR (p,q)-DFE) *The (p,q)-block maximum* a posteriori *decision procedure* (3.2.20) *for $k = 0, q, 2q, \ldots$ with $q \leq p$, asymptotically satisfies*

$$\mathcal{D}_{p,q}(Z_k) \sim [I_q \mid 0\,] \times \underset{U \in \mathbb{B}^p}{\arg\min} \{\|Z_k - DU\|^2\} \in \mathbb{B}^q \quad \text{as } \sigma \to 0 \tag{3.2.23}$$

where σ^2 is the noise variance, I_q is the $q \times q$ identity matrix, 0 denotes a $q \times (p-q)$ matrix of zeros, and D is given in (3.2.3).

A proof may be found in appendix C.2. So, in fact, $\mathcal{D}_{p,q}(\cdot)$ is obtained by taking the first q components of the $\mathcal{D}_{p,p}(\cdot)$ decision function (3.2.19) whenever the signal-to-noise ratio is sufficiently high. This result is of practical significance because a high

SNR block DFE is computationally much simpler than its optimal counterpart—only simple operations (add, multiply, compare and select) are needed in the decision circuit. In fact, for a number of studies performed, the high SNR block DFE incurs only a marginal performance loss with respect to the optimum (section 3.3.5).

This asymptotically optimal structure is equivalent to "System 1" developed in [43] when the channel is FIR and $q = 1$. We make the point here that the "tree-search" approaches of [14] and [13] assume the same quadratic cost function as [43] (equivalent to (3.2.23)) and, therefore, all three are equivalent. We delay further comparison until section 3.4.2.

3.3 Implementation Examples

So far we supplied a general definition of the decision function $\mathcal{D}(\cdot)$ in (3.2.7) and determined its optimal form $\mathcal{D}_{p,q}(\cdot)$ according to a maximum *a posteriori* criterion (3.2.19) and (3.2.20), including the high SNR case (3.2.23). We now attempt to make these decision procedures more concrete by determining explicitly the decision functions for low order cases on finite impulse response channels. We shall also present simulations which demonstrate the performance of the block DFE with the optimal and high SNR decision functions.

3.3.1 Conventional DFE

In the one-input case, $p = 1$, we have from (3.2.14)-(3.2.16) and (3.2.19) that

$$\begin{aligned}\widehat{u}_k &= \arg\min_{u\in\mathcal{B}} \Big\{(u_k + \sum_{j=1}^{L} h_j e_{k-j} + n_k - u)^2\Big\} \\ &= \arg\min_{u\in\mathcal{B}} \Big\{(y_k - \sum_{j=1}^{L} h_j \widehat{u}_{k-j} - u)^2\Big\}.\end{aligned}$$

That is

$$\widehat{u}_k = \operatorname{sgn}(y_k - \sum_{j=1}^{L} h_j \widehat{u}_{k-j}) \tag{3.3.1}$$

and so a (p, q)-DFE with $p = q = 1$ using symbol-by-symbol maximum *a posteriori* decisions (3.2.19) is just a conventional DFE.

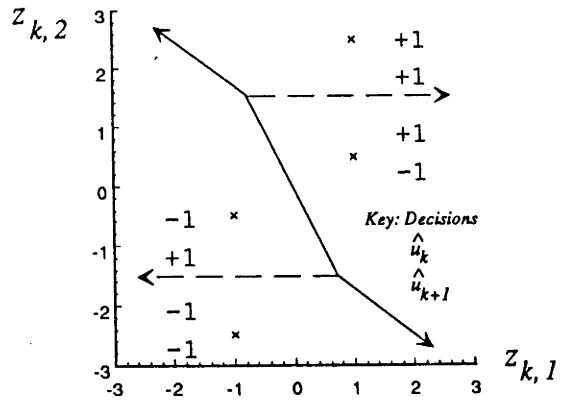

Figure 3-4: Decision regions for (2,2)-DFE, $h_1 = 1.5$.
The decision boundaries and corresponding decisions for the high SNR (2,1)-DFE may be obtained by omitting the dashed lines in the figure and taking the first component of the (2,2)-DFE decision.

3.3.2 Two-Input High SNR Block DFE

We consider firstly the decision procedure $\mathcal{D}_{2,2}(\cdot)$ for the (2,1)-DFE in the high signal-to-noise ratio case (3.2.23), and save the optimal realisation (3.2.20) for the next section. A realisation example for the (2,2)-DFE may be found in [15]. Since $p = 2$, the decision function $\mathcal{D}_{p,q}(\cdot)$ has two inputs and one binary output. Assuming the past L decisions are correct, noise is absent and that $h_0 = 1$, the components of Z_k in (3.2.19) are

$$z_{k,1} = u_k \;\; ; \;\; z_{k,2} = h_1 u_k + u_{k+1},$$

so that the four possible noiseless, error-free values of Z_k are

$$Z_{++} = \begin{bmatrix} +1 \\ +1+h_1 \end{bmatrix} ; \; Z_{+-} = \begin{bmatrix} +1 \\ -1+h_1 \end{bmatrix} ;$$

$$Z_{-+} = \begin{bmatrix} -1 \\ +1-h_1 \end{bmatrix} ; \; Z_{--} = \begin{bmatrix} -1 \\ -1-h_1 \end{bmatrix}. \qquad (3.3.2)$$

The decision function has as its input the corrupted signal Z_k (3.2.16). It computes the Euclidean distance between Z_k and each of the above four points, assigning a binary decision according to the minimum distance rule: $\widehat{u}_k = -1$ if Z_{-+} or Z_{--} is closest to Z_k, $\widehat{u}_k = +1$ if Z_{++} or Z_{+-} is closest to Z_k. In this way Z_k-

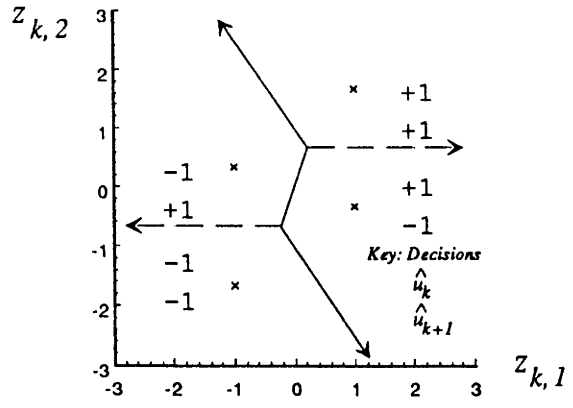

Figure 3-5: Decision regions for (2,2)-DFE, $h_1 = 2/3$.
The decision regions for the high SNR (2,1)-DFE may be obtained by omitting the dashed lines in the figure.

space is partitioned into two (polygonal) regions, each with an associated binary decision. The *decision boundary* is the set of points on the boundary of the two decision regions. Fig.3-4 shows the decision boundary for $h_1 = 1.5$ (representative of $h_1 > 1$), and Fig.3-5 for $h_1 = 2/3$ (representative of $0 < h_1 < 1$). Note that the decision boundary for a scalar DFE (3.3.1) is simply the line $z_{k,1} = 0$. Simple geometrical arguments lead to the explicit formula for the decision device output which we state below without proof.

Lemma 3.3.1 *The high SNR (2,1)-DFE operating on a finite impulse response channel with unit cursor $h_0 = 1$ has decisions given for $k = 0, 1, 2, \ldots$ by*

$$\widehat{u}_k = \text{sgn}\left\{\text{sgn}(l_1) + \text{sgn}(l_2) + \text{sgn}(l_3)\right\} \tag{3.3.3}$$

where

$$\begin{aligned} l_1 &\triangleq h_1 z_{k,2} + z_{k,1} - h_1 \\ l_2 &\triangleq h_1 z_{k,2} + z_{k,1} + h_1 \\ l_3 &\triangleq z_{k,1} + (h_1 - \text{sgn}(h_1)) z_{k,2}. \end{aligned}$$

Here $l_i = 0$, $i = 1, 2, 3$ are the equations of the straight lines comprising the decision boundary. The above result shows that, like a conventional DFE, a two-input high SNR block DFE can be implemented using only tapped delay lines and hard

limiters. We note that similar expressions exist for the decisions formed by the (optimal) (2,2)-DFE, and these may be found in [15]. Due to the assumption of correct past decisions, the decision function in Lemma 3.3.1 depends on the impulse response parameters h_0 and h_1. For a general (p,q)-DFE, the decision laws depend only on $\{h_k : 0 \leq k \leq p-1\}$. Loosely speaking, this means the decision procedure for the block DFE effectively utilizes information carried not just by the cursor but by the first p impulse response values. However, for $p > 2$ the explicit form of the decision laws is very difficult to obtain (see section 3.3.4).

3.3.3 Two-Input Sliding-Window Block DFE

The decision boundary for the *optimal* sliding-window block DFE (or decision boundaries for the (p,q)-DFE in Theorem 3.2.2) depend not only on the first p channel parameters, but also on the *signal-to-noise ratio*, and are generally *curved*, as we will see in this and the following example. We consider a FIR channel whose first two coefficients are $h_0 = 1$ and $h_1 = 0.5$ for this example. The (2,1)-DFE computes its decisions according to

$$\widehat{u}_k = \underset{u\in\{-1,+1\}}{\arg\max}\ g(z_1, z_2, u; h_0, h_1, \sigma), \tag{3.3.4}$$

where

$$\begin{aligned} g(z_1, z_2, u; h_0, h_1, \sigma) \ &\triangleq\ \exp\left\{\frac{(z_1 - h_0 u)^2 + (z_2 - h_1 u - h_0)^2}{-2\sigma^2}\right\} \\ &+\ \exp\left\{\frac{(z_1 - h_0 u)^2 + (z_2 - h_1 u + h_0)^2}{-2\sigma^2}\right\}, \end{aligned} \tag{3.3.5}$$

and z_1 and z_2 are the components of the decision device input vector Z_k. For a given channel and signal-to-noise ratio (which fixes the noise variance σ^2), we choose a value for z_2 and solve the following non-linear equation for z_1

$$g(z_1, z_2, -1; h_0, h_1, \sigma) - g(z_1, z_2, +1; h_0, h_1, \sigma) = 0, \tag{3.3.6}$$

using a Newton-Raphson iterative method. The set of points (z_1, z_2) satisfying (3.3.6) defines the decision boundary for the (2,1)-DFE. Fig.3-6 shows the resulting

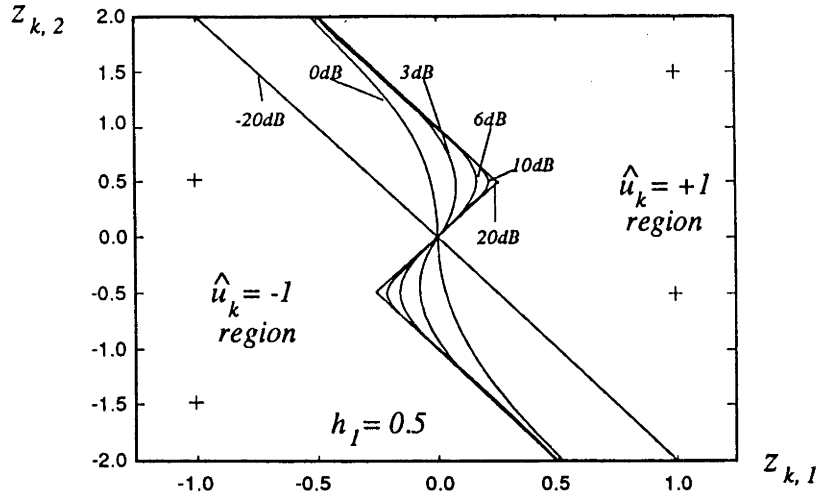

Figure 3-6: Decision boundaries for (2,1)-DFE.

decision boundaries in Z_k-space for a range of signal-to-noise ratios. Note that as the SNR increases the regions become the same as those of Theorem 3.2.3 (similar to Fig.3-4), *i.e.*, the boundary is effectively piece-wise linear for SNRs above about +20dB. It is also apparent that in the low SNR extreme (≤ -20dB), the decision boundary reduces to a straight line through the origin. This latter property is easily proved, but is of little practical significance and will not be elaborated.

3.3.4 Three-Input High SNR Block DFE Example

As the dimension of the block DFE's decision device increases, it becomes prohibitively difficult to find explicit expressions for the decision procedure. This is true even in the high SNR limit, where the boundary is composed of (hyper)planes. The lack of such information hinders analysis, as we will see in chapter 4. We illustrate this difficulty here by giving two visualisations of decision boundaries for the three-input, sliding-window, high SNR block DFE (3.2.23). The minimum distance decision procedure takes the first component u_1 of the vector $[u_1, u_2, u_3]'$ which achieves the minimum

$$\min_{[u_1,u_2,u_3]\in I\!B^3} \left\| \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ h_1 & 1 & 0 \\ h_2 & h_1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \right\|^2, \tag{3.3.7}$$

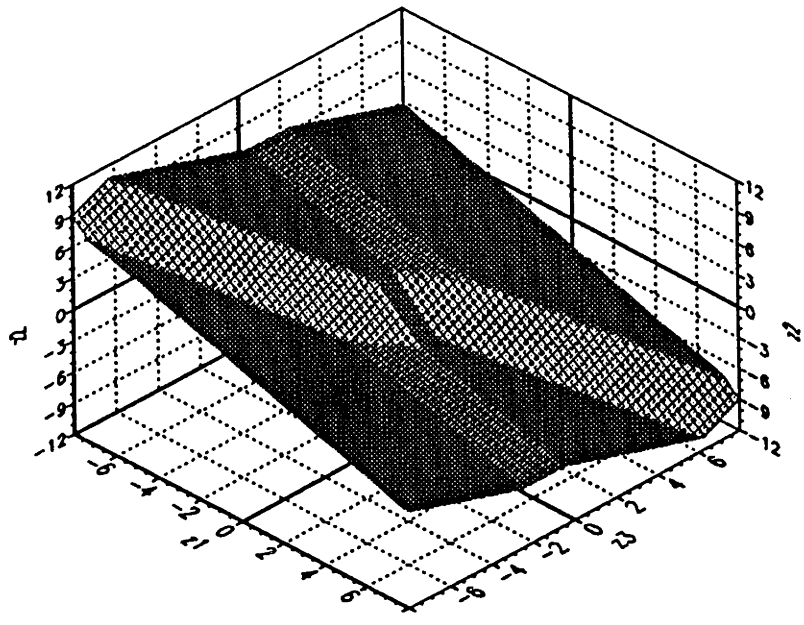

Figure 3-7: Decision surface for high SNR $(3,1)$-DFE, $h_1 = 1.5,\ h_2 = 1$.

where $||\cdot||$ is the Euclidean norm of a real vector, and the decision device input is $Z_k = [z_1, z_2, z_3]'$. We can obtain a mesh picture of the decision boundary surface for a given channel in the following way. We take a two-dimensional grid of points in the (z_1, z_3)-plane (say). For each grid point, we perform a binary search over a prescribed range of z_2 values, thus homing in on the point at which the decision changes sign. We are assuming, of course, that there is just one such change of sign for each grid point. Fig.3-7 and Fig.3-8 show the form of the decision surface obtained for the channels $[h_0 = 1, h_1 = 1.5, h_2 = 1]$ and $[h_0 = 1, h_1 = 0.5, h_2 = 1]$ respectively. In both cases, the decision surface consists of 11 planar sheets, eight of which are semi-infinite. Clearly, the normal to each plane is determined by a pair of points from the set of eight points $\{DU \mid U \in \mathbb{B}^3\}$ with D as in (3.2.12), but discovering a formula akin to (3.3.3) for the decision surface remains an open problem. We turn in the next section to consider of the bit error rate performance of block DFEs, obtained by numerical simulation.

3.3.5 Performance Comparisons

In the preceding sections we detailed the theoretical development of a family of block DFEs. Our aim in this section is to give numerical performance examples of these

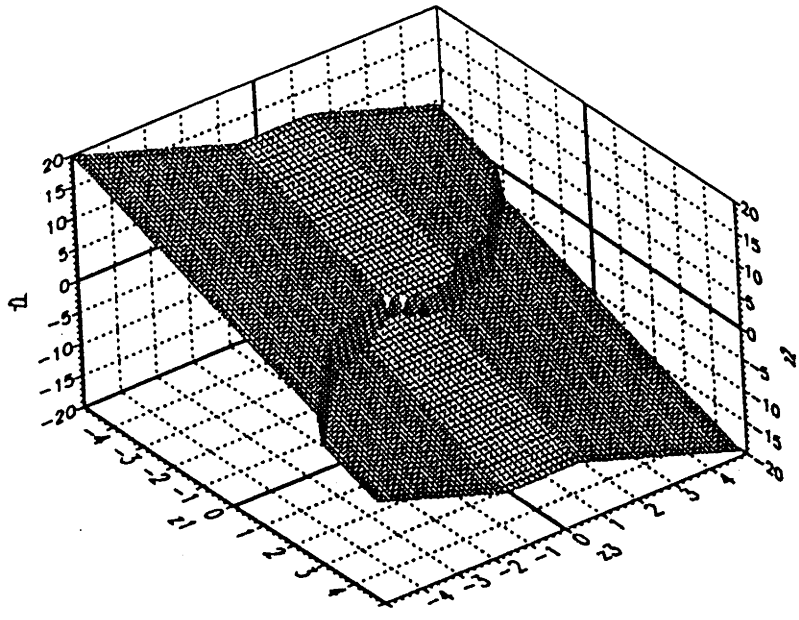

Figure 3-8: Decision surface for high SNR $(3,1)$-DFE, $h_1 = 0.5,\ h_2 = 1$.

equalizers and compare the results with those of the decision feedback equalizer and the Viterbi decoder. We will see that by varying the block size p, the block DFE's performance can range between these two extremes, and that even relatively small values of p give considerable improvement over the DFE. Moreover, we will observe that the replacement of the optimal block DFE's decision device by its high SNR manifestation (3.2.23) has only a small adverse effect on performance.

Following [11], we consider the $(1 + z^{-1})^2$ binary partial response signalling channel[6] which is known to guarantee poor performance for the conventional DFE. The channel output is given by

$$y_k = u_k + 2u_{k-1} + u_{k-2} + n_k \tag{3.3.8}$$

where $\{u_k\}$ is a binary *iid* input data sequence, and n_k is a zero-mean white Gaussian noise sequence. The (p,q)-DFE (3.2.20) and the high SNR (p,q)-DFE (3.2.23) were simulated for various combinations of p and q. Fig.3-9 shows simulation results (bit error rates versus SNR) for different block DFE realisations corresponding to the (p,q)-pairs (arranged in order of implementation complexity): $(1,1)$, $(2,2)$, $(2,1)$,

[6] Here, z^{-1} is the backward shift operator: $z^{-1}u_k = u_{k-1}$.

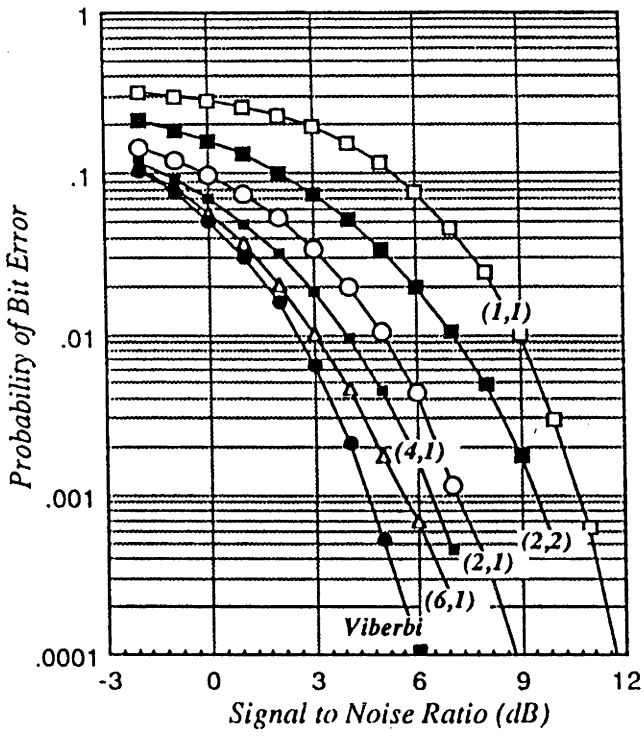

Figure 3-9: Probability of error.

$(4,1)$, and $(6,1)$ when used on the channel (3.3.8). All simulations include error propagation. The bit error rate for the maximum likelihood sequence estimator (using a Viterbi algorithm with a 30 sample delay, simulated over a minimum of 10^5 points) is also plotted in the figure, and may be used as a reference for comparison with figure 7 in [11] (whose definition of SNR differs from ours by a constant). Clearly, as the complexity increases, the block DFE has performance ranging from the conventional DFE to the MLSE. In the section 3.4 we will explain these trends in terms of the respective decision criteria.

The differences between the optimal (p,q)-DFE and the high SNR (p,q)-DFE for this channel are minimal. For this example channel the (p,q)-DFE is numerically superior to the high SNR (p,q)-DFE but only to the order of a few percent at 0dB SNR. (For other channels the difference has been observed to be closer to 5%.) This simulation substantiates the claim in [13] concerning the marginal loss in performance of high SNR approximations at least on some channels. In the next section we comment on the computational complexity of block DFE realisations, as this is one of the major factors governing the implementability an equalizer.

3.3.6 Computational Complexity

We have seen that the input dimension p of a block DFE can be increased to enhance its bit error rate performance. In this section we look at the complexity of particular block DFE realisations. By "complexity", we mean the number of operations[7] required to compute each decision output. This is really only a partial measure of the difficulty of implementation, as we are ignoring such factors as memory requirements and transfer of data during computation [64].

We restrict the present discussion to the $(p,1)$-DFE and high SNR $(p,1)$-DFE on a FIR(L) channel with binary input. The high SNR decision function (3.2.23) for $q = 1$ involves an integer programming problem of generic complexity 2^p (the number of candidate binary vectors). Let us count the maximum number of operations required in obtaining the decision $\widehat{u}_k$ (ignoring the structure of the matrices (3.2.11,3.2.12)). At the start of each block iteration k, the block DFE stores the received data block Y_k (3.2.4) and the state estimate $\widehat{x}_k$ (3.2.13). The operation count is as follows: $2pL$ operations to compute Z_k (3.2.5); $2p^2 + 2p - 1$ operations for each of the 2^p values of $||Z_k - DU||^2$, to which we add 1 for the comparison needed to find the minimum. The (maximum) total is then $2^p 2p(p+1) + 2pL$ operations for one decision. On the other hand, the optimal $(p,1)$-DFE requires (roughly) an additional $\zeta 2^p$ operations per decision, where ζ is the number of flops required to find the exponential of a real number. All this indicates that the technique is unworkable if the p parameter is too large.

A recursive tree search algorithm is discussed in [13] which is computationally competitive with the Viterbi algorithm and applicable to high SNR systems like (3.2.23). We infer that similar reduced-complexity algorithms exist for more general block DFEs (3.2.20). The alternative to such algorithms which give an exact solution of an integer optimisation problem is the investigation of approximate decision schemes requiring substantially less computation with some loss in optimality of performance. Work along these lines has been undertaken in [34, 43] with the implicit assumption of high signal-to-noise ratios. The possibility of applying functional representation techniques, such as multi-layer perceptron feedforward neural

[7] "operation" means a floating point addition or multiplication, *i.e*, a *flop*.

networks [40], to approximate the decision criterion also suggests itself. An approach along these lines has been attempted for purely feedforward equalizers in [56], but application of these ideas to block decision feedback equalizers remains an open problem. Perhaps further investigation of the block DFE's decision function may lead to a specific non-linear processing architecture as was achieved for the DFE in chapter 2. We leave these practical considerations for now and return to a theoretical discussion of the connections between the block DFE and other classical non-linear detectors.

3.4 Relationship to Classical Detection

Our aim in this section is to reinforce further the methodology of block decision feedback equalization by considering its relationship to (i) maximum likelihood sequence estimation and (ii) minimum bit error rate detectors. Throughout, we confine the discussion to binary signalling on a FIR channel with additive white Gaussian noise.

3.4.1 Viterbi Decoding

Here, we establish the relationship between the MLSE optimal block DFE, firstly by comparing decision criteria, and secondly via a trellis-based interpretation. The MLSE (using the Viterbi algorithm) determines the semi infinite sequence of estimates $\{\widehat{u}_0,\ \widehat{u}_1,\ \widehat{u}_2,\ \ldots\}$ which minimises the cost[8]

$$\mathcal{J} \triangleq \sum_{j=0}^{\infty} \left(y_j - \widehat{u}_j - \sum_{i=1}^{L} h_i \widehat{u}_{k-i} \right)^2 \tag{3.4.1}$$

in which $\{\widehat{u}_{-1},\ \widehat{u}_{-2},\ \ldots,\ \widehat{u}_{-L}\}$ defines some arbitrary initial condition.

Similarly, the $p = q$ block DFE (3.2.19) forms the length p vector of estimates $\{\widehat{u}_k,\ \widehat{u}_{k+1},\ \ldots,\ \widehat{u}_{k+p-1}\}$ that minimise the cost

$$\begin{aligned} \mathcal{J}_p \ = \ & (z_{k,1} - \widehat{u}_k)^2 + (z_{k,2} - \widehat{u}_{k+1} - h_1 \widehat{u}_k)^2 + \cdots \\ & + \cdots + (z_{p,k} - \widehat{u}_{k+p-1} - h_1 \widehat{u}_{k+p-2} - \cdots - h_{p-1} \widehat{u}_k)^2 \end{aligned}$$

[8] In reality, the message length is finite.

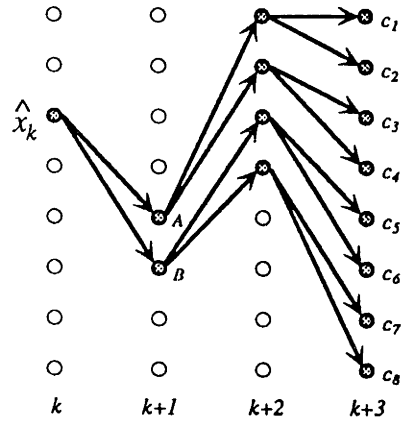

Figure 3-10: Trellis interpretation of a three-input block DFE.

which, with the aid of (3.2.16), may be re-expressed as

$$\mathcal{J}_p \triangleq \sum_{j=k}^{k+p-1} \left(y_j - \widehat{u}_j - \sum_{i=1}^{L} h_i \widehat{u}_{k-i} \right)^2. \tag{3.4.2}$$

By comparison with (3.4.1), we see that $\mathcal{J}_p$ is essentially identical to a p-length window of the MLSE cost (3.4.1), with the exception that: (i) the past L decisions $\{\widehat{u}_{k-1}, \widehat{u}_{k-2}, \ldots, \widehat{u}_{k-L}\}$ in (3.4.2) act as initial conditions at time k and are not necessarily the same as the corresponding terms in (3.4.1); and (ii) there is a truncation to p terms in (3.4.2) rather than the infinite number of terms in (3.4.1). The MLSE and (p, q)-DFE with $q = p$ will generate identical estimates given matching initial conditions (at time k) and letting $p \to \infty$. We delay comparison with the classical non-linear MAP probability detector until section 3.4.3.

3.4.2 Trellis Interpretation

The Viterbi algorithm is easily visualised in terms of a 2^L-state trellis with associated quadratic metric (see description in appendix A.1). In this subsection we give trellis-based interpretations of the optimal block DFE decision procedures (3.2.19), (3.2.20), and (3.2.23) assuming FIR channels. (We will see that for the block DFE the metrics need not be quadratic.) In this setting, a clear comparison between the (p, q)-DFE and the methods in [13, 14] is possible. We illustrate the decision rules for block length $p = 3$.

At time k, having L past decisions $\widehat{x}_k$ in (3.2.13) corresponds to fixing a state in

the trellis of Fig.3-10. For binary signalling there are only two possible transitions from each state. We have shown the 8 possible trajectories from $\widehat{x}_k$ up to time $k+3$. These form a tree structure with starting node $\widehat{x}_k$. Each trajectory corresponds to a different $\widehat{U}_k \in I\!B^3$ and has an associated total "cost" c_i given, in the notation of section 3.2.4, by $\|Z_k - D\widehat{U}_k\|^2$. We remark for clarity that this sum of squares cost arises naturally in Viterbi decoding, the (p,p)-DFE and the high SNR block DFE.

The interpretation of (3.2.19) is that we select the 3 step path (and hence those 3 decisions) that gives the minimum cost among $\{c_i : i = 1,\ldots,8\}$. The recursion then proceeds by advancing from k to $k+3$ and we start afresh. Note that the conventional DFE would choose the *one* step trajectory (A or B in Fig.3-10) with the least *incremental* cost, as given by (3.3.1).

The high SNR decision rule (3.2.23) is obtained by taking the least cost 3 step path, in the same way as (3.2.19), but retaining only the first $q \in \{1,2\}$ steps. The recursion advances from k to $k+q$. Thus (3.2.23) is seen to be a receding horizon strategy, borrowing the terminology from optimal control theory [65], *i.e.*, only a truncated version of the full optimal path (generated from (3.2.19)) is actually implemented. Under this framework one can see that the approaches described in [14, 13] are equivalent to "System 1" developed in [43], and all three are identical in action to a high SNR $(p,1)$-DFE.

We now proceed to the optimal (p,q)-DFE ($q < p$), and develop a trellis-based interpretation of the cost (3.2.20). Suppose $q = 1$ in Fig.3-10. The optimum (3,1)-DFE must decide between the one step paths leading to A and B. To do this, it computes the 2^q respective aggregated costs

$$c_A = \sum_{i=1}^{4} e^{-c_i/2\sigma^2} \quad , \quad c_B = \sum_{i=5}^{8} e^{-c_i/2\sigma^2}$$

and takes the maximum, thus determining $\widehat{u}_k$. Whereas the tree-search algorithms of [13, 14] are essentially versions of (3.2.23), developed using a suboptimum metric, the optimal block DFE metric (3.2.20) in fact corresponds to the non-linear function of quadratic costs described above.

3.4.3 Minimum Bit Error Rate Detectors

In the last section we saw that the (p,p)-DFE (with correct initial conditions) becomes an MLSE in the limit as $p \to \infty$. We now look at two classical "minimum probability of error" detection criteria and examine the conditions under which the (p,q)-DFE is recovered. The block DFE will be seen to have a central place among these detection methods. We let K denote the total number of transmitted symbols and continue using our notation. The reader is referred to [27, 30, 31] for details on optimum non-linear receivers, and to [54] for a concise review of the relevant literature.

The two detection strategies which, as we will show, are closely related to the block DFE are: (i) the sequential symbol-by-symbol optimum detector [30, 31] which minimises the error in detecting each symbol based on the entire received sequence, that is

$$\arg\max_{u_k \in \mathcal{B}} Pr(u_k \mid y_0,\ y_1,\ \ldots,\ y_{K-1}); \tag{3.4.3}$$

and (ii) the detector [26] resulting from (i) when a fixed decision-delay constraint is imposed, with criterion

$$\arg\max_{u_k \in \mathcal{B}} Pr(u_k \mid y_0,\ y_1,\ \ldots,\ y_{k+p-1}). \tag{3.4.4}$$

Recursive implementations exist for both of the above detectors.

In order to clarify the connection between (3.4.4) and the (p,q)-DFE when $q = 1$, we now adopt the decision feedback strategy $x_k = \hat{x}_k$, where x_k is the state of the FIR channel given by (3.2.10). Noting that u_k does not appear explicitly in $y_0, \ldots, y_{k-1}$ (by causality), and that the information these observations convey about the past inputs relevant to deciphering the value of u_k is subsumed by the state, we are led to consider a modification of the MAP criterion (3.4.4) which incorporates decision feedback

$$\begin{aligned} \arg\max_{u_k \in \mathcal{B}} Pr(u_k \mid y_0,\ \ldots,\ y_{k-1},\ \underbrace{y_k,\ \ldots,\ y_{k+p-1}}_{Y_k},\ \underbrace{x_k = \hat{x}_k}_{E_k = 0}) \\ = \arg\max_{u_k \in \mathcal{B}} Pr(u_k \mid Z_k,\ E_k = 0) \end{aligned}$$

$$= \mathcal{D}_{p,1}(Z_k) \tag{3.4.5}$$

where $Z_k \triangleq Y_k - H\hat{x}_k$ from (3.2.5). Therefore, the $(p,1)$-DFE (3.4.5) arises from (3.4.4) with the assumption of correct past decisions (decision feedback). In contrast to the tree-search algorithm [13], this derivation, which is easily generalised to the (p,q)-DFE for $q > 1$, is not dependent on the signal-to-noise ratio.

Examination of (3.4.5) leads to the conclusion that the $(p,1)$-DFE decision is asymptotically equivalent (as $p \rightarrow$ message length K) to the optimum symbol-by-symbol detector (3.4.3), given matching initial conditions and in the absence of past errors.

3.5 Extensions

3.5.1 Non-Linear Channels

Modelling the transmission channel as a linear filter with additive Gaussian noise (3.2.9) may not be adequate for some physical channels. For example, telephone lines suffer from non-linear distortions of varying degrees [20]. Traditionally, in the finite impulse response case where the state is a vector of past inputs, the receiver design entails the expansion of the noiseless y_k in a Volterra series [49]. This series is truncated, and a non-linear tapped delay line equalizer is developed using a minimum mean-square error (MMSE) criterion. We now investigate the applicability of block decision feedback equalization ideas to the equalization of non-linear channels. The channel model is assumed known at the outset, and we develop a suitable block DFE structure. Formal proofs are omitted as we aim only to convince the reader of the feasibility of the approach.

Suppose we have a causal non-linear finite-dimensional single-input, single-output system described in state-space by the *non-linear* equations

$$\begin{aligned} x_{k+1} &= f_1(x_k, u_k) \\ y_k &= g_1(x_k, u_k) + n_k, \quad k = 0, 1, 2, \ldots, \end{aligned} \tag{3.5.1}$$

for some deterministic functions $f_1(\cdot)$ and $g_1(\cdot)$, where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{B}$, $y_k \in \mathbb{R}$, and $n_k \sim N(0, \sigma^2)$. Furthermore, we assume that the sequence u_k is *iid* and equiprobable. Such a model could describe the output y_k of a finite-memory non-linear channel with additive white noise n_k fed by an input sequence u_k.

With the intention of arriving at a block processing description of the system, we form successive states and outputs by composition in the first argument

$$\begin{aligned} x_{k+2} &= f_1(x_{k+1}, u_{k+1}) = f_1(f_1(x_k, u_k), u_{k+1}) \\ &\triangleq f_2(x_k, u_k, u_{k+1}) \\ y_{k+1} &= g_1(x_{k+1}, u_{k+1}) + n_{k+1} = g_1(f_1(x_k, u_k), u_{k+1}) + n_{k+1} \\ &\triangleq g_2(x_k, u_k, u_{k+1}) + n_{k+1}. \end{aligned} \tag{3.5.2}$$

Proceeding analogously, after $(p-1)$ such steps, we obtain

$$\begin{aligned} x_{k+p} &= f_p(x_k, u_k, \ldots, u_{k+p-1}) = f_p(x_k, U_k'), \quad k = 0, p, 2p, \ldots, \\ Y_k &= \begin{bmatrix} g_1(x_k, u_k) \\ g_2(x_k, u_k, u_{k+1}) \\ \vdots \\ g_p(x_k, u_k, \ldots, u_{k+p-1}) \end{bmatrix} + N_k \triangleq G_p(x_k, U_k') + N_k, \end{aligned} \tag{3.5.3}$$

where U_k, Y_k and N_k are defined in section 3.2.1, and $f_i(\cdot)$ and $g_i(\cdot)$ are defined for $i = p, p-1, \ldots, 2$ by the backward recurrences

$$\begin{aligned} f_i(x_k, u_k, \ldots, u_{k+i-1}) &= f_{i-1}(f_{i-1}(x_k, u_k, \ldots, u_{k+i-2}), u_{k+i-1}) \\ g_i(x_k, u_k, \ldots, u_{k+i-1}) &= g_{i-1}(f_{i-1}(x_k, u_k, \ldots, u_{k+i-2}), u_{k+i-1}), \end{aligned}$$

with $f_1(\cdot)$ and $g_1(\cdot)$ as in (3.5.1). Thus, assuming the same initial conditions, the model (3.5.3) generates the same outputs as (3.5.1), but in blocks of size p.

In the philosophy of the block DFE development for a linear channel, we suppose a state estimate $\hat{x}_k$ is available at time k. How can we design a p-input, p-output decision device for this system? The non-linearity of $f_p(\cdot)$ means that we cannot generally write Y_k as a sum of block cursor, ISI and noise terms, as we did in section 3.2.2. However, we can gain insight by reconsidering the (p,p)-DFE criterion $\mathcal{D}_{pp}(\cdot)$

(3.2.19). Recall that this decision criterion selects that $\widehat{U}_k \in \mathbb{B}^p$ which minimises the squared magnitude of the vector $Z_k - D\widehat{U}_k$. First rewrite this using (3.2.5) and (3.2.6) as

$$Z_k - D\widehat{U}_k = Y_k - H\widehat{x}_k - D\widehat{U}_k = Y_k - \widehat{Y}_k,$$

where $\widehat{Y}_k = H\widehat{x}_k + D\widehat{U}_k$ is the estimated noiseless block channel output. We see immediately that the same objective may be achieved in the non-linear channel case by choosing $\widehat{U}_k$ according to

$$\widehat{U}_k = \underset{U_k \in \mathbb{B}^p}{\arg\min} \|Y_k - G_p(\widehat{x}_k, U_k')\|^2, \ k = 0, p, 2p, \ldots, \tag{3.5.4}$$

and these decisions can be used to generate the next state estimate via

$$\widehat{x}_{k+p} = f_p(\widehat{x}_k, \widehat{U}_k').$$

A $(p,1)$-DFE for the non-linear channel can be realised as in section 3.2.5, using the same design assumptions as before, namely: the input sequence u_k is *iid* and equiprobable; and $E_k \triangleq x_k - \widehat{x}_k$ is independent of U_k and N_k at the design stage. We thus arrive at the sliding-window decision criterion for a non-linear channel,

$$\widehat{u}_k = \underset{u \in \mathbb{B}}{\arg\min} \sum_{V \in \mathbb{B}^{p-1}} e^{-\frac{1}{2\sigma^2}\|Y_k - G_p(\widehat{x}_k, [\begin{smallmatrix} u \\ V \end{smallmatrix}]')\|^2}, \ k = 0, 1, 2, \ldots, \tag{3.5.5}$$

with Y_k and $G_p(\cdot)$ defined in (3.5.3). We can use

$$\widehat{x}_{k+1} = f_1(\widehat{x}_k, \widehat{u}_k)$$

to generate successive state estimates. The proof that this result holds follows the same argument as the proof of Theorem 3.2.2, and is therefore not supplied. We note here that the block DFE for a non-linear channel has a non-linear vector quantizer and a *non-linear* filter in its feedback path. The non-linear feedback filter, which generates $G_p(\widehat{x}_k, U_k')$ from the current state and block input estimates, could be realised with a look-up table (random access memory) if (3.5.1) has only a finite number of possible states x_k. This is the case if the channel has finite memory. The reader is referred to [50] for more examples of feedforward equalizers for non-

linear channels.

We continue describing possible extensions to block decision feedback equalization in the next section, treating the case of quadrature amplitude modulation, and giving some examples of maximum likelihood decision device realisations for M-ary signalling.

3.5.2 Quadrature Amplitude Modulation

We first introduce the idea of an M-ary signalling alphabet. This is a finite set of symbols of the form

$$\mathcal{M} = \{-M+1, -M+3, \ldots, -3, -1, 1, 3, \ldots, M-3, M-1\}, \tag{3.5.6}$$

where M is an even positive integer. For example, with $M = 2$, we recover the binary signalling set. Physical channels are two-dimensional and can support transmission of information on two carriers that are 90° out of phase. Such an arrangement is called *quadrature amplitude modulation* (QAM). The symbol set for QAM, called a *constellation*, is often a square grid of points in the complex plane $\mathbb{C}$ with centre of mass at the origin. For instance, a signalling set for 4-QAM is[9] $\{e^{j\pi/4}, e^{j3\pi/4}, e^{j5\pi/4}, e^{j7\pi/4}\}$. If all symbols in the QAM constellation have the same magnitude, then the signalling scheme is termed phase-shift keying (or n-PSK, where there are n symbols). The previous example could therefore equally well be called 4-PSK. For further discussion of QAM techniques, see [20].

In order to model both in-phase and quadrature channels, we adopt a channel model with *complex* coefficients. This approach has the advantage that the algebra is symbolically the same as in the real case. Preserving the notation of section 3.2.1, we generate the complex channel output y_k in response to the complex input sequence u_k via

$$\begin{aligned} x_{k+1} &= Ax_k + bu_k \\ y_k &= cx_k + du_k + n_k, \end{aligned} \tag{3.5.7}$$

[9] We use $j = \sqrt{-1}$.

where $x_k \in \mathbf{C}^n$, $\{A, b, c, d\}$ is a complex realisation and $n_k = n_k^R + jn_k^I$ is a complex Gaussian white noise process whose independent components satisfy $n_k^R, n_k^I \sim N(0, \sigma^2)$ and have identity covariances. The input signal $u_k = u_k^R + ju_k^I$ is such that $u_k^R, u_k^I \in \mathcal{M}$, so that $u_k \in \mathcal{Q}$, where $\mathcal{Q} \triangleq \{m_1 + jm_2 | m_1, m_2 \in \mathcal{M}\}$ is a square QAM constellation. The same block processing realisation (3.2.2) holds, recognising that the matrices $\{F, G, H, D\}$ now have complex elements.

We now focus our attention on the form of the decision device. Assuming, as before, that both components of the input sequence u_k^R and u_k^I are *iid* and equiprobable, we can develop maximum *a posteriori* (or equivalently maximum likelihood) vector quantizers for the full-blocking and sliding-window block DFEs. The reasoning for the binary case carries over, the optimum decision device for the (p, p)-DFE being a "slicer" or nearest-neighbour quantizer, computing

$$\widehat{U}_k = \underset{U \in \mathcal{Q}^p}{\arg\min} \|Z_k - DU\|^2, \tag{3.5.8}$$

in which $\mathcal{Q}^p$ denotes the set of vectors of dimension p with components in the QAM constellation $\mathcal{Q}$, and $\|X\|^2 = X^*X$ where X^* is the conjugate transpose of the vector X. As before, $Z_k = Y_k - H\widehat{x}_k$, in which $\widehat{x}_k$ denotes the (complex) channel state estimate obtained via (3.2.21). Similarly, as expected, the vector quantizer for a $(p, 1)$-DFE uses (3.2.20), replacing $\mathcal{B}$ by $\mathcal{Q}$ (and invoking the usual design assumptions of independence).

We now give a concrete example of a decision procedure. We choose, for ease of visualisation, a high SNR (2,1)-DFE for quaternary ($M = 4$) signalling on a real FIR channel with first two impulse response coefficients $h_0 = 1$ and $h_1 = 0.3$. As in the binary case, in the high SNR limit, the quantizer reduces to a minimum Euclidean distance metric, selecting its estimates as

$$\widehat{u}_k = [1,\ 0] \times \underset{u_1, u_2 \in \mathcal{M}}{\arg\min} \left\| \begin{bmatrix} z_{k,1} \\ z_{k,2} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ h_1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \right\|^2, \tag{3.5.9}$$

$\mathcal{M}$ denoting the set $\{-3, -1, +1, +3\}$. The two-dimensional Z_k-space is partitioned into 4 regions (Fig.3-11), corresponding to the 4 alphabet symbols. The decision boundaries consist of straight line segments which perpendicularly bisect pairs

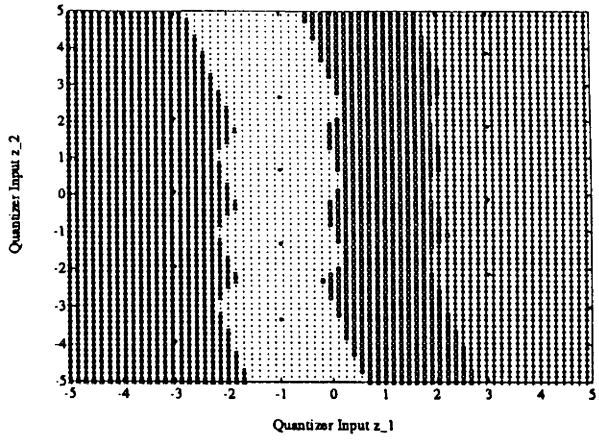

Figure 3-11: Decision regions for quaternary signalling.
From left to right, the respective regions correspond to the decisions $\widehat{u} = -3,\ -1,\ 1,\ 3$.

of points (displayed in the figure as "*") from the set

$$\left\{ \begin{bmatrix} 1 & 0 \\ 0.3 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} ;\ u_1, u_2 \in \mathcal{M} \right\}.$$

We mention in passing that a quaternary decision feedback equalizer would use a slicer with charteristic [21]

$$\mathrm{sgn}(z-1) + \mathrm{sgn}(z) + \mathrm{sgn}(z+1),$$

where z is the scalar input to the slicer. It should be clear that the block DFE framework is well suited to the incorporation of quadrature amplitude modulated signals.

3.5.3 Coloured Noise

In the sampling operation that occurs at the front end of an optimal receiver, a matched filter is used to ensure that no information is lost. Supposing the channel noise is initially white, after the filtering it will generally be coloured. A whitening filter [3] can be used to remove this colouration, further distorting the signal. This is a necessary measure in Viterbi decoding, since the minimum distance metric assumes whiteness. On the other hand, in block decision feedback equalization, we recognise that the problem of designing practical decision criteria necessitates some

(small) loss in optimality. We have made certain "weak independence" assumptions in the design of the (p,p)-DFE, using the approximation

$$Pr(Z_k \mid E_k = 0, U_k = U) = p_N(Z_k - DU). \tag{3.5.10}$$

In the $(p,1)$-DFE design, we assumed that E_k was independent of U_k and N_k. These steps, although not rigorous (since weak correlations do arise through the use of decision feedback), result in practical detection strategies and should be seen as a compromise between implementability and optimality. With these remarks in mind, we now consider what form the optimal decision function for the block DFE could take in the presence of additive *coloured* Gaussian noise.

If there is correlation in the noise sequence, then the validity of assuming the independence of E_k and N_k will be weakened. Nonetheless, we can attempt to incorporate the colouration of the sampled noise process into the design in the following manner. We present this at a tutorial level only.

Denote by Σ the autocorrelation matrix of the noise sequence with elements σ_{ij} given by

$$\sigma_{ij} = \mathcal{E}\{n_{k+i-1} n_{k+j-1}\}. \tag{3.5.11}$$

The vector of noise samples N_k will then have an (invertible) $p \times p$ autocorrelation matrix Σ_p (the upper left $p \times p$ submatrix of Σ) which is the same for all such N_k (assuming stationarity). The multivariate probability density of N_k is

$$p_N(N_k) \triangleq \frac{1}{(2\pi)^{p/2}|\Sigma_p|^{\frac{1}{2}}} e^{-\frac{1}{2}N_k' \Sigma_p^{-1} N_k}, \tag{3.5.12}$$

where $|\cdot|$ represents the determinant of a square matrix. We modify the (p,p)-DFE criterion by weighting the distance measure by Σ_p^{-1}

$$\widehat{U}_k = \underset{U_k \in I\!B^p}{\arg\min} \left\{ \|Z_k - DU_k\|^2_{\Sigma_p^{-1}} \right\}, \tag{3.5.13}$$

where the vector norm $\|X\|^2_Y = X'YX$ for some positive definite Y. We can do the

same with the $(p,1)$-DFE criterion

$$\widehat{u}_k = \arg\max_{u_k \in \mathbb{B}} \{ \sum_{V_k \in \mathbb{B}^{p-1}} e^{-\frac{1}{2}\|Z_k - D\left[\begin{smallmatrix} u_k \\ V_k \end{smallmatrix}\right]\|^2_{\Sigma_p^{-1}}} \}. \tag{3.5.14}$$

This formulation makes the implicit assumptions

$$Pr(Z_k \mid U_k, E_k = 0) = p_N(Z_k - DU_k)$$

$$Pr(V_k = [u_{k+1}, \ldots, u_{k+p-1}]' \mid E_k = 0) = Pr(V_k = [u_{k+1}, \ldots, u_{k+p-1}]'),$$

for which we do not seek theoretical justification, but point out that, in the white noise case, such assumptions lead to practically realisable detection strategies that perform well. We add that in the high SNR coloured noise case, if we extract the dominant term from the maximisation in (3.5.14), the decision boundaries are still hyperplanes in Z_k-space. To see this, note that for a high SNR $(p,1)$-DFE, a point Z lies on the decision boundary if and only if the weighted norm $\|Z - DU\|^2_{\Sigma_p^{-1}}$ is minimised by $U_1, U_2 \in \mathbb{B}^p$ where U_1 and U_2 have distinct first components. In other words,

$$(Z - DU_1)'\Sigma_p^{-1}(Z - DU_1) = (Z - DU_2)'\Sigma_p^{-1}(Z - DU_2), \tag{3.5.15}$$

which implies by symmetry of Σ_p^{-1} that

$$U_1'D'\Sigma_p^{-1}DU_1 - U_2'D'\Sigma_p^{-1}DU_2 + 2Z'\Sigma_p^{-1}D(U_2 - U_1) = 0. \tag{3.5.16}$$

But (3.5.16) is the equation of a hyperplane in Z_k-space. Thus, the effect of coloured noise is to tilt and displace the decision hyperplanes with respect to the white noise case.

3.5.4 Adaptation

In the present formulation, we require explicit *a priori* knowledge of the channel model in order to implement the block decision feedback equalizer. This assumption is generally unrealistic—physical channels tend to be unknown in advance and also time-varying. As mentioned in the section (1.1.3), the remedy is to make the equalizer adapt its internal channel model automatically—first to identify and

then to track the actual channel parameters. An adaptive equalizer [6] often uses an initial (known) training data sequence and then, once the channel is identified, switches to a decision-directed mode, replacing the training sequence with its own decisions.

We treat the subject of adaptation of the block DFE by analogy with the conventional adaptive decision feedback equalizer [36, 20]. For simplicity, we assume a finite impulse response channel with real coefficients and ignore the adaptation of the would-be feedforward filter which cancels the precursor intersymbol interference. We are assuming, then, that the true input sequence (and hence channel state) is known, and derive *minimum mean-square error* (MMSE) settings for the decision device and the causal multivariable feedback filter of the block DFE. The analysis applies equally to the (p,p)-DFE and the $(p,1)$-DFE, with their respective optimal decision devices. Thus we define (in a notation consistent with section 3.2.2)

$$\begin{aligned} \mathcal{V}_k &\triangleq Z_k - \hat{D}U_k \\ &= Y_k - \hat{H}x_k - \hat{D}U_k \\ &= Y_k - \hat{Y}_k, \end{aligned} \tag{3.5.17}$$

where we have used (3.2.5), and defined $\hat{Y}_k \triangleq \hat{H}x_k + \hat{D}U_k$, the hats "$\hat{}$" denoting estimated variables.

The equalizer adapts its channel model $\hat{\mathbf{h}} = [\hat{h}_0, \ldots, \hat{h}_L]'$ to minimise the "noise variance" at the quantizer input

$$\min_{\hat{H},\hat{D}} \mathcal{E}\left\{\|\mathcal{V}_k\|^2\right\} = \mathcal{E}\left\{\|Y_k - \hat{Y}_k\|^2\right\}. \tag{3.5.18}$$

Let us rewrite $\hat{Y}_k$ as

$$\hat{Y}_k = \left[\hat{H} \,\middle|\, \hat{D}\right] \begin{bmatrix} x_k \\ U_k \end{bmatrix} \triangleq \hat{\mathcal{H}}\mathcal{U}_k, \tag{3.5.19}$$

where

$$\widehat{\mathcal{H}} = \begin{bmatrix} \widehat{h}_L & \widehat{h}_{L-1} & \cdots & \cdots & \widehat{h}_0 & 0 & \cdots & 0 \\ 0 & \widehat{h}_L & \cdots & \cdots & \widehat{h}_1 & \widehat{h}_0 & \cdots & 0 \\ \vdots & & \ddots & & & & \ddots & \vdots \\ 0 & \cdots & \cdots & \widehat{h}_L & \cdots & \cdots & \cdots & \widehat{h}_0 \end{bmatrix} \in \mathbb{R}^{p\times(L+p)} \tag{3.5.20}$$

and

$$\mathcal{U}_k = [u_{k-L}, \ldots, u_{k-1}, u_k, \ldots, u_{k+p-1}]' \in \mathbb{B}^{L+p}. \tag{3.5.21}$$

We now need the following facts presented in the Lemma below (appendix C.3 contains a proof).

Lemma 3.5.1 *Let the input sequence $\{u_k\}$ to the channel be a stationary white random process with zero mean and unit variance. There holds*[10]

$$\mathcal{E}\left\{Y_k'\widehat{\mathcal{H}}\mathcal{U}_k\right\} = \sum_{j=0}^{L} \widehat{h}_{L-j} \sum_{i=1}^{p} \phi_{i,i+j}, \tag{3.5.22}$$

$$\begin{aligned} \mathcal{E}\left\{\mathcal{U}_k'\widehat{\mathcal{H}}'\widehat{\mathcal{H}}\mathcal{U}_k\right\} &= \text{tr } \widehat{\mathcal{H}}'\widehat{\mathcal{H}} \\ &= p\sum_{i=0}^{L} \widehat{h}_i^2, \end{aligned} \tag{3.5.23}$$

where $\phi_{ij} \triangleq \mathcal{E}\{y_{k+i-1}u_{k-L+j-1}\}$ for $i = 1, \ldots, p$ and $j = 1, \ldots, L+p$, with $\widehat{\mathcal{H}}$, $\mathcal{U}_k$ as in (3.5.20), (3.5.21) and y_k, Y_k as in (3.2.9) and (3.2.4) respectively.

Denoting the channel output autocorrelation matrix by

$$C = \mathcal{E}\{Y_kY_k'\} \in \mathbb{R}^{p\times p}$$

where C has components

$$c_{ij} = \mathcal{E}\{y_{k+i-1}y_{k+j-1}\}, \ 1 \le i, j \le p,$$

we have $\mathcal{E}\{Y_k'Y_k\} = \text{tr } C$. Making use of the Lemma, we can express the MMSE

[10] tr A is the trace of the square matrix A.

criterion as

$$\begin{aligned}\mathcal{E}\left\{\|\mathcal{V}_k\|^2\right\} &= \mathcal{E}\left\{Y_k'Y_k - 2Y_k'\widehat{\mathcal{H}}\mathcal{U}_k + \mathcal{U}_k'\widehat{\mathcal{H}}'\widehat{\mathcal{H}}\mathcal{U}_k\right\} \\ &= \text{tr } C - 2\sum_{j=0}^{L}\widehat{h}_{L-j}\sum_{i=1}^{p}\phi_{i,i+j} + p\sum_{i=0}^{L}\widehat{h}_i^2. \end{aligned} \tag{3.5.24}$$

The minimum mean-square error tap coefficients are the values of $\widehat{h}_j$ for which

$$\frac{\partial}{\partial\widehat{h}_j}\mathcal{E}\left\{\|\mathcal{V}_k\|^2\right\} = 0, \quad j = 0,\ldots,L, \tag{3.5.25}$$

which yields the simple (batch) solution

$$\widehat{h}_j = \frac{1}{p}\sum_{i=0}^{p-1}\mathcal{E}\left\{y_{k+i}u_{k+i-j}\right\} = 0, \quad j = 0,\ldots,L. \tag{3.5.26}$$

The block DFE under MMSE adaptation would thus use the tap settings (3.5.26) to form the $\{F,G,H,D\}$ (3.2.2) matrices which determine the decision device and feedback filter. Unfortunately, in practice, we could not expect to realise this MMSE solution for lack of knowledge of the channel-input cross correlation. Instead, we try for an approximate solution. This is furnished by a *stochastic gradient descent* strategy [20], which essentially ignores the expectation in (3.5.18), and uses the (real time) iterative adaptation rule

$$\widehat{h}_j(k+1) = \widehat{h}_j(k) - \frac{1}{2}\eta\frac{\partial}{\partial\widehat{h}_j(k)}\|\mathcal{V}_k\|^2, \; j = 0,\ldots,L, \tag{3.5.27}$$

in which $\widehat{h}_j(k)$ is the current (time k) estimate of the jth channel tap h_j, and η is a small step size. We will now state the form this algorithm takes for the block DFE in the following theorem whose proof may be found in appendix C.4.

Theorem 3.5.1 (Adaptive Block DFE) *Let Y_k be as defined in (3.2.4), and $\widehat{\mathcal{H}}$, $\mathcal{U}_k$ as defined in (3.5.19). The stochastic gradient descent training algorithm for the adaptation of the parameters $\widehat{h}_j(k)$, $j = 0,\ldots,L$ of the block DFE matrices $\widehat{D}$ and $\widehat{H}$ forming $\widehat{\mathcal{H}}$ (3.5.20) is given, for $k = 0,1,2,\ldots$, by*

$$\widehat{h}_j(k+1) = \widehat{h}_j(k) + \eta\sum_{i=0}^{p-1}u_{k+i-j}\epsilon_{k+i}, \tag{3.5.28}$$

where $\epsilon_k = y_k - \widehat{y}_k$ *is the channel output noise estimate at time* k *and* $\widehat{y}_k$ *defined as*

$$\widehat{y}_k = \sum_{i=0}^{L} \widehat{h}_i(k) u_{k-i}. \tag{3.5.29}$$

The block DFE, then, replaces the unknown h_i, which parametrise its decision device and feedback filter, with their estimates $\widehat{h}_i(k)$ and updates these according to (3.5.28) during the training phase. After training, the block DFE can use its own decisions $\widehat{u}_k$, in lieu of the u_k in (3.5.28) automatically to track variations in the physical channel. For QAM signalling, the derivation of the adaptation algorithm proceeds analogously.

3.6 Conclusions

In this chapter, we have presented a natural generalisation of conventional decision feedback equalization—the block DFE—based on a block processing channel model. For a fixed block size, we can distinguish two principal types—the full-blocking block DFE ((p,p)-DFE) and the sliding-window block DFE ($(p,1)$-DFE). The $(p,1)$-DFE has superior performance to the (p,p)-DFE and requires more computation, although for reasonably high signal-to-noise ratios its variant, the high SNR $(p,1)$-DFE, offers comparable performance at reduced complexity.

We investigated the extremes in performance of the block DFE and showed how to recover the conventional DFE, the maximum likelihood sequence estimator and the symbol-by-symbol maximum *a posteriori* detector as special limiting cases. We demonstrated, using trellis-based ideas, the equivalence of earlier approaches in [13, 14, 43] with the high SNR block DFE. We discussed briefly issues relating to computational complexity of block DFE decision device realisations, which is exponential in the block size. This indicates a need for finding simpler vector quantizers if the channel response is long compared to the sampling interval, even though substantial BER improvements over the DFE are obtainable for modest block sizes.

Concerning possible extensions, we maintain that the block DFE is a very flexible structure. This flexibility is a by-product of the separate design of the feedback filter

and the decision device. The block processing model can be applied to linear and non-linear ARMA channels. The decision device can easily be modified to handle M-ary signalling or quadrature amplitude modulation and even coloured noise. We showed that it is straightforward to incorporate adaptation into the block DFE structure, making it a viable practical scheme. We did not discuss the important aspect of coding, but suggest that the block DFE is naturally amenable to the equalization of block codes.

We have reserved any discussion of performance analysis for the next chapter. As in the analysis of the conventional decision feedback equalizer [23, 16, 17], it is a wise move to study first the performance of the non-adaptive (tuned) structure. This involves calculating or bounding the primary (noise-induced) error probability, the analysis of error propagation, error recovery times, and error probability enhancement due to error propagation. The feedback mechanism of the block DFE, while assuring its generally good performance, combines with the non-linearity of the decision device to make most analyses very difficult. For this reason, in the sequel we will only be attempting a performance analysis of the two-input ($p = 2$) block DFE on second order channels.

Chapter 4

Two-Input Block DFE - Detailed Performance Analysis

4.1 Introduction

There are many aspects that should be considered in evaluating a communication system. Some examples are the system's sensitivity to various timing and phase errors and mistuning of parameters, its noise immunity, and the effects of finite precision arithmetic. All of these factors influence the eventual system's performance and are reflected in the symbol error probability $Pr(\hat{u}_k \neq u_k)$. Since we are treating the transmission of uncoded binary *iid* data, the symbol error probability is the same as the bit error rate and we will refer to these simply as the error probability.

As we mentioned at the end of chapter 3, the analysis of error probability in a non-linear feedback equalizer is not only a study of the *primary* (noise-induced) error probability, but also of the effects of feedback of decision errors (causing error propagation) which enhances the primary error probability. In our study of error propagation, we only treat the noiseless case. This has the virtue of simplifying the finite-state Markov process description—since the probability of a particular state transition is then just the *a priori* probability of observing a particular input symbol $Pr(u_k)$.

When we come to discuss the noiseless propagation of errors in the block DFE, we use a definition of stability that has appeared in the literature on decision feed-

back equalizers [16]. Loosely speaking, we say that an equalizer (incorporating decision feedback) is *stable* on a given channel, or alternatively, that a channel is in the stability class for that equalizer, if there is no input sequence that can cause an indefinitely long stream of decision errors. On such channels, we can expect the equalizer to recover from an initial error condition in a "reasonable time" (in the absence of noise). We make these notions more precise in section 4.5.

In chapter 3, we introduced the block decision feedback equalizer by extending the concept of decision feedback equalization to many dimensions. There, we dealt mainly with aspects of design, various realisations and their simulated performance, and extensions of the technique. In this chapter, we will concern ourselves with the analysis of performance and stability of the two-input block DFE, which we review in section 4.2. The reason for treating only the two-input case is that, in the general p-input case for $p > 2$, no explicit expressions for the solution of the integer programming problem for the decision rule (3.2.20) have been found. We reiterate here that an exact decision rule can be written down for the (2,2)-DFE, whereas this rule is only an, albeit good, approximation for the (2,1)-DFE (3.3.3), valid for moderate signal-to-noise ratios. We also restrict the stability discussions to first and second order channels. Firstly, this constrains the complexity of a finite-state Markov process based analysis (relying on the independence of the input sequence to the channel). Secondly, the results are easily interpreted and visualised graphically.

Our analysis of the two-input block DFE splits up into three broad categories:

1. Primary error probability analysis.

2. Sufficient conditions for error recovery.

3. Necessary conditions for error recovery.

A primary error is a decision error caused by channel noise. The analysis of such errors is commonplace in the equalization literature and assumes that there have been no past decision errors, ignoring the possibility of error propagation which may occur regardless of the signal-to-noise ratio. We will give an example of this kind of analysis applied to a high SNR (2,1)-DFE, on a first order channel (section 4.3), yielding the error probability as a function of the signal-to-noise ratio. We take a direct approach, using knowledge of the block DFE's decision boundary to

express the primary bit error rate as a definite integral. This is unconventional—many authors prefer to use the union bound [3], which is essentially a trellis-based approach (see [13], for example), to bound the primary error probability.

Error recovery analyses concentrate on the existence of certain undesirable or *pathological* input sequences which may generate bursts of decision errors. We seek conditions on the parameters of the finite impulse response channel that preclude this behaviour in the absence of noise. We look firstly at sufficient conditions derived from considerations of the noise immunity of the decision device. This leads to *eye conditions* which we present in section 4.4. We establish the stability class of second order channels for the (2,2)-DFE and (high SNR) (2,1)-DFE in section 4.5. These conditions are necessary and sufficient for the two-input block DFE to have a bounded noiseless error recovery time (or a short expected error recovery time). The complexity of this analysis, however, would seem to prohibit generalisation to longer channels or higher dimensional block DFEs, although it is applicable in principle.

4.2 The Two-Input Block DFE

Before proceeding with the error probability analyses, we rederive the two-input block DFE model, mimicking the treatment in chapter 3. We assume a finite impulse response channel with unit cursor[1] $h_0 = 1$ and coefficients h_i $(i = 1, \ldots, L)$. The input to the channel is a sequence of *independent* random variables $\{u_k\}_{k=0}^{\infty}$, where u_k takes values in $\mathbb{B} \triangleq \{-1, +1\}$ with equal probability[2]. The channel output at sampling instant k, corrupted by zero-mean white Gaussian noise n_k with variance σ^2, gives the received signal

$$y_k = u_k + \sum_{j=1}^{L} h_j u_{k-j} + n_k. \tag{4.2.1}$$

Defining the channel state x_k as the vector of the last L channel inputs,

$$x_k \triangleq [u_{k-L}, u_{k-L+1}, \ldots, u_{k-1}]' \in \mathbb{B}^L$$

[1] Again, as in section 3.2.3, we are assuming that the precursor ISI has been removed by linear equalization.

[2] Equiprobability is not an essential assumption.

where x' denotes the transpose of x, we can express the system in state-space form

$$\begin{aligned} x_{k+1} &= Ax_k + bu_k, \ k = 0, 1, \ldots \\ y_k &= cx_k + u_k + n_k \end{aligned} \tag{4.2.2}$$

where

$$A = \begin{bmatrix} 0 & I_{L-1} \\ 0 & 0 \end{bmatrix}; \ b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \ c = [h_L, \ \ldots \ , h_2, \ h_1] \tag{4.2.3}$$

and where I_n is the identity matrix of order n and 0 is a matrix of zeros of the appropriate size.

We now form a block processing realisation of block length $p = 2$ for (4.2.2) by defining

$$U_k \triangleq [u_k, u_{k+1}]', \ Y_k \triangleq [y_k, y_{k+1}]', \ N_k \triangleq [n_k, n_{k+1}]', \tag{4.2.4}$$

then

$$\begin{aligned} x_{k+2} &= Fx_k + GU_k, \ \ k = 0, 2, 4, \ldots \\ Y_k &= Hx_k + DU_k + N_k, \end{aligned} \tag{4.2.5}$$

where, like (3.2.2) with $p = 2$,

$$F = \begin{bmatrix} 0 & I_{L-2} \\ 0 & 0 \end{bmatrix}; \ G = \begin{bmatrix} 0 \\ I_2 \end{bmatrix} \quad \text{if } L > 2$$

or

$$F = 0 \ ; \ G = \begin{bmatrix} 0 & I_L \end{bmatrix} \quad \text{if } L < 2$$

and

$$H = \begin{bmatrix} h_2 & h_1 \\ h_3 & h_2 \end{bmatrix} \ ; \ D = \begin{bmatrix} 1 & 0 \\ h_1 & 1 \end{bmatrix}, \tag{4.2.6}$$

and we define $h_k = 0$ if $k > L$.

In analogy to the DFE, DU_k is the direct term of current channel inputs for decoding (acting as the block cursor), Hx_k contains past input terms and acts as

the tail of the intersymbol interference.

The block DFE (Fig.3-3) assembles its past decisions in a state vector estimate $\widehat{x}_k = [\widehat{u}_{k-L}, \ldots, \widehat{u}_{k-1}]'$ with which it attempts to cancel the Hx_k term at the decision device input

$$Z_k \triangleq [z_{k,1}, z_{k,2}]' = Y_k - H\widehat{x}_k = DU_k + HE_k + N_k \tag{4.2.7}$$

where

$$E_k \triangleq x_k - \widehat{x}_k \tag{4.2.8}$$

is the state estimation error.

On the assumption that past decisions are correct ($E_k = 0$), the input to the memoryless decision device is ISI-free and in a from suitable for vector quantization. Since the block length $p = 2$, there are two possible maximum *a posteriori* decision device strategies: the full-blocking (2,2)-DFE, producing two input estimates, and the sliding-window (2,1)-DFE, producing one estimate per iteration. The (2,2)-DFE computes its decisions via

$$\widehat{U}_k = [\widehat{u}_k, \widehat{u}_{k+1}]' = \underset{U_k \in \mathbb{B}^2}{\arg\min} \|Z_k - DU_k\|^2, \tag{4.2.9}$$

using (4.2.5) to generate successive state estimates. The optimal (2,1)-DFE uses a sum of exponentials criterion like (3.2.20) which, in the moderate to high SNR case, reduces to the high SNR (2,1)-DFE criterion

$$\widehat{u}_k = [1,\ 0] \times \underset{U_k \in \mathbb{B}^2}{\arg\min} \|Z_k - DU_k\|^2, \tag{4.2.10}$$

with D as in (4.2.6), obtaining $\widehat{x}_{k+1}$ using (3.2.21). In this chapter, we will assume the block DFE is operating under a minimum distance criterion. This is optimal for the (2,2)-DFE but a high signal-to-noise ratio approximation for the (2,1)-DFE. We are therefore assuming that the decision boundary is piecewise linear for the purposes of the analysis.

Geometrical arguments yield the explicit solutions to the minimisation in (4.2.9)

$$\widehat{u}_k = \mathrm{sgn}\{\mathrm{sgn}(l_1) + \mathrm{sgn}(l_2) + \mathrm{sgn}(l_3)\} \tag{4.2.11}$$

$$\widehat{u}_{k+1} = \mathrm{sgn}\{\mathrm{sgn}(l_4) + \mathrm{sgn}(l_5) - \mathrm{sgn}(h_1)\,\mathrm{sgn}(l_3)\} \tag{4.2.12}$$

in which

$$\begin{aligned} l_1 &= h_1 z_{k,2} + z_{k,1} - h_1 \\ l_2 &= h_1 z_{k,2} + z_{k,1} + h_1 \\ l_3 &= (h_1 - \mathrm{sgn}(h_1)) z_{k,2} + z_{k,1} \\ l_4 &= z_{k,2} + h_1 \;;\; l_5 = z_{k,2} - h_1, \end{aligned} \tag{4.2.13}$$

for the (2,2)-DFE ($k = 0, 2, 4, \ldots$) and by (4.2.11) for the high SNR (2,1)-DFE ($k = 0, 1, 2, \ldots$) (see Lemma 3.3.1). The decision boundaries for $h_1 = \frac{2}{3}$ are shown in chapter 3 Fig.3-5.

We can equally well write the decision rules (4.2.11)-(4.2.12) in a piecewise form

$$\widehat{u}_k = \begin{cases} \mathrm{sgn}(h_1 z_{k,2} + z_{k,1}), & |h_1 z_{k,2} + z_{k,1}| > |h_1| \\ \mathrm{sgn}(l_3), & |h_1 z_{k,2} + z_{k,1}| < |h_1| \end{cases} \tag{4.2.14}$$

$$\widehat{u}_{k+1} = \begin{cases} \mathrm{sgn}(z_{k,2}), & |z_{k,2}| > |h_1| \\ -\mathrm{sgn}(h_1)\,\mathrm{sgn}(l_3), & |z_{k,2}| < |h_1| \end{cases} . \tag{4.2.15}$$

This results from the observation that the decision regions depend on the strips formed by the pairs of parallel lines $l_1 = 0$, $l_2 = 0$ and $l_4 = 0$, $l_5 = 0$ in (4.2.13). This alternate representation will be useful when we determine the finite-state Markov process corresponding to a particular channel class in section 4.5.2.

4.3 Primary Error Probability Example

Determining the error probability of a non-linear equalizer is a hard problem in the sense that exact solutions often cannot be found. The conventional method for trellis-based detection algorithms, such as the Viterbi algorithm and its variants [11, 13, 14, 19], is to make the simplifying assumption that there have been no past decision errors, then to find a bound on this primary error probability. These analyses use the idea of an *error event*, which is a sequence of incorrect state transitions in the trellis. The probability of error is the probability of the union of the various

error events, which is upper bounded by the sum of their individual probabilities [19]. This union bound is dominated at high signal-to-noise ratios by the minimum (Hamming) distance error event. Although this method works well for the Viterbi decoder, the effect of past decision errors on the error probability is more pronounced when decision feedback is incorporated. This is the case in most of the simplified Viterbi schemes. As we will see later, error propagation in a block DFE can easily double the primary error probability.

The style of analysis presented in this section differs from the trellis-based calculation in that we compute the primary error probability directly, using knowledge of the decision boundary's geometry. We assume piecewise linear decision boundaries, but otherwise the calculations are exact. As we saw in section3.5, the optimal decision device typically performs only slightly better than its high SNR approximation.

The primary error probability of the $(p,1)$-DFE on a particular channel (and for a given SNR) is defined as

$$\mathcal{P}_0 \triangleq Pr(\hat{u} \neq u_k \mid \mathcal{A}), \tag{4.3.1}$$

where the conditioning $\mathcal{A}$ is indicative of the assumption of no past errors, *i.e.*, $\hat{u}_{k-i} = u_{k-i}\ \forall i > 0$. Applying Bayes' rule, we have

$$\begin{aligned} \mathcal{P}_0 &= \sum_{\alpha \in I\!B} Pr(\hat{u} \neq u_k \mid \mathcal{A}, u_k = \alpha) Pr(u_k = \alpha) \\ &= \frac{1}{2} Pr(\hat{u}_k = -1 \mid \mathcal{A}, u_k = 1) + \frac{1}{2} Pr(\hat{u}_k = 1 \mid \mathcal{A}, u_k = -1) \end{aligned} \tag{4.3.2}$$

by equiprobability of the input sequence $\{u_k\}$ (the conditioning on the symbol probabilities is redundant). As a preliminary example, let us compute this quantity for a conventional decision feedback equalizer, whose decisions are given by (1.2.1). In the absence of past decision errors, the DFE's output is just (assuming $h_0 = 1$)

$$\hat{u}_k = \mathrm{sgn}(u_k + n_k), \tag{4.3.3}$$

so that

$$\begin{aligned}\mathcal{P}_0(\text{DFE}) &= \frac{1}{2}Pr(\text{sgn}(u_k + n_k) = 1 \mid u_k = -1) \\ &\quad + \frac{1}{2}Pr(\text{sgn}(u_k + n_k) = -1 \mid u_k = 1) \\ &= \frac{1}{2}Pr(n_k > 1) + \frac{1}{2}Pr(n_k < -1). \end{aligned} \tag{4.3.4}$$

Now, assuming that $n_k \sim N(0, \sigma^2)$, we have

$$Pr(n_k > 1) = Pr(n_k < -1) = \int_1^\infty \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{-t^2}{2\sigma^2}} dt = Q(\frac{1}{\sigma}), \tag{4.3.5}$$

where $Q(\cdot)$ is the complement of the cumulative normal distribution function

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt. \tag{4.3.6}$$

Hence, from (4.3.4), the DFE's primary error probability is just

$$\mathcal{P}_0(\text{DFE}) = \frac{1}{2}Q(\frac{1}{\sigma}) + \frac{1}{2}Q(\frac{1}{\sigma}) = Q(\frac{1}{\sigma}). \tag{4.3.7}$$

Turning again to the two-input block DFE, the computation is complicated by the non-linearity of the decision device and its dependence on the channel (we are assuming the high SNR case, so there is no dependence on σ^2). The decision device inputs are (with $h_0 = 1$)

$$\begin{aligned} z_1 &\triangleq z_{k,1} = u_k + n_k \\ z_2 &\triangleq z_{k,2} = u_{k+1} + h_1 u_k + n_{k+1}, \end{aligned} \tag{4.3.8}$$

and the joint probability density of the vector N_k (4.2.5) of *iid* noise samples is decomposable as

$$p_N(N_k) = p_n(n_k)p_n(n_{k+1}) \triangleq p_N(n_1, n_2), \tag{4.3.9}$$

where

$$p_n(m) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-m^2}{2\sigma^2}}, \ \sigma^2 = \text{Var}(n_k). \tag{4.3.10}$$

Applying Bayes' rule, we condition on the four possible values of U_k, obtaining

$$Pr(\widehat{u}_k \neq u_k \mid \mathcal{A}) = \sum_{\alpha \in I\!B} \sum_{\beta \in I\!B} Pr(\widehat{u}_k = \alpha \mid \mathcal{A}, u_k = -\alpha, u_{k+1} = \beta) \times Pr(u_k = \alpha) Pr(u_{k+1} = \beta), \quad (4.3.11)$$

where we have used the independence of $\{u_k\}$ and causality to remove the conditioning on the *a priori* symbol probabilities (which are assumed equiprobable). We first investigate a symmetry property, pertaining to the high SNR $(p,1)$-DFE decision rule, that will simplify the calculations.

Property 4.3.1 (Symmetry of Decision Rule) *Suppose that the decision device input $Z_k = Z_1 \in I\!R^p$ to the high SNR $(p,1)$-DFE results in the decision $\widehat{u}_k = u_1 \in I\!B$. Then when $Z_k = -Z_1$, the corresponding decision is $\widehat{u}_k = -u_1$, where u_1 is the first component of U_1.*

Proof: Let $U = U_1$ minimise $\|Z_1 - DU\|^2$. Then $\|-Z_1 - DU\|^2 = \|Z_1 - D(-U)\|^2$ is minimised by $-U = U_1$ or $U = -U_1$.

□

Remarks

1. If a point Z lies on the decision boundary, then so does its reflection in the origin $-Z$.

2. The same symmetry property applies to the optimal (p,q)-DFE with decisions given by (3.2.20).

Now define the following regions in connection with the (2,1)-DFE with decision rule $\mathcal{D}_{2,1}(\cdot) : I\!R^2 \rightarrow I\!B$ given by (4.2.10)

$$\begin{aligned} \mathcal{H}_+ &= \{Z \in I\!R^2 \mid \mathcal{D}_{2,1}(Z) = +1\} \\ \mathcal{H}_- &= \{Z \in I\!R^2 \mid \mathcal{D}_{2,1}(Z) = -1\}. \end{aligned} \quad (4.3.12)$$

Property 4.3.1 implies that $Z_1 \in \mathcal{H}_+ \iff -Z_1 \in \mathcal{H}_-$. We have not specified the decision device's output for a point Z lying on the decision boundary, but this is

unimportant since the latter is a set of Lebesgue measure zero in Z_k-space and will have no contribution to the probability of error due to noise.

Consider one of the conditional probabilities in equation (4.3.11). Using (4.3.8) we have (using the subscripts 1 and 2 in place of k and $k+1$ for the components of both Z_k and N_k)

$$\begin{aligned}
& Pr(\widehat{u}_k = -1 \mid \mathcal{A}, u_k = 1, u_{k+1} = 1) \\
= \; & Pr(Z = [z_1, z_2]' \in \mathcal{H}_- \mid z_1 = n_1 + 1, z_2 = n_2 + 1 + h_1) \\
= \; & Pr(-Z \in \mathcal{H}_+ \mid z_1 = n_1 + 1, z_2 = n_2 + 1 + h_1), \text{ by Property 4.3.1} \\
= \; & Pr(Z \in \mathcal{H}_+ \mid z_1 = -n_1 - 1, z_2 = -n_2 - 1 - h_1) \\
= \; & Pr(Z \in \mathcal{H}_+ \mid z_1 = n_1 - 1, z_2 = n_2 - 1 - h_1) \\
= \; & Pr(\widehat{u}_k = 1 \mid \mathcal{A}, u_k = -1, u_{k+1} = -1), \qquad (4.3.13)
\end{aligned}$$

having used the symmetry of the noise density. The upshot of this is that only two of the four terms in (4.3.11) are distinct. From here, it is a straightforward but messy step to write down the expression for the primary error probability. This may be found in appendix D.1. As an illustration, we go through the computation for the special case of $h_1 = 1$. This case is a representative "bad" channel in respect of giving a relatively high simulated error probability over the class of first order channels (normalised to have the same energy).

When $h_0 = 1$, $h_1 = 1$, the decision region for $\widehat{u}_k = -1$ can be represented as the union of the three disjoint sets below (see (D.1.1) in appendix D.1),

$$\begin{aligned}
\mathcal{H}_1 &= \{(z_1, z_2) \in I\!R^2 \mid 1 \leq z_2 < \infty, -\infty < z_1 < 1 - z_2\} \\
\mathcal{H}_2 &= \{(z_1, z_2) \in I\!R^2 \mid -\infty < z_1 < 0, -\infty < z_2 < 1\} \\
\mathcal{H}_3 &= \{(z_1, z_2) \in I\!R^2 \mid 0 < z_1 < \infty, -\infty < z_2 \leq -1 - z_1\}.
\end{aligned}$$

Consequently, we find from (4.3.11) that the primary error probability is

$$\begin{aligned}
2\mathcal{P}_0 = \; & \{\int_1^{\infty}\int_{-\infty}^{-n_2} + \int_{-\infty}^{-1}\int_{-\infty}^{-n_2-2} + \int_{-1}^{1}\int_{-\infty}^{-1} + \int_{-1}^{\infty}\int_{-\infty}^{-n_2-2} + \\
& \int_{-\infty}^{-3}\int_{-\infty}^{-n_2-4} + \int_{-3}^{-1}\int_{-\infty}^{-1}\} p_N(n_1, n_2) dn_1 dn_2, \qquad (4.3.14)
\end{aligned}$$

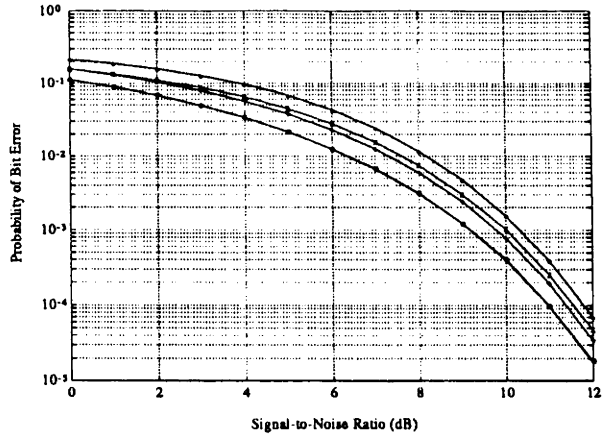

Figure 4-1: Bit error rates with and without error propagation.
The curves are (from top to bottom): the simulated error probability for the DFE; the simulated error probability for the (2,1)-DFE; the theoretical bounds and simulated primary error probability for the (2,1)-DFE.

where $p_N(\cdot,\cdot)$ is given by (4.3.9). This is expressible more compactly as

$$\begin{aligned} \mathcal{P}_0 &= \frac{1}{2}Q(\frac{1}{\sigma})\left\{1-Q(\frac{1}{\sigma})-Q(\frac{3}{\sigma})\right\}+\frac{1}{2}\int_1^\infty Q(\frac{t}{\sigma})p_n(t)dt+ \\ &\quad \frac{1}{2}\int_{-\infty}^\infty Q(\frac{t+2}{\sigma})p_n(t)dt+\frac{1}{2}\underbrace{\int_3^\infty Q(\frac{4-t}{\sigma})p_n(t)dt}_{\mathcal{I}_3}. \end{aligned} \tag{4.3.15}$$

In addition, the first two of the three integrals above reduce to

$$\int_1^\infty Q(\frac{t}{\sigma})p_n(t)dt = \frac{1}{2}Q^2(\frac{1}{\sigma}) \tag{4.3.16}$$

$$\int_{-\infty}^\infty Q(\frac{t+2}{\sigma})p_n(t)dt = \frac{1}{2}Q(\frac{\sqrt{2}}{\sigma}), \tag{4.3.17}$$

whereas we can obtain simple bounds on the third as

$$Q(\frac{3}{\sigma})Q(\frac{1}{\sigma}) < \mathcal{I}_3 < Q(\frac{3}{\sigma}). \tag{4.3.18}$$

These results are derived in appendix D.2. We combine (4.3.15-4.3.18) to obtain upper and lower bounds on the primary error probability. These have been plotted in Fig.4-1 together with the corresponding curve (4.3.4) for the DFE. Fig.4-1 also shows the performance curves for the high SNR (2,1)-DFE with and without error propagation (simulated over a minimum of 10^6 points). The BER curve for the

DFE (simulated over 10^6 points) has been included as a reference. The bounds are indistinguishable for this range of signal-to-noise ratios, and coincide with the simulation. Clearly, the enhancement of the primary bit error rate through error propagation is a significant factor in both the (2,1)-DFE and the DFE. We will return to this point in section 4.5.

4.4 Sufficient Conditions for Noiseless Error Recovery

4.4.1 Eye Conditions

Recall that a correctly tuned decision feedback equalizer, in the non-adaptive mode, produces its decisions according to

$$\widehat{u}_k = \text{sgn}(u_k + r_k + n_k),$$

where

$$r_k = \sum_{i=1}^{L} h_i(u_{k-i} - \widehat{u}_{k-i})$$

represents the residual intersymbol interference after decision feedback. Clearly if the channel parameters are such that $|r_k| < 1$ for all k, then a decision error can only be caused by noise (although the ISI can still detract from the noise immunity–making the occurrence of an error more likely). Channels satisfying a condition of this type are known as *open eye* channels for the DFE. We say alternatively that the channel parameters satisfy an *eye condition* [20]. On such channels, all errors are noise-induced and error propagation cannot occur. In the absence of noise, the DFE is guaranteed to recover from an initial error condition in a finite number of steps (of the order of the channel length). Eye conditions for M-ary decision-directed equalizers may be found in [21]. We consider the problem of deriving eye conditions for a high SNR (2,1)-DFE operating on a second order channel $[h_0 = 1, h_1, h_2]$ in this section.

In the absence of noise, the decision device input Z_k is given by

$$\begin{bmatrix} z_{k,1} \\ z_{k,2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ h_1 & 1 \end{bmatrix} \begin{bmatrix} u_k \\ u_{k+1} \end{bmatrix} + \begin{bmatrix} h_2 & h_1 \\ 0 & h_2 \end{bmatrix} \begin{bmatrix} e_{k-2} \\ e_{k-1} \end{bmatrix}.$$

An eye condition will be satisfied in this case if the magnitude of the residual block intersymbol interference is always less than the minimum distance to the decision boundary, $r_{\min}$, or

$$\left\| \begin{bmatrix} h_2 & h_1 \\ 0 & h_2 \end{bmatrix} \begin{bmatrix} e_{k-2} \\ e_{k-1} \end{bmatrix} \right\|^2 < r^2_{\min}(h_1), \ \forall e_{k-1}, e_{k-2} \in I\!E \tag{4.4.1}$$

where the notation indicates the dependence of the minimum distance on the channel parameter h_1.

The decision boundary in the high SNR case is composed of three straight lines in Z_k-space

$$l_1 \ : \quad h_1(z_2 - 1) + z_1 = 0 \tag{4.4.2}$$

$$l_2 \ : \quad (h_1 - \operatorname{sgn}(h_1))z_2 + z_1 = 0 \tag{4.4.3}$$

$$l_3 \ : \quad h_1(z_2 + 1) + z_1 = 0, \tag{4.4.4}$$

where $Z_k = [z_1, z_2]'$ is the decision device input. Recalling the reasoning of section 3.2, the intersymbol interference (and noise) has the effect of displacing the processed received signals Z_k from their natural positions given in (3.3.2). Equation (4.4.1), then, simply expresses the condition that these displacements can never alter the decision, assuming that the block DFE is tuned in the usual sense.

We now turn to computing $r_{\min}(h_1)$. By symmetry, we need only consider the distances from Z_{++} and Z_{+-} (3.3.2) to the decision boundary. The minimum of these distances will determine $r_{\min}(h_1)$. Let us label the point of intersection of lines l_1 and l_2 by P_1 and of l_2 and l_3 by P_2. These have respective co-ordinates

$$[h_1(1 - h_1 \operatorname{sgn}(h_1)), h_1 \operatorname{sgn}(h_1)]' \text{ and } [h_1(h_1 \operatorname{sgn}(h_1) - 1), -h_1 \operatorname{sgn}(h_1)]'.$$

We also denote by $d(\cdot, \cdot)$ the Euclidean distance between two points or the perpendicular distance between a point and a line (depending on the arguments which appear). Defining $r_+(h_1)$ and $r_-(h_1)$ as the minimum distances from Z_{++} and Z_{+-}

to the decision boundary respectively, we have by definition

$$r_{\min}(h_1) = \min\{r_+(h_1), r_-(h_1)\}. \tag{4.4.5}$$

It is clear from the geometry (see Fig.3-5, for example) that Z_{++} is always closer to l_1 and l_2 than to l_3 for any h_1, as Z_{+-} is to l_2 and l_3. This says that

$$r_+(h_1) = \min\{d(Z_{++}, l_1), d(Z_{++}, \mathrm{P}_1), d(Z_{++}, l_2)\}$$

$$r_-(h_1) = \min\{d(Z_{+-}, l_2), d(Z_{+-}, \mathrm{P}_2), d(Z_{+-}, l_3)\}.$$

But these distances are easily computed as

$$\begin{aligned}
d_1(h_1) &\triangleq d(Z_{++}, l_1) = \sqrt{1 + h_1^2} \\
d_2(h_1) &\triangleq d(Z_{++}, \mathrm{P}_1) = \sqrt{(1 - h_1 + sh_1^2)^2 + (h_1(s-1) - 1)^2} \\
d_3(h_1) &\triangleq d(Z_{++}, l_2) = (1 + (h_1 - s)(h_1 + 1) \;/\; \sqrt{1 + (h_1 - s)^2} \\
d_4(h_1) &\triangleq d(Z_{+-}, l_2) = (1 + (h_1 - s)(h_1 - 1) \;/\; \sqrt{1 + (h_1 - s)^2} \\
d_5(h_1) &\triangleq d(Z_{+-}, \mathrm{P}_2) = \sqrt{(-1 - h_1 + sh_1^2)^2 + (h_1(s+1) - 1)^2} \\
d_6(h_1) &\triangleq d(Z_{+-}, l_3) = \sqrt{1 + h_1^2},
\end{aligned}$$

where $s \triangleq \mathrm{sgn}(h_1)$. Furthermore, direct comparison of these distances, as a function of h_1, allows us to write

$$r_+(h_1) = \begin{cases} d_3(h_1), & h_1 \geq h_1^* \\ d_2(h_1), & 1 \leq h_1 \leq h_1^* \\ d_1(h_1), & -0.5 \leq h_1 \leq 1 \\ d_3(h_1), & h_1 \leq -0.5 \end{cases},$$

and

$$r_-(h_1) = \begin{cases} d_4(h_1), & h_1 \geq 0.5 \\ d_6(h_1), & -1 \leq h_1 \leq 0.5 \\ d_5(h_1), & -h_1^* \leq h_1 \leq -1 \\ d_4(h_1), & h_1 \leq -h_1^* \end{cases},$$

where $h_1^* \simeq 1.5437$ is the positive real root of the equation $h_1^3 - 2h_1^2 + 2h_1 - 2 = 0$.

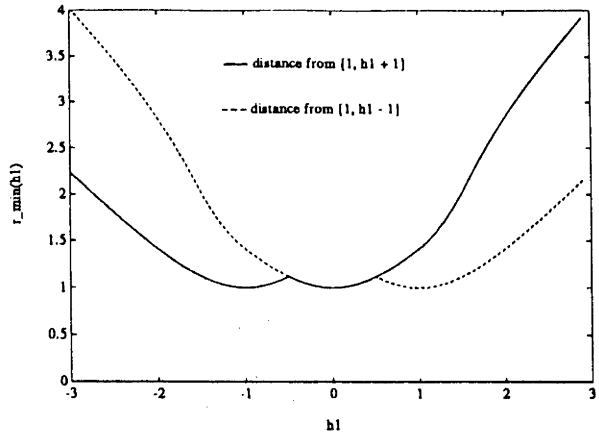

Figure 4-2: Minimum distance to the decision boundary.

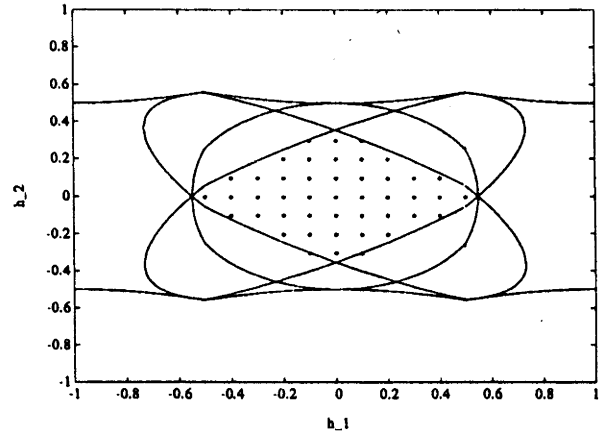

Figure 4-3: Open eye region (starred) for high SNR (2,1)-DFE.

The above functions are plotted in Fig.4-2. Taking the minimum, as suggested by (4.4.5), gives

$$r_{\min}(h_1) = \begin{cases} \sqrt{h_1^2 + 2h_1 + 2}, & h_1 \leq -0.5 \\ \sqrt{h_1^2 + 1}, & |h_1| \leq 0.5 \\ \sqrt{h_1^2 - 2h_1 + 2}, & h_1 \geq 0.5. \end{cases}$$

Notice that $r_{\min}(h_1) \geq 1$, whereas $r_{\min} \equiv 1$ for the DFE. This indicates that the block DFE has better noise immunity than the DFE. We now apply the eye condition (4.4.1) for the eight non-zero choices of $[e_{k-2}, e_{k-1}]' \in I\!E^2$, each of which defines a region in (h_1, h_2)-channel space.(For $E_k = [0,0]'$, (4.4.1) is trivially satisfied.) The intersection of these regions defines the class of channels for which the eye condition for the high SNR (2,1)-DFE is satisfied. This is the lozenge-shaped region surrounding the origin in Fig.4-3. We remark that this is only a sufficient condition

for error propagation to be absent in such a block DFE, and is liable to be fairly conservative. As a comparison, the corresponding region for the DFE is shown dashed in the figure. (This is the intersection of the regions $\{|h_1 e_{k-1} + h_2 e_{k-2}| < 1; e_{k-1}, e_{k-2} \in I\!E\}$, or the l_1-ball $|h_1| + |h_2| < \frac{1}{2}$.) We will be considering in the next section *necessary and sufficient* conditions that guarantee noiseless error recovery of the block DFE in a finite time in the next section.

4.5 Necessary Conditions for Noiseless Error Recovery

A full error probability analysis of the block DFE must include the important effect of error propagation. The approach taken here, based on finite state Markov processes, follows similar lines to the error analysis of the tuned DFE in [23]. We consider the recovery of the block DFE from an initial error condition *in the absence of noise.* This is complementary to the calculations of section 4.3 in that a complete performance analysis should account for both noise-induced errors and their propagation. The key assumptions are the *statistical independence* of the binary input sequence to the channel and the fact that the decision device is *memoryless.* Under these assumptions, an analysis of the dynamics of error propagation in a block DFE in terms of finite-state Markov processs is possible. The noiseless assumption is by way of simplification and is inessential. The inclusion of noise into the analysis is possible once the noiseless case has been solved, and we sketch the required steps in the appendix, using a DFE on a first order channel as an example.

4.5.1 Finite State Markov Process Description

In modelling error propagation in a block DFE the quantity of interest is E_k (4.2.8). A non-zero entry in E_k indicates that a decision error has occurred in the last L time instants (due to a noise spike, for instance). We wish to track the progress of this initial error as it is propagated around the feedback loop of the block DFE.

For a two-input block DFE with decision function $\mathcal{D}_{pq}(\cdot)$ (3.2.20), operating on a finite impulse response channel, the decision error vector (4.2.8) evolves by shift

register action (*c.f.* (2.3.3)) via

$$E_{k+1} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} E_k + \begin{bmatrix} 0 \\ u_k - \mathcal{D}_{2,1}(DU_k + HE_k + N_k) \end{bmatrix}, \; k = 0, 1, 2, \ldots, \quad (4.5.1)$$

in the sliding-window case, and via

$$E_{k+2} = U_k - \mathcal{D}_{2,2}(DU_k + HE_k + N_k), \; k = 0, 2, 4, \ldots, \quad (4.5.2)$$

in the full-blocking $q = 2$ case. Some initial error condition, arising at time $k = 0$, is responsible for the initial error state E_0 of the block DFE. We assume that subsequent errors are due to error propagation alone and hence set the noise N_k to zero in (4.5.1) and (4.5.2)—which is basically a high signal-to-noise ratio approximation. If we further assume that $\mathcal{D}(\cdot)$ is a *memoryless* decision function and that $\{u_k\}$ is a sequence of *independent* binary random variables, then we can label E_k as the state in a finite state Markov process model of the block DFE. The FSMP for a given channel is determined by its initial state distribution and the set of *transition probabilities* between its states (which we will define shortly). These transition probabilities are easily obtained if we assume equiprobable input symbols.

The zero-error state $E_k = [0, \ldots, 0]' = \underline{0}$, corresponding to a succession of L correct decisions, has special status in the FSMP. We see from (4.5.1) and (4.5.2) that when $E_k = \underline{0}$, then $E_{k+q} = \underline{0}$, since we must have $\mathcal{D}_{pq}(DU_k) = (I_q \mid 0)U_k$, in the absence of ISI and noise, for any sensible decision device. This means that $E_k = \underline{0}$ is an *absorbing state* of the FSMP for any channel. Thus, in the absence of noise, once the system reaches the zero-error state it stays there.

Definition 4.5.1 (Error Recovery) *We say the block DFE has* recovered from error *once it reaches the zero-error state, and define the* error recovery time R *as the first time* $k = R$ *such that* $E_k = \underline{0}$ *(assuming that* $E_0 \neq \underline{0}$*).*

We would like to determine the statistics of the error recovery time for the block DFE on a given channel and whether there exist any input sequences that would prevent its recovery from an initial error condition. This last question will lead us in section 4.5.2 to necessary (and sufficient) conditions on the parameters of a second order FIR channel that guarantee error recovery. That the concept of error

recovery is well-defined is a consequence of the *reachability* of the zero-error state. This means that we can always find an input sequence $\{u_k\}_{k=0}^{R-1}$ which drives the system to the state $E_R = \underline{0}$ in a finite time R, regardless of its initial state. This property is demonstrated for the high SNR (2,1)-DFE in appendix D.3.

Since u_k is binary, each component e_{k-i} of E_k has only three possible values: $e_{k-i} \in \{-2, 0, 2\} \stackrel{\Delta}{=} I\!E$, so there are 3^L E_k-states in the FSMP. We can roughly halve the number of state transitions to consider, and therefore facilitate the error recovery analysis, by observing the following simple property of the FSMP.

Property 4.5.1 (Symmetry of Finite State Markov Process) *On a given FIR channel* $[h_0, \ldots, h_L]$, *the finite-state Markov process governing the* E_k*-state transitions in the* (p, q)*-DFE has the symmetry property*

$$\begin{aligned} & E_k \;\rightarrow\; E_{k+q} \;\; \textit{under} \;\; U_k \\ \Longleftrightarrow\; & -E_k \;\rightarrow\; -E_{k+q} \;\; \textit{under} \;\; -U_k, \end{aligned} \tag{4.5.3}$$

with respect to inputs U_k.

Proof: This follows directly from the negatives of (4.5.1) and (4.5.2), observing the odd symmetry of the block DFE's decision function (Property 4.3.1).

□

Remarks

1. Here,"$\rightarrow$" should be read "transits to".

2. The same symmetry property with respect to the probability of a transition also holds in the noisy case when the noise is white.

If we define an *aggregated state* as a complementary pair of E_k states, $A(k) = \{E_k, -E_k\}$, then these new aggregated states $A(k)$, of which there are only $M = \frac{1}{2}(3^L + 1)$, form a finite-state Markov process. This follows from Property 4.5.1 and the equiprobability of the input sequence. The same approach to simplifying the error recovery analysis was used in [23] in connection with the DFE. We now make some general definitions concerning the aggregated finite-state Markov process.

We first choose an indexing scheme for the E_k-states that assigns an integer $\langle E_k \rangle \in \{1, \ldots, 3^L\}$ to error state E_k. This we do by transforming $E_k \mapsto \widetilde{E}_k$ according to

$$\widetilde{E}_k = \frac{1}{2} E_k + [1, \ldots, 1]',$$

so that the elements of $\widetilde{E}_k$ belong to $\{0, 1, 2\}$, then treating $\widetilde{E}_k$ as a ternary number with its L^{th} component as the most significant bit. This reads

$$\langle E_k \rangle = (1 + \frac{1}{2} e_{k-1}) 3^{L-1} + (1 + \frac{1}{2} e_{k-2}) 3^{L-2} + \cdots + (1 + \frac{1}{2} e_{k-L}) 3^0, \qquad (4.5.4)$$

so that if $\langle E_k \rangle = i$, then $\langle -E_k \rangle = 3^L - i + 1$ for $1 \leq i \leq M = \frac{1}{2}(3^L + 1)$. In particular, the absorbing zero-error state has index $\langle E_k = \underline{0} \rangle = M$. Denoting the M possible values of the aggregated state $A(k)$ by A_i, $1 \leq i \leq M$, we map these to the indexes of the E_k-states via

$$A_1 = \{1, 3^L\},\ A_2 = \{2, 3^L - 1\}, \ldots,\ A_{M-1} = \{M-1, M+1\},\ A_M = \{M\}. \quad (4.5.5)$$

We can now define the state distribution vector $\pi_k = [\pi_{k,1}, \ldots, \pi_{k,M}]'$ and transition probability matrix $P = (p_{ij})$ of the finite-state Markov process by

$$\begin{aligned} \pi_{k,i} &= Pr(A(k) = A_i) \qquad (4.5.6) \\ p_{ij} &= Pr(A(k+1) = A_i \mid A(k) = A_j),\ 1 \leq i, j \leq M, \end{aligned}$$

which satisfy

$$\pi_{k+1} = P \pi_k.$$

These definitions will be helpful in section 4.5.3.

4.5.2 Channel Space Partition

For the remainder of this chapter, we confine the discussion to an arbitrary second order channel $[h_0 = 1, h_1, h_2]$. This will allow graphical interpretation of the results. With this restriction, the finite-state Markov process for the two-input block DFE

has 9 E_k-states or 5 aggregated states (4.5.5)

$$A_1 = \{[-2,-2]', [2,2]'\},\ A_2 = \{[-2,0]', [2,0]'\},\ A_3 = \{[-2,2]', [2,-2]'\},$$

$$A_4 = \{[0,-2]', [0,2]'\},\ A_5 = \{[0,0]'\}$$

in response to the 4 possible inputs

$$U_k \in I\!B^2 \triangleq \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \end{bmatrix} \right\}.$$

Our aim is to determine the partitioning of (h_1, h_2)-channel space into its respective channel classes. We treat the more involved (2,2)-DFE case first and then specialise to the high SNR (2,1)-DFE case.

As explained at the start of section 4.5, we ignore the noise terms in (4.2.7). The decision device input is therefore

$$Z_k = \begin{bmatrix} 1 & 0 \\ h_1 & 1 \end{bmatrix} U_k + \begin{bmatrix} h_2 & h_1 \\ 0 & h_2 \end{bmatrix} E_k, \tag{4.5.7}$$

with $E_k = [e_{k-2}, e_{k-1}]'$ defined by (4.2.8) and $U_k = [u_k, u_{k+1}]'$.

In order to classify channel space according to the finite-state Markov process arising from a choice of h_1 and h_2, we consider the values of these parameters which result in a zero argument of one (or more) of the signum functions in (4.2.11)-(4.2.12). This leads us to define a *switching boundary* below.

Definition 4.5.2 (Switching Boundary) *For given values of E_k and U_k and a given condition holding in (4.2.14) or (4.2.15), the* switching boundary *is the set of values $\{h_1, h_2\}$ that makes the argument of the corresponding sign function zero.*

It is intuitively clear that the set of switching boundaries divides the channel space into regions, or classes of channels, in each of which a *single* FSMP applies. To see this, note that a change in a decision is due to a change in sign of one or more sign functions in (4.2.11)-(4.2.12), which, by definition, only occurs if a switching boundary is crossed.

We need to consider the switching boundaries that arise from the various choices

of E_k and U_k (or equivalently for the purposes of error recovery, $A(k)$ and U_k). We look first at the zero-error state $E_k = [0,0]'$. Using the "strip" representation (4.2.14)-(4.2.15) of the decision function, it is easily demonstrated that under all 4 inputs $U_k \in I\!B^2$, transitions can only be made to $E_{k+2} = [0,0]'$. For instance, let $U_k = [1,1]'$, (4.5.7) gives $Z_k = [1, 1+h_1]'$ which is independent of h_2. The '>' part of (4.2.14) holds for all h_1 and implies $\widehat{u}_k = \text{sgn}(1 + h_1 + h_1^2) = 1 = u_k$. Similarly $\widehat{u}_{k+1} = u_{k+1}$ for all h_1. Thus, as expected, the zero-error state is an *absorbing* state of the FSMP for any h_1 (and in fact for any FIR channel) so that once this state is reached, no further decision errors can be made due to error propagation. Similar statements also apply to the (2,1)-DFE.

Next we consider E_k states that have a non-zero entry. The 4 curves defined when an argument of a sign function in (4.2.14)-(4.2.15) is zero may be expressed in the form $c(h_1, h_2) = 0$, where

$$c(h_1, h_2) = a_1 h_1^2 + a_2 h_2^2 + a_3 h_1 h_2 + a_4 h_1 + a_5 h_2 + a_6$$

for real constants a_i, and are thus conic sections in the (h_1, h_2)-plane. As an example, to determine which part of (4.2.14) applies in a particular region of channel space, it is necessary to consider the values of h_1 and h_2 for which $|h_1 z_{k,2} + z_{k,1}| = |h_1|$. Applying the following identity,

$$|x| = |y| \Longleftrightarrow (x+y)(x-y) = 0$$

valid for real x and y, defines two curves in the (h_1, h_2)-plane

$$[1 + h_1^2, h_1]U_k + [h_2, h_1 + h_2 h_1]E_k \pm h_1 = 0,$$

after substituting for Z_k in (4.5.7) and simplifying. The same reasoning is applicable to (4.2.15). Thus for each of the 32 combinations (16 if we use property 4.5.1) of the 8 non-zero error states ($E_k \neq \underline{0}$) and the 4 inputs (U_k), we must plot a total of 8 curves (4 from the conditioning and 4 from the arguments of the signum functions) in order to determine the set of switching boundaries and hence what transitions can occur in each partition of channel space.

We choose the case $E_k = [0,2]'$, $U_k = [1,-1]'$ as a representative example. The curves in question are given by

$$c_i(h_1,h_2) \triangleq c_i = 0,\ i = 1,\ldots,8,$$

where

$$\begin{aligned} c_1 &= -1 + 2h_2\ ;\ c_2 = c_1 + h_1\ ;\ c_3 = c_1 + 2h_1 \\ c_4 &= 1 + h_1^2 + 2h_1h_2\ ;\ c_5 = c_4 + h_1\ ;\ c_6 = c_4 + 2h_1 \\ c_7 &= c_4 - c_1\ ;\ c_8 = c_4 + c_3, \end{aligned}$$

and the decisions (4.2.14)-(4.2.15) may be re-expressed as

$$\widehat{u}_k = \begin{cases} \operatorname{sgn}(c_5) & \text{if } c_4c_6 > 0 \\ \operatorname{sgn}(c_7) & \text{if } c_4c_6 < 0 \text{ and } h_1 > 0 \\ \operatorname{sgn}(c_8) & \text{if } c_4c_6 < 0 \text{ and } h_1 < 0 \end{cases} \tag{4.5.8}$$

$$\widehat{u}_{k+1} = \begin{cases} \operatorname{sgn}(c_2) & \text{if } c_1c_3 > 0 \\ -\operatorname{sgn}(c_7) & \text{if } c_1c_3 < 0 \text{ and } h_1 > 0 \\ \operatorname{sgn}(c_8) & \text{if } c_1c_3 < 0 \text{ and } h_1 < 0 \end{cases} \tag{4.5.9}$$

Plotting these curves and using (4.5.8)-(4.5.9) establishes the switching boundaries depicted in Fig.4-4 for this particular state/input pair. To illustrate the decision procedure, we take the point $h_1 = 0.6$, $h_2 = 0.8$ (the cross in Fig.4-4). We compute $c_4c_6 \simeq 8.17 > 0$ so the first line of (4.5.8) gives $c_5 = 2.92$ and $\widehat{u}_k = \operatorname{sgn}(c_5) = 1$. Similarly $c_1c_3 = 0.28 > 0$ and (4.5.9) gives $c_2 = 0.8$ and $\widehat{u}_{k+1} = \operatorname{sgn}(c_2) = 1$, so that $E_{k+2} = [1,-1]' - [1,1]' = [0,-2]'$. Note that all $[h_1,h_2]$ points in any one region of Fig.4-4 will result in the same E_{k+2}. These boundaries are the same for the state $E_k = [0,-2]$ with input $U_k = [-1,1]'$. Continuing this way, yields the full set of switching boundaries (for all possible error states and input combinations) which, when overlaid, allows the complete determination of the FSMP for any second order channel with $h_0 = 1$. This is shown in Fig.4-5. For instance, in the region of Fig.4-5 containing the point $[h_1 = 0.6,\ h_2 = 0.8]$ the particular FSMP has the state transition diagram shown in Fig.4-6 (the transition probabilities are marked on the

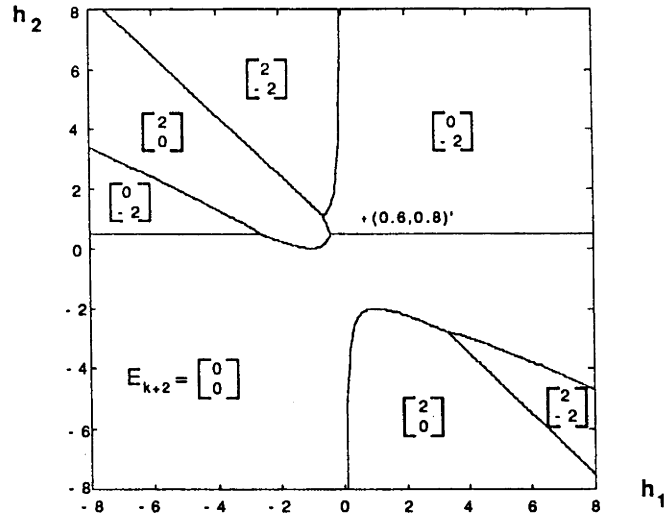

Figure 4-4: Possible transitions from error state $[0, 2]$.

The $(2, 2)$-DFE is in state $E_k = [0, 2]'$ at time k with input $U_k = [1, -1]'$.

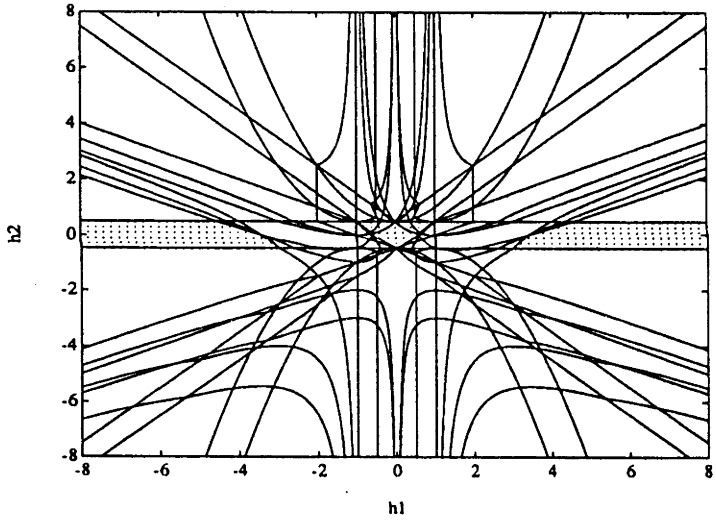

Figure 4-5: FSMP regions for (2,2)-DFE.

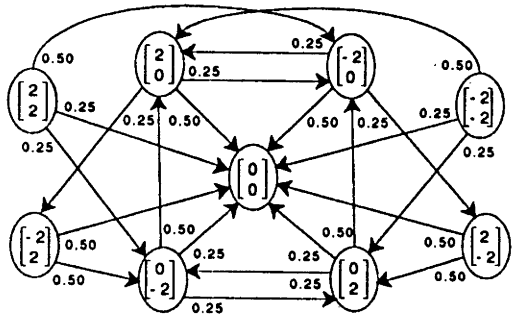

Figure 4-6: FSMP diagram for $h_0 = 1$, $h_1 = 0.6$, $h_2 = 0.8$.

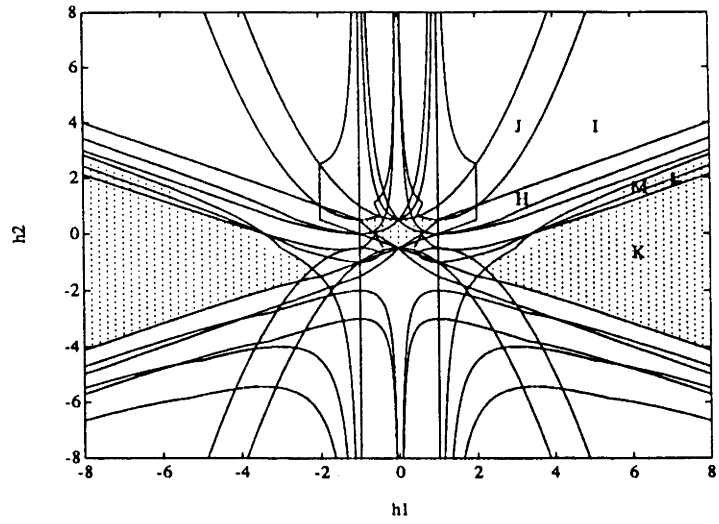

Figure 4-7: FSMP regions for the (2,1)-DFE.

branches).

As the same FSMP applies to each point inside any one region, it is possible to classify classes of channels with "desirable" error recovery properties, as will be explained in the section 4.5.4. By ignoring those boundaries relevant to the $\widehat{u}_{k+1}$ decision in (4.2.15), the channel space partition for the high SNR (2,1)-DFE is easily deduced from the switching boundaries of Fig.4-5 and is displayed in Fig.4-7. The inner region has been blown up in Fig.4-8.

4.5.3 Noiseless Error Recovery Statistics

The usefulness of determining the channel space partition for a block DFE is that it enables us to arrange the channels in classes. In each class, just *one* finite-state Markov process governs the (noiseless) error recovery of the device. In particular, this allows us to determine the statistics of the noiseless error recovery time R (definition 4.5.1) on *any* channel by considering just a finite (although large) number of finite-state Markov processs. By way of example, there are a maximum of 149 possible noiseless FSMPs for the high SNR (2,1)-DFE on a second order channel (this is the number of separate regions in the channel partition of Fig.4-7). In this section, we choose a sample of thirteen[3] of these regions and compute the mean and variance of the error recovery time. We will use the aggregated state FSMP description of

[3]My lucky number.

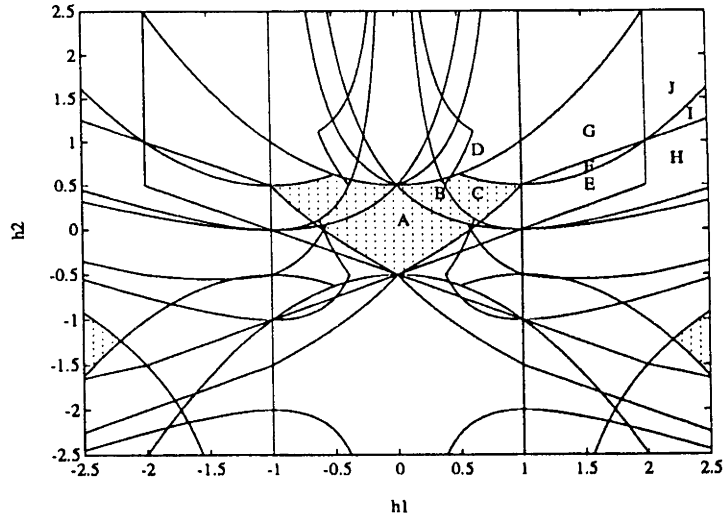

Figure 4-8: Detail of the inner FSMP regions.

section 4.5.1 to simplify the calculations.

The channel has order 2, so that there are $M = 5$ aggregated states (4.5.5) in the FSMP. The absorbing zero-error state is A_5. We partition the 5×5 transition probability matrix P defined in (4.5.6) as follows:

$$P = \left[\begin{array}{c|c} Q & 0 \\ \hline r' & 1 \end{array}\right], \tag{4.5.10}$$

where Q is a 4×4 matrix. Now we make the following claim [23].

Claim 4.5.1 *The eigenvalues of Q in (4.5.10) have modulus less than unity.*

Proof: (Sketch only). Since P is a stochastic matrix, its maximum eigenvalue is unity. The absorbing zero-error aggregated state $A_5 = \underline{0}$ is reachable from any other non-zero aggregated state (see appendix D.3). Hence A_5 is the only irreducible closed subset of the FSMP (all other closed subsets of the FSMP must contain A_5) [60]. It follows that the multiplicity of the unity eigenvalue is one, and this corresponds to the absorbing state A_5.

□

Next we suppose that the initial error state E_0 induces a distribution (4.5.6) π_0

across the aggregated states. We partition π_0 conformably with P as

$$\pi_0 = [\overline{\pi}_0', \rho]'.$$

Standard procedures [8] from the theory of finite-state Markov processs give the mean error recovery time (expected time to reach the absorbing state) as

$$\mu_R = \mathcal{E}\{R\} = \mathbf{1}'T\overline{\pi}_0, \tag{4.5.11}$$

and the variance of the error recovery time as

$$\sigma_R^2 = \mathcal{E}\{R^2\} - \mu_R^2 = \mathbf{1}'(2T-1)T\overline{\pi}_0 - \mu_R^2, \tag{4.5.12}$$

where $T \triangleq (I-Q)^{-1}$ is well defined by Claim 4.5.1, and $\mathbf{1} = [1,1,1,1]'$.

We consider the recovery time from a noise-induced error at time -2 such that $E_0 = [0,2]'$. The corresponding aggregated state is $A(0) = A_4$, so that $\overline{\pi}_0 = [0,0,0,1]'$. Table 4.1 lists the 13 different channels (all have $h_0 = 1$) with their respective error recovery time statistics. (Stability of a particular channel class is defined in section 4.5.4.) Each channel is representative of the channels in its region of (h_1, h_2)-channel space, and is indicated by a capital letter in Figs.4-7 and 4-8. We remark that regions A, B, C, K, L and M have the most desirable error recovery properties (these are all finite error recovery channel classes—see next section). Regions E and I in fact have identical FSMPs—regions that are adjacent along a line always have distinct FSMPs, whereas the same is not always the case for regions adjacent at a point. There is also equivalence of FSMPs in regions that are symmetric with respect to the $h_1 = 0$ axis (although in higher dimensions this need not hold [23]). The aggregated state FSMPs for regions A, B, C and D are given in Fig.4-9. We have picked the clustering of regions E–J around the point $[h_1 = 2,\ h_2 = 1]$ (see simulation example in chapter 3) to show that closeness of channel parameters does not necessarily imply closeness of error recovery time statistics.

class	h_1	h_2	stable	μ_R	σ_R^2
A	0	0	yes	2	4
B	0.3	0.3	yes	2	4
C	0.6	0.3	yes	2.5	6.5
D	0.6	0.8	no	5	38
E	1.5	0.5	no	3	11
F	1.5	0.65	no	4	22.67
G	1.5	1.5	no	3.556	16.69
H	2.2	0.5	no	2.667	7.778
I	2.2	1.15	no	3	11
J	2.2	1.5	no	6	58
K	6	0.5	yes	2	4
L	6	1.2	yes	2	4
M	6	1.5	yes	2.25	5.25

Table 4.1: Mean and variance of noiseless error recovery time.

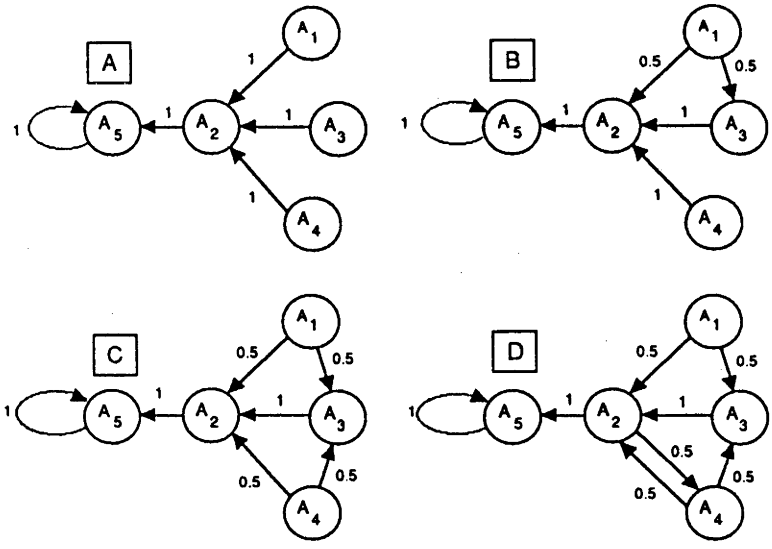

Figure 4-9: Aggregated FSMPs for channel classes A, B, C and D.

4.5.4 Pathology of Error Propagation

The FSMP for certain values of h_1 and h_2 (*e.g.*, Fig.4-6) occasionally contains subsets of non-zero-error states among which indefinite error propagation could occur (subject to the realisation of a particular, undesirable input sequence). Thus on certain channels the block DFE may suffer from long bursts of errors triggered by a single (primary) incorrect decision. In view of this, it is important to identify channels where the phenomenon of prolonged error propagation cannot occur. To do so, we need only examine the *possible* state transitions—the actual transition probabilities of the FSMP are not required. This motivates the definition of stability below (introduced in [17] in connection with the DFE).

Definition 4.5.3 (Stability) *For a given noiseless channel and initial error state* $E_0 \neq \underline{0}$, *we say the input sequence* $\{u_k\}$ *to the block DFE is* pathological *if it causes an indefinite sequence of transitions between non-zero error states* $E_k \neq \underline{0}$, $k > 0$. *We define the block DFE to be* stable *on a given channel if no pathological sequences exist for any initial error state.*

The next example shows what we mean by pathological behaviour of the block DFE.

Example 4.5.1 (Pathological Sequence) *For the channel* $[h_0 = 1, h_1 = 0.6, h_2 = 0.8]$, *if the (2,2)-DFE is initially in* E_k*-state* $[0, 2]'$ *then the (period 4) input sequence*

$$\{u_k\}_{k=0}^{\infty} = \{1, -1, -1, 1, 1, -1, -1, 1, \cdots\}$$

results in the error-state sequence

$$\{E_k\}_{k=1}^{\infty} = \left\{ \begin{bmatrix} 0 \\ -2 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ -2 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \cdots \right\},$$

so that the (2,2)-DFE never recovers from the initial error condition.

The simplest form of pathological behaviour (as illustrated in the example) is manifested in an FSMP that has a transition from a non-zero aggregated state to itself. In theory, then, there exist channels on which the block DFE may have an *indefinite* recovery time due to certain input sequences. Of course the probability of such an input condition persisting tends to zero exponentially with time. Nonetheless, the

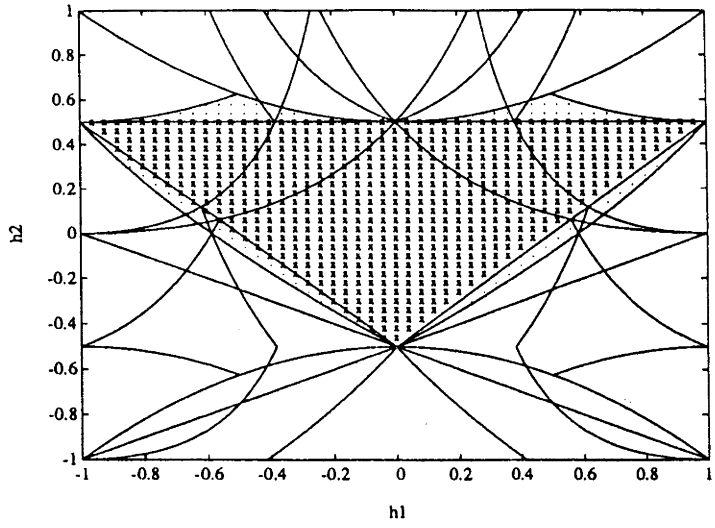

Figure 4-10: Comparison of stability regions.
Inner stability region of the high SNR (2,1)-DFE shown dotted, stability triangle of the decision feedback equalizer shaded with "x"s. The inner stability region of the (2,2)-DFE is obtained from the dotted region by disregarding the two cusp-shaped regions above the line $h_2 = 0.5$.

existence of pathological sequences generally increases the block DFE's expected error recovery time on a given channel.

Finite Recovery Time Channels

The class of second order channels on which the block DFE is stable is readily determined by examining the FSMPs for each channel class. On stable channels, the block DFE has a bounded error recovery time equal to the maximum number of steps required to reach the zero-error state from a non-zero initial error state. The stability regions (equivalence classes of stable channels) for the (2,2)-DFE have been shaded in Fig.4-5 and for the high SNR (2,1)-DFE in 4-7 (if $h_0 \neq 1$, the axes must be scaled appropriately). On their respective stable channels, the (2,1)-DFE will recover in at most 3 steps, whereas the (2,2)-DFE has a noiseless error recovery time of at most 2 (although the latter has inferior BER performance).

We pause to make a comparison with the conventional DFE. The latter has a triangular stability region [4] with vertices $[0, -0.5]'$, $[-0.5, 0.5]'$ and $[0.5, 0.5]'$, shown as the region shaded with "x"s in Fig.4-10. This is a subset of the stability region for the (2,2)-DFE which is in turn contained in that of the (2,1)-DFE. We remark that for points in the outer semi-infinite regions, the block DFE is effectively treating h_1

as the cursor—the DFE is unable to do this.

It is curious to note that the stability region of the (2,2)-DFE lies in the strip $|h_1| < \frac{1}{2}$ of Fig.4-5. The reason for this may be gleaned from the switching boundary diagrams for error states $[0,2]'$ and $[0,-2]'$ (not shown). These states have transitions between each other when $h_1 > \frac{1}{2}$ and transitions to themselves when $h_1 < -\frac{1}{2}$. Notice that when $|h_1| \gg |h_2| > \sqrt{2}$ for the (2,1)-DFE, the dominant part of (4.5.7) is $Z_k \approx [h_1 e_{k-1}, h_1 u_k]'$, so that (4.2.14) implies

$$\widehat{u}_k = \mathrm{sgn}(h_1 e_{k-1} + h_1^2 u_k) = u_k.$$

This explains the roughly triangular shape of the outer semi-infinite regions in Fig.4-7. The above arguments run in favour of the (2,1)-DFE in terms of stability over a wider class of channels than either the (2,2)-DFE or the DFE.

4.6 Conclusions

We review here the findings of this chapter and make some comments as to their utility.

We presented an analysis of the primary error probability of the two-input block DFE operating on a second order finite impulse response channel. We used a direct approach, expressing the error probability as an integral over the decision region. This calculation was complicated by the non-linearity of the block DFE's decision boundary. This technique is extendible to longer channels, although the complexity, as measured by the number of terms involved, grows exponentially with the channel length. Bounds could be obtained easily by considering the dominant terms—corresponding to points near the decision boundary. Unfortunately, this kind of analysis is not easy to apply to higher dimensional block DFEs (with $p > 2$), and this suggests the need for considering more conventional error probability bounds, as in [13], based on minimum distance error events.

We derived eye conditions for the high SNR (2,1)-DFE on an arbitrary second order channel. These were necessarily conservative and show the limitations of ignoring the structure of the block DFE's error dynamics. We corrected the shortcomings of this naïve method by providing a detailed and exhaustive finite-state Markov

process analysis, again assuming a second order noiseless channel. This allowed us to associate all second order channels having the same error recovery properties with a finite number of channel classes. We gave examples of the mean and variance of noiseless error recovery times for various channel classes. We saw that the (2,1)-DFE is more robust than either the (2,2)-DFE or the conventional DFE, since it is stable on a wider class of channels. This conclusion is also supported by the statistics of the noiseless error recovery times (see [23] for DFE error recovery statistics). It is not clear how the inclusion of noise affects the error recovery properties of block DFEs—although we know how to handle the analysis in principle (the calculations of the error probability for the DFE in appendix D.4 serve as an illustration).

Most importantly, we saw how complicated the analysis of the error probability of a block DFE can become. The step involved in passing from the conventional DFE to the block DFE is substantial—from an analysis point of view. The added dimension, encountered with the two-input block DFE, results in the curvature of the FSMP boundaries (which are straight lines for the DFE), and complicates the topology of the regions associated with the various channel classes. We stress that this analysis assumed a piecewise linear decision boundary, whereas the true decision boundary is curved for lower signal-to-noise ratios. It is hard to see how the three-input case could be handled—not to mention higher order block DFEs on arbitrary channels, even if explicit expressions can be found for the decision functions.

A final remark seems in order. Although it may be possible to obtain simple bounds on the primary error probability of a non-adaptive block DFE, the observed error probability, when the effects of error propagation are included, is liable to be considerably higher. Hence, in any comprehensive performance analysis of the block DFE, we should not simply assume that past decisions are correct, but should try to allow for the important phenomenon of error propagation.

Chapter 5

Maximum A Posteriori Decision Feedback Detection

5.1 Introduction

In this chapter, we present a new symbol-by-symbol detection scheme for the equalization of uncoded binary data transmitted on noisy, dispersive communications channels. The new detector—the *MAP decision feedback detector* [54]—derives from a generalisation of block decision feedback equalization (chapter 3) in which the assumption of *correct* decision feedback is dropped. We conserve the basic feedback structure of the block DFE, but incorporate a new decision device, designed around a fixed-delay maximum *a posteriori* criterion which exploits knowledge of certain operational error probabilities to give improved performance over the block DFE. The formulation uses decision feedback in a structured way—effectively reducing the dimension of the space on which decisions are formed. This permits substantial simplification of previous fixed-delay optimal detectors that minimise maximum *a posteriori* sequence estimation criteria. The idea of applying decision feedback to simplify detector design was mentioned in [26].

The new detector depends on a parameter p, the block size, related to the delay in obtaining decisions and representative of the number of significant taps in the sampled channel impulse response. Choice of this parameter fixes the input dimension of the detector's decision device. The decision device depends on the channel

impulse response coefficients and the signal-to-noise ratio (which together determine the probabilities of certain error events needed in the decision criterion). The detector's criterion is a natural extension of the block DFE's decision function [15], and reduces to the latter under the assumption of correct past decisions (ignoring the extra information provided by decision error probabilities).

The main drawback of the technique is the complexity of the decision function which, in the present formulation, is exponential in the channel length. In order to cope with long channels, the feedback filter can be chosen to cancel the tail (as in conventional decision feedback equalization), thus presenting an effectively shorter channel around which the decision device can be designed. Furthermore, the recent application of feedforward neural networks to the equalization problem (see [56], for instance) may provide a means of implementing these complicated decision functions. In [56] non-linear feedforward processing was applied to the problem of generating data estimates with a fixed delay from a window of received signals. Although in [56] no use was made of decision feedback in cancelling the intersymbol interference.

In the next section, we detail the system model and give a review of the relevant classical non-linear detection methods that will help put the subsequent development of the new detector into perspective. We show the steps leading to the derivation of the MAP decision feedback detector's decision function in section 5.3.1. There is some overlap with section 3.4.3 of the material on non-linear detectors. although a more thorough treatment is given in this chapter. In section 5.3.2, we describe an iterative method that can be used to acquire the error probabilities needed to realise the detector's decision device. Section 5.4.1 gives a first order example of the detector's operation for block size $p = 2$. In section 5.4.2 we give a more substantial test of the performance of the new detector, comparing it with the (2,1)-DFE (section 3.3.3), the decision feedback equalizer and the Viterbi algorithm on a second order channel. We take what could be referred to as a "worst case channel" (see section 2.3.2) for the comparisons—this allows the improvement in bit error rate performance over the block DFE to be seen at the relatively low SNRs (or equivalently, high BERs) attainable in the simulations. We also include examples comparing of the decision regions formed by the new detector and the block DFE in the two-input case on first and second order channels.

5.2 Overview of Classical Non-Linear Detection

In this section we give a brief overview of the classical non-linear detection systems alluded to earlier, leading to the design of the new detector which we will examine in detail in section 5.3. We assume a standard baseband pulse-amplitude modulation model for the communication system (as presented in chapter 1), consisting of the following elements:

1. A transmitted symbol sequence (data) of equiprobable *independent* binary random variables (the data) $\{u_k\}_{k=0}^{K}$ taking values in the set $\mathbb{B} = \{-1, +1\}$.

2. A communication channel (including whitened matched filter [19]) which is modelled as a causal discrete-time linear filter with *known* impulse response coefficients $\{h_i\}_{i=0}^{L}$.

3. An *iid* noise sequence $\{n_k\}$ with $n_k \sim N(0, \sigma^2)$.

The data are input to the channel which introduces intersymbol interference (ISI) and measurement noise, producing the received sequence $\{y_k\}$ given at time k by

$$y_k = \sum_{i=0}^{L} h_i u_{k-i} + n_k.$$

The equalizer (or detector) attempts to recover the data sequence $\{u_k\}$ from the measured sequence $\{y_k\}_{k=0}^{K}$. Optimal performance, in the sense of minimising the probability of incorrectly decoding the whole received sequence, would be obtained by the (MAP) detector that estimates the complete transmitted sequence, based on the entire received sequence of length K, by maximising the *a posteriori* probability [27]

$$Pr(u_0, \ldots, u_K \mid y_0, \ldots, y_{K-1})$$

over the 2^K candidate sequences $\{u_k\}_{k=0}^{K-1}$. This is clearly impractical since the computational load increases exponentially with the message length. A closely related problem is that of maximum likelihood sequence estimation (MLSE) in which the maximising sequence of inputs is sought for the conditional probability

$$Pr(y_0, \ldots, y_{K-1} \mid u_0, \ldots, u_{K-1}).$$

The MLSE and MAP problems are equivalent in the case of equiprobable input symbols. The Viterbi algorithm provides a recursive realisation of the MLSE [19], whose complexity is exponential in the channel length.

Instead of minimising the *sequential* error probability, we can move towards *symbol-by-symbol maximum* a posteriori or *minimum bit error rate detectors* that estimate the data by smoothing the measurement sequence according to

$$\max_{u_k \in \mathbb{B}} Pr(u_k \mid y_0, \ldots, y_{K-1}), \quad k = 0, \ldots, K-1, \tag{5.2.1}$$

where $\mathbb{B} \triangleq \{-1, +1\}$. The first step in this direction, an iterative algorithm proposed by Chang and Hancock [27], used a MAP criterion to estimate blocks of data (of size p) via

$$\max_{u_i \in \mathbb{B},\ k \le i \le k+p-1} Pr(u_k, \ldots, u_{k+p-1} \mid y_0, \ldots, y_{K-1}).$$

The disadvantage of such an approach is that it is an off-line technique, and the full received sequence is required for computation, even though the complexity of the algorithm increases only linearly with the message length K.

Seeking an on-line technique, we can aim at implementing a minimum bit error rate detector with a "truncated" MAP criterion, utilising only received signals up to time $k+p-1$ in estimating the symbol u_k transmitted at time k. The *fixed-delay* constraint, introduced by Abend and Fritchman [26] and Bowen [31], allows the recursive solution of the integer programming problem

$$\max_{u_k \in \mathbb{B}} Pr(u_k \mid y_0, \ldots, y_{k+p-1}), \quad k = 0, 1, 2, \ldots, \tag{5.2.2}$$

so that data estimates are obtained sequentially as a fixed-lag smoothing operation with delay $p-1$. This method, valid for finite impulse response channels, establishes a recursion on the joint probability

$$Pr(u_k, \ldots, u_{k+p-1}, y_1, \ldots, y_{k+p-1})$$

and assumes that certain channel and data dependent probabilities are calculated and stored in advance. The possibility of using past decisions to simplify the calculations was flagged in [26] and this leads us to consider another class of recently developed detectors—maximum *a posteriori* decision feedback detectors [54]. Like the block decision feedback equalizer, these combine block processing, MAP estimation and decision feedback, but also take advantage of decision error probabilities in forming their decisions. This obviates the assumption of correct decision feedback which is inherent in block DFEs. We describe these developments in the next section.

5.3 Design of the New Detector

5.3.1 Generalising the Block Decision Feedback Equalizer

We now revisit the block processing channel model used in the derivation of the block DFE (chapter 3). We require the channel model to have a finite number of states in deriving the new detection criterion. Therefore, we restrict the treatment to finite impulse response channels with state vector

$$x_k = [u_{k-L}, u_{k-L+1}, \ldots, u_{k-1}]' \in I\!B^L.$$

The input sequence $\{u_k\}$ is assumed to be binary. We remark that the method can easily be extended to M-ary or quadrature amplitude modulated signals (see section 3.5.2), but in the interests of simplicity, we restrict the presentation to binary pulse amplitude modulated signals. The scalar channel output y_k, corrupted by white Gaussian noise n_k, is generated by the system

$$\begin{aligned} x_{k+1} &= Ax_k + bu_k \\ y_k &= cx_k + du_k + n_k, \end{aligned} \tag{5.3.1}$$

for known matrices $A \in I\!R^{L\times L}$, $b, c' \in I\!R^L$ and $d \in I\!R$ given in section 3.2.3. With binary inputs there are 2^L channel states. Stacking the input and output variables, we obtain the block processing [47] representation with block size p (or decision delay

$p-1$)

$$Y_k \triangleq [y_k, y_{k+1}, \ldots, y_{k+p-1}]' = Hx_k + DU_k + N_k, \tag{5.3.2}$$

where $U_k \triangleq [u_k, \ldots, u_{k+p-1}]'$ and $N_k \triangleq [n_k, \ldots, n_{k+p-1}]'$. We thereby generate $y_k, \ldots, y_{k+p-1}$ in one step. The block processing normally advances in units of the block size. In this case, we will be considering an algorithm that generates only one data estimate at a time and hence choose to process blocks of received signals advancing in single steps (as in the sliding-window block DFE of section 3.2.5). The matrices $H \in \mathbb{R}^{p \times L}$ and $D \in \mathbb{R}^{p \times p}$ are formed from the realisation (A, b, c, d), and are given by equations (3.2.11) and (3.2.12).

The information conveyed by the measurement sequence up to time k is redundant if the state x_k of the channel is known. Since the state is not directly measurable, we assume that a state estimate $\hat{x}_k$, which we generate subsequently, is available at time k. As in section 3.2.2, from the current block of channel outputs Y_k we subtract an estimate of that portion of the intersymbol interference due to the estimated channel state $H\hat{x}_k$ (this is an estimate of the dispersion due to symbols before time k), forming the signal

$$Z_k \triangleq Y_k - H\hat{x}_k. \tag{5.3.3}$$

From (5.3.2) we can express the decision device input as

$$Z_k = DU_k + H(x_k - \hat{x}_k) + N_k. \tag{5.3.4}$$

Thus the three terms contributing to Z_k are

- A "block cursor" DU_k.
- An ISI cancellation term $H(x_k - \hat{x}_k)$.
- A vector of measurement noise terms N_k.

Since, in generating Z_k we are feeding back past decisions, the structure resembles a tuned (non-adaptive) non-linear DFE [6] which calculates decisions according to

$$\hat{u}_k = \operatorname{sgn}\left(h_0 u_k + \sum_{i=1}^{L} h_i u_{k-i} - \sum_{i=1}^{L} h_i \hat{u}_{k-i} + n_k\right). \tag{5.3.5}$$

The resemblance of the block DFE to a DFE is more than coincidental, indeed for $p = 1$ with $\hat{u}_k \triangleq \text{sgn}(Z_k)$ we recover the DFE (section 3.3.1). The $(p, 1)$-DFE (section 3.2.5) uses the signal Z_k (5.3.4), under the assumption of correct state estimation ($\hat{x}_k = x_k$), to compute the data estimate $\hat{u}_k$ according to the modified MAP criterion

$$\hat{u}_k = \underset{u_k \in I\!B}{\arg\max}\, Pr(u_k \mid Z_k, \hat{x}_k = x_k), \tag{5.3.6}$$

where $I\!B \triangleq \{-1, +1\}$. From section 3.2.5, the practical embodiment of this criterion is

$$\begin{aligned} \hat{u}_k &= \underset{u_k \in I\!B}{\arg\max} \sum_{V_k \in I\!B^{p-1}} p_N \left(Z_k - D \begin{bmatrix} u_k \\ V_k \end{bmatrix} \right) \\ &= \underset{u_k \in I\!B}{\arg\max} \sum_{V_k \in I\!B^{p-1}} \exp(- \left\| Z_k - D \begin{bmatrix} u_k \\ V_k \end{bmatrix} \right\|^2 / 2\sigma^2), \end{aligned} \tag{5.3.7}$$

where σ^2 is the variance of the white noise. As we saw in section 3.2.6, for reasonably high signal-to-noise ratios, one term dominates the sum in (5.3.7), and so the following minimum distance rule applies

$$\hat{u}_k = [\, \underbrace{1, 0, \cdots, 0}_{p} \,] \times \underset{U \in I\!B^p}{\arg\min} \|Z_k - DU\|^2. \tag{5.3.8}$$

This approximation typically incurs only a small performance penalty with respect to the optimal rule (5.3.7). Decision feedback is used to obtain the next state estimate using

$$\hat{x}_{k+1} = A\hat{x}_k + b\hat{u}_k, \; k = 0, 1, 2, \ldots, \tag{5.3.9}$$

required in forming Z_{k+1} at the next iteration (and advancing the block processing in steps of size 1). The state estimate is simply the vector of L past decisions

$$\hat{x} = [\hat{u}_{k-L}, \hat{u}_{k-L+1}, \ldots, \hat{u}_{k-1}]'.$$

The suboptimal realisation of (5.2.2) embodied in (5.3.7) thus yields a substantial simplification in computation relative to the optimal MAP detector because of the use of decision feedback as suggested by [26], with an understandable trade off in

performance.

Maximum *A Posteriori* Decision Feedback Detection

We are now in a position to investigate a further development of the block decision feedback equalizer, called the maximum *a posteriori* decision feedback detector, which ensures better performance for medium to high signal-to-noise ratios, at the expense of increased complexity. This new non-linear equalizer disposes of the assumption of correct state estimation $\widehat{x}_k = x_k$ explicit in (5.3.6), but retains the decision feedback, *i.e.*, the generation of Z_k from Y_k, that distinguishes the technique from the more elaborate minimum BER detectors of [26, 27]. The MAP decision feedback detector selects the binary u_k which maximises the conditional probability

$$\max_{u_k \in \mathbb{B}} Pr(u_k \mid Z_k), \tag{5.3.10}$$

which is equivalent to computing

$$\widehat{u}_k = \operatorname{sgn}\{Pr(u_k = +1 \mid Z_k) - Pr(u_k = -1 \mid Z_k)\}, \tag{5.3.11}$$

and forms the next state estimate using (5.3.9). We now derive the central equation characterising the decision function of the MAP decision feedback detector for an FIR(L) channel. To this end, we re-introduce the notion of an error state already encountered in the previous chapters

$$E_k \triangleq x_k - \widehat{x}_k = [e_{k-L}, e_{k-L+1}, \ldots, e_{k-1}]', \tag{5.3.12}$$

whose components are the past decision errors $e_{k-j} = u_{k-j} - \widehat{u}_{k-j}$. These take values in the set $\mathbb{E} = \{0, -2, +2\}$ in the case of binary inputs ($e_{k-j} = 0$ indicating a correct decision at time $k-j$). Hence, there are a total of 3^L different error states E_k. We now apply Bayes' rule, conditioning on the error states, to re-express the new detection criterion (5.3.10) as

$$Pr(u_k|Z_k) = \sum_{E_k \in \mathbb{E}^L} Pr(u_k|Z_k, E_k) Pr(E_k|Z_k). \tag{5.3.13}$$

Two more applications of Bayes' rule yield

$$Pr(E_k|Z_k) = Pr(Z_k|E_k)Pr(E_k)/Pr(Z_k) \tag{5.3.14}$$

and

$$Pr(u_k|Z_k, E_k) = Pr(Z_k|u_k, E_k)Pr(u_k|E_k)/Pr(Z_k|E_k). \tag{5.3.15}$$

Equally, we can rewrite the first term on the right hand side of (5.3.15) as

$$Pr(Z_k|u_k, E_k) = \sum_{V_k \in I\!B^{p-1}} Pr(Z_k|U_k = \begin{bmatrix} u_k \\ V_k \end{bmatrix}, E_k)\ Pr(V_k|u_k, E_k), \tag{5.3.16}$$

where $V_k = [u_{k+1}, \ldots, u_{k+p-1}]'$. Now, observing that the input is a sequence of *independent* random variables, u_k can be dropped from the conditioning of $Pr(V_k|u_k, E_k)$ in (5.3.16). On substitution of (5.3.16) in (5.3.15), then (5.3.15) and (5.3.14) in (5.3.13) we find, after cancelling the term $Pr(Z_k|E_k)$, that

$$\begin{aligned} Pr(u_k|Z_k) \;=\; & [Pr(Z_k)]^{-1} \sum_{E_k \in I\!E^L} Pr(u_k, E_k) \times \\ & \sum_{V_k \in I\!B^{p-1}} Pr(Z_k|U_k = \begin{bmatrix} u_k \\ V_k \end{bmatrix}, E_k)\ Pr(V_k|E_k). \end{aligned} \tag{5.3.17}$$

At this stage of the design, we are forced to make the simplifying assumption that the current inputs U_k and the noise vector N_k are statistically independent of the past decision errors E_k. In fact, in the resulting detector there will necessarily be a correlation between these quantities due to the overlap in elements of U_k and U_{k+1} and the use of decision feedback. Nonetheless, we contend that this assumption will not significantly degrade the performance of the detector. Recall that the same philosophy was adopted during the design of the (p, q)-DFE (section 3.2.5). Invoking this assumption, (5.3.17) becomes

$$\frac{Pr(u_k)}{Pr(Z_k)} \sum_{E_k \in I\!E^L} Pr(E_k) \sum_{V_k \in I\!B^{p-1}} Pr(V_k) p_N(Z_k - D \begin{bmatrix} u_k \\ V_k \end{bmatrix} - H E_k),$$

where $p_N(\cdot)$ is the p-dimensional joint Gaussian density of the noise vector N_k given in (3.2.17). Lastly, since the inputs are equiprobable and $Pr(Z_k)$ is constant in the

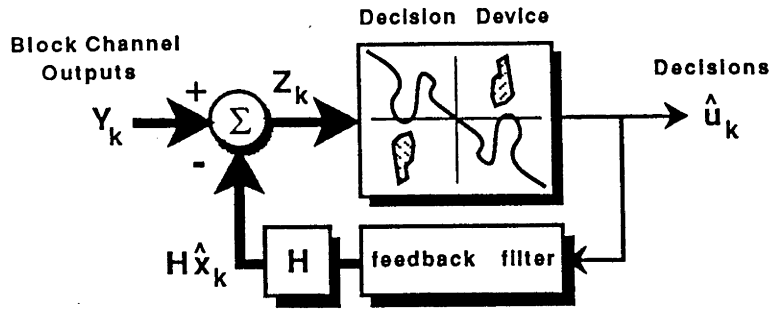

Figure 5-1: The MAP decision feedback detector.

maximisation, we can ignore their contribution, obtaining the decision rule for the new MAP decision feedback detector stated below.

Theorem 5.3.1 (**$(p,1)$-Detector**) *The memoryless decision function with input $Z_k \in \mathbb{R}^p$ and binary output $\hat{u}_k$ which optimizes the maximum* a posteriori *criterion (5.3.10) is given in terms of the 3^L steady state probabilities $Pr(E_k)$ of the error states E_k (5.3.12), for $k = 0, 1, 2, \ldots$, by*

$$\hat{u}_k = \arg\max_{u_k \in \mathbb{B}} \sum_{E_k \in \mathbb{E}^L} Pr(E_k) \sum_{V_k \in \mathbb{B}^{p-1}} e^{-\frac{1}{2\sigma^2}\left\| Z_k - D\begin{bmatrix} u_k \\ V_k \end{bmatrix} - HE_k \right\|^2}, \qquad (5.3.18)$$

where $\|x\|^2$ denotes the squared l_2-norm $x'x$, Z_k is defined in (5.3.3), $D \in \mathbb{R}^{p \times p}$ is given by (3.2.12), L is the order of the finite impulse response channel, $\mathbb{B}$ and $\mathbb{E}$ denote respectively the sets $\{-1, +1\}$ and $\{-2, 0, +2\}$, and σ^2 is the variance of the Gaussian noise.

Comparing with (5.3.7) shows how past decision errors have been incorporated into the new detection criterion. Notice that with the assumption of *correct* past decisions, $Pr(E_k \neq \underline{0}) = 0$, (5.3.18) collapses to give the $(p,1)$-DFE criterion (5.3.7).

The operation of the new detector, whose block diagram is shown in Fig.5-1, is characterised by equations (5.3.4), (5.3.9) and (5.3.18). In effect, we have generated a *family* of detectors (for a given channel and SNR), depending on the two parameters p (the block size) and L (the channel order). The device is called generically a MAP decision feedback detector. However, to distinguish particular realisations, we will refer to this device as a $(p,1)$-detector, on the understanding that L, which dimensions E_k in (5.3.18), is taken as the full channel length. The terminology is

in keeping with section 3.2.5. The data estimates $\widehat{u}_k$ are obtained by selecting the larger of two objective values, each of which is a sum of $2^{p-1}3^L$ terms. In order to lessen the computational burden, p should be chosen as the length of the significant part of the channel impulse response, while L should be viewed as an "effective channel length", *i.e.*, the effective length of the channel after the linear cancellation afforded by the decision feedback filter, although for the examples in the following subsections L is taken to be the full channel length.

5.3.2 Iterative Realisation

The $(p,1)$-detector we derived in the preceding section generates its decisions according to (5.3.18), which is a non-linear difference equation in the symbol estimates $\widehat{u}_j$, $j = k-L,\ldots,k$ (the dependence on past estimates arises from the inclusion of the error states in the detection criterion). Finding a closed-form expression for the error state probabilities $Pr(E_k)$ appears to be intractable, so an approximate method must be sought. On a given channel with known input sequence at a fixed signal-to-noise ratio, the following simple iterative procedure yields the required error probabilities and hence realises the detector. We first simulate the $(p,1)$-DFE (section 3.2.5), keeping track of the frequency of occurrence of each error state E_k. The 3^L sample values of $Pr(E_k)$ we obtain on this first run can be used as an initial estimate of the true values for (5.3.18). In practice, at reasonable signal-to-noise ratios, there will only be a slight difference between the respective (steady state) error state probabilities for the $(p,1)$-detector and the $(p,1)$-DFE, but this can result in a large difference in BER performance (see section 5.4.2). We continue the iterative process of simulating the detector, each subsequent run using the values for $Pr(E_k)$ obtained from the previous run, until convergence is observed. Further to this, the definition of the error state (5.3.12) and the statistical assumptions on the noise and input sequence to the channel guarantee the following property (which is shared by the $(p,1)$-DFE):

Property 5.3.1 (Sign Symmetry) *The transition probabilities of the finite-state Markov process for the error states E_k (5.3.12) of the MAP decision feedback detector satisfy*

$$Pr(E_{k+1} \mid E_k) = Pr(-E_{k+1} \mid -E_k)$$

Property 5.3.1 implies that at most $\frac{1}{2}(3^L - 1)$ of the 3^L steady state error probabilities need be computed. In practice, only some of the 3^L error events will have significant probability and hence only these would need to be included in the sum in (5.3.18). These effects are generally channel dependent, but may be used to simplify the decision device on a given channel. (See table 5.1 for a numerical example.) The reader is referred to appendix E.1 for comments on the structure of the finite state Markov process governing the dynamics of the error state transitions.

As an alternative to this "batch realisation" approach, one could envisage a fully recursive detection algorithm using running estimates of the error state probabilities substituted for the corresponding terms in (5.3.18). However, this would require knowledge of the transmitted data during the training phase. The adaptation of these error probability estimates could be turned off once their steady values were reached.

5.4 Performance Examples

5.4.1 First Order Channel

We take a first order channel to serve as a simple, initial illustration of the operation of the detector. Since $L = 1$, the error state $E_k = e_{k-1}$ is a scalar, so there are only 3 probabilities $Pr(E_k)$ to identify. In the steady state, these are related as follows

$$Pr(E_k = 2) = Pr(E_k = -2) = \frac{1}{2}\{1 - Pr(E_k = 0)\} = \frac{1}{2}\text{BER}.$$

So, in fact, we need only identify one parameter—the bit error rate (BER). This must be obtained by multiple simulation passes (as described in the previous section). In this example we take the channel parameters as $h_0 = 1$, $h_1 = 1.5$. This choice gives a noticeable difference in performance between the (2,1)-DFE and the $(2,1)$-detector for signal-to-noise ratios at which it is practical to simulate. On the first pass, we simulate the (2,1)-DFE, with a SNR of 4dB, over 500 000 points, then use the BER thus obtained as input to the detector (5.3.18). After two further passes, the algorithm converges (using the same data and noise sequences) to a steady state

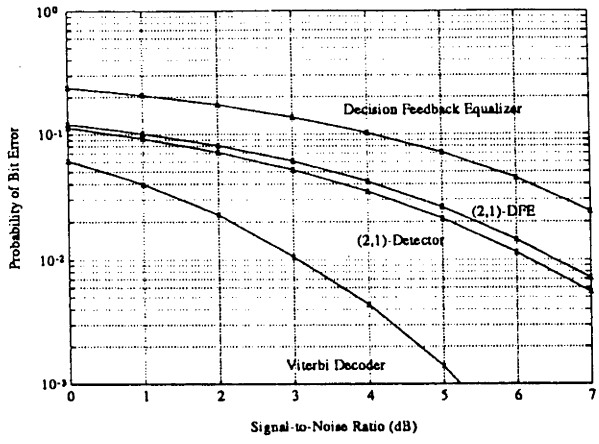

Figure 5-2: $(2,1)$-Detector performance, first order channel.

bit error rate of 0.03485. Fig.5-2 shows the BER performance of the $(2,1)$-detector in this example, along with the corresponding curves for the (2,1)-DFE with decision function (5.3.7), and the scalar DFE (3.3.1). For this channel, the detector shows an improvement of around 0.5dB over the block DFE at the low SNRs in the range of the simulation.

We can gain further understanding of the detector by considering the *decision boundary.* This is defined below (*c.f.* section 3.3.2).

Definition 5.4.1 (Detector Decision Boundary) *The set of all points* $Z_k \in \mathbb{R}^p$ *satisfying*

$$Pr(\widehat{u}_k = +1 \mid Z_k) = Pr(\widehat{u}_k = -1 \mid Z_k), \tag{5.4.1}$$

where $\widehat{u}_k$ *is obtained by (5.3.18), is called the* decision boundary *of the maximum* a posteriori *decision feedback detector.*

That is, the decision boundary consists of the set of all points Z_k for which the value of the objective (5.3.18) is the same for both $u_k = 1$ and $u_k = -1$. Note that this set need not form a closed curve in Z_k-space, as is the case for the DFE and block DFE, and we will see an example of this in the next section. Other examples of decision boundaries for non-linear equalizers can be found in [56]. The (2,1)-DFE's decision boundary is a curve whose shape depends on the channel parameters h_0 and h_1 and the SNR (section 3.3.2). On the other hand, the decision boundary of the $(2,1)$-detector will depend on h_0, h_1 and h_2 and the SNR (and the corresponding error state probabilities).

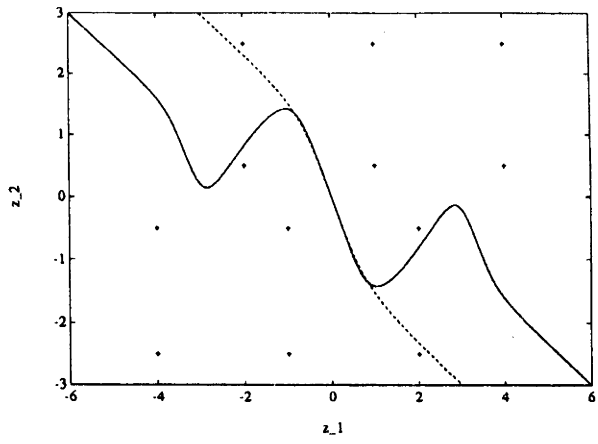

Figure 5-3: Detector decision boundary, first order channel.
The decision boundary of the (2,1)-DFE is shown dashed. The "+" signs denote the centres of the conditional probability densities.

We can examine the differences in performance between the (2,1)-DFE and the $(2,1)$-detector by looking at the decision regions formed by their respective criteria. In this simple first order case, equation (5.4.1) is amenable to numerical solution using the same technique described in section 3.3.3. The results are shown in Fig.5-3 along with the block DFE's decision boundary for $h_0 = 1$, $h_1 = 1.5$ at a SNR of 4dB (z_1 and z_2 are the first and second components of Z_k). The two decision boundaries differ significantly in the regions of low error probability, away from the centres of the conditional noise densities $p_N(N_k|E_k, u_k)$ (shown as '+' signs), but are close in the high error probability region (near the origin), which accounts for the small but significant difference in performance between the two equalizers in this example. Fig.5-4 has been included to give further clarification of the decision criterion (5.3.18). The eight conditional noise densities for non-zero decision errors are dwarfed by the four main densities (shown cut off) $p_N(N_k|E_k = 0, U_k)$, $U_k \in \mathbb{B}^2$. Note that we have depicted the densities corresponding to $u_k = -1$ (the first component of U_k) as having negative height. The decision boundary (Fig.5-3) would be the curve of zero height in this picture.

5.4.2 Second Order Channel

We now look at the $(2,1)$-detector's performance on a second order channel. Since $L = 2$ there are $3^2 = 9$ error state probabilities $Pr(E_k)$ to identify, not all of which

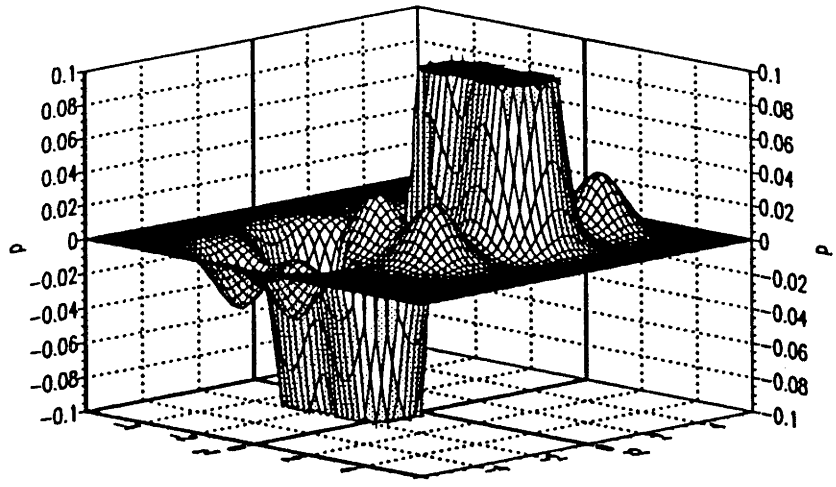

Figure 5-4: Conditional probability densities.

are independent. We can reduce this number to $\frac{9-1}{2} = 4$ on account of to the symmetry property 5.3.1. Hence, knowledge of the probabilities,

$$\mathcal{P}_1 = Pr(E_k = \begin{bmatrix} 0 \\ 2 \end{bmatrix}) \qquad \mathcal{P}_2 = Pr(E_k = \begin{bmatrix} -2 \\ 2 \end{bmatrix})$$
$$\mathcal{P}_3 = Pr(E_k = \begin{bmatrix} 2 \\ 2 \end{bmatrix}) \qquad \mathcal{P}_4 = Pr(E_k = \begin{bmatrix} 2 \\ 0 \end{bmatrix}), \tag{5.4.2}$$

would be sufficient to calculate all 9 error state probabilities and would allow us to realise the detector equation (5.3.18) for the chosen channel and signal-to-noise ratio.

At the first pass of the algorithm, the $(2,1)$-block DFE (using decision rule (5.3.7), or equivalently (5.3.18) with $Pr(E_k \neq \underline{0}) = 0$) is simulated and the above four probabilities are estimated from their sample occurrences. On subsequent passes, the detector criterion (5.3.18) is implemented, using the error probability estimates from the preceding pass. For moderate SNRs only three passes are required for convergence of the estimates to three significant figures, whereas for low SNRs more passes may be required.

The simulated BER performance of the new detector differs most markedly from that of the block DFE (at SNRs within the range of simulation) for channels with large tail energy. We take the second order non-minimum phase channel [$h_0 = 1$, $h_1 = 2$, $h_2 = 3$] as a representative example. For such a channel, the centres

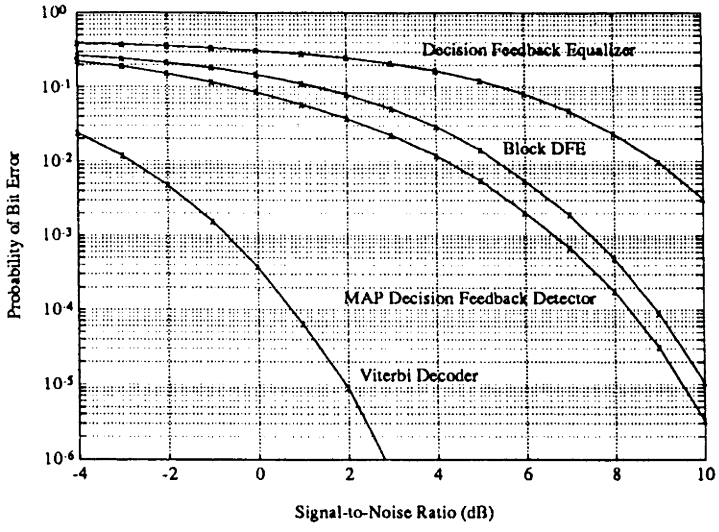

Figure 5-5: Performance comparison on the [1,2,3] channel.
Block DFE and MAP decision feedback detector implemented with $p = 2$ (and decision delay 1), Viterbi decoder decision delay 20.

of the conditional probability densities are spread out over a wide area in the Z_k plane (whereas they are more closely clustered for channels with decaying impulse responses). Even though the respective weightings of these densities are of the order of the BER for the non-zero error state terms $E_k \neq \underline{0}$ in (5.3.18), their inclusion has a significant effect on the resulting performance. For an example of the bit error rates obtained, see Table 5.1.

Fig.5-5 gives a comparison of bit error rates (simulated over 10^7 points) for the (2,1)-block DFE and $(2,1)$-detector for the $[1,2,3]$ channel. At a SNR of 4dB on this channel, the $(2,1)$-detector shows a performance improvement of 1.5dB over the (2,1)-DFE, 5dB over the conventional DFE, and falls short of the Viterbi decoder (implemented with a decision delay of 20 samples) by around 7dB. As mentioned in section 3.4, for larger values of the block size, the performance of the block DFE approaches that of Viterbi decoding [15]. Presumably the detector's gain in performance over the block DFE persists at high SNRs, although this is not easily verified either in theory or by simulation. For very low SNRs (~ -10dB) the $(2,1)$-detector and the (2,1)-DFE perform more or less identically.

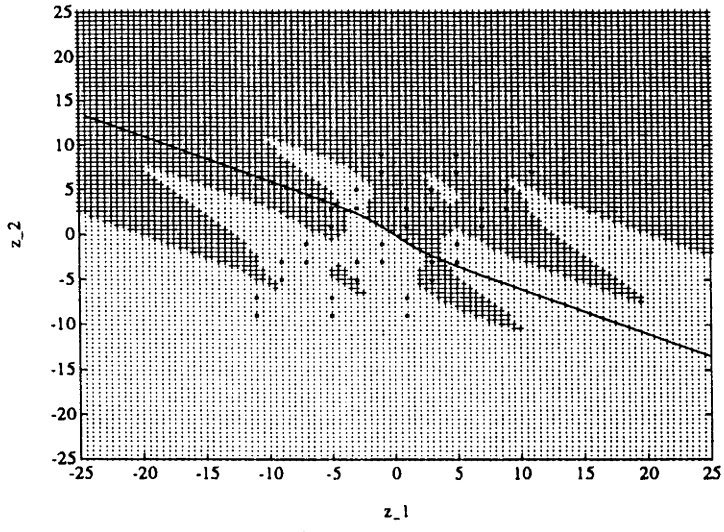

Figure 5-6: Decision boundaries for SNR=−2dB.
Simulation for [1, 2, 3] channel. (2,1)-Block DFE: solid curve; (2, 1)-detector: "+" denotes a +1 decision and "·" denotes a −1 decision. Centres of conditional probability densities for the detector are shown as "*".

Decision Regions

As seen in section 5.4.1, the form of the decision regions can yield insight into the workings of the MAP decision feedback detector. Whereas in that section the decision boundary consisted of a simple curve in Z_k-space, for a second order channel the topology of the decision regions can be much more complicated. Fig.5-6 stems from a simulation on the [1, 2, 3] channel at −2dB on which we have portrayed the (2, 1)-detector's Z_k-space as a grid of points (the x and y axes corresponding to the first and second components of Z_k respectively). The detector's output on the grid is denoted by a "+" for $\widehat{u}_k = +1$ and by a "·" for $\widehat{u}_k = -1$. The centres of the conditional error densities are marked by a star. The corresponding decision boundary for the (2,1)-block DFE is also displayed. The intervening decision regions are multiply connected—isolated regions appear in the vicinity of some of the conditional error densities (*c.f.* (5.3.18)) whose presence distorts the decision boundary.

Fig.5-7 shows the decision regions for the (2, 1)-detector on the [1, 2, 3] channel at a SNR of 8dB with the same conventions as in Fig.5-5. (The absence of points in the corners of the figure is due to numerical underflow.) In this case the required error probabilities were found after three passes and are displayed in Table 5.1. There are clear differences in the boundaries for the two devices—even in the region around

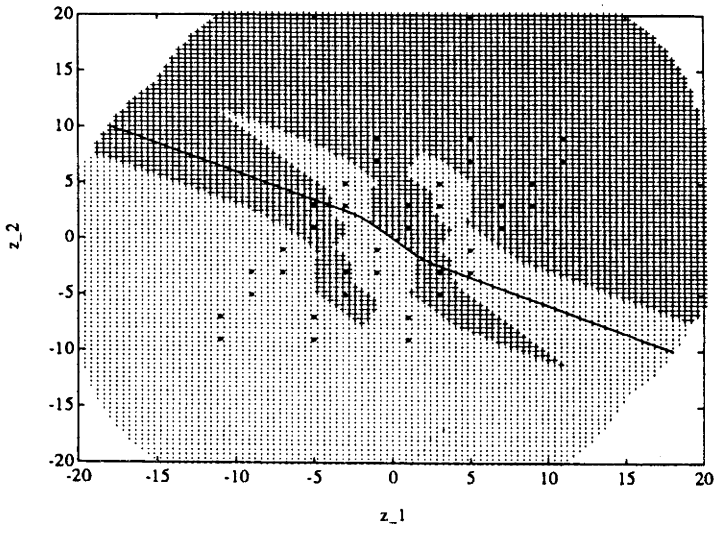

Figure 5-7: Decision boundaries for SNR=8dB.

Pass	BER	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$
1	0.490×10^{-3}	0.995×10^{-4}	0	0.146×10^{-3}	0.101×10^{-3}
2	0.178×10^{-3}	0.529×10^{-4}	0	0.359×10^{-4}	0.548×10^{-4}
3	0.178×10^{-3}	0.524×10^{-4}	0	0.368×10^{-4}	0.541×10^{-4}

Table 5.1: Error state probabilities for $(2,1)$-detector.

The symbols $\mathcal{P}_i$ are defined in equation (5.4.2).

the origin where the probability densities are relatively large. This fact goes some way to explaining the difference in performance. Note that the decision boundary of the conventional DFE in these examples is simply a vertical line through the origin.

The upshot of the example is the following. The decision region for a true fixed-delay MAP (minimum BER) detector may be a very complicated object, even for low-dimensional decision devices. The possibility of applying multi-layer neural network techniques (or other approximate functional representations) to the problem of implementing the decision device comes to mind. The corresponding problem for the block DFE would need to be tackled first. As stated in section 3.3, the material in chapter 2 concerning the conventional DFE could serve as an initial step in this direction. The application of such techniques in non-adaptive equalization has already been investigated in [50] (obtaining isolated decision regions for MLP equalizers operating on non-linear FIR channels). However, an advantage of the current scheme is the use of decision feedback to provide cancellation of the significant intersymbol interference before vector quantization.

5.5 Conclusions and Discussion

The MAP decision feedback detector developed in this chapter, which combines (5.3.4), (5.3.9) and (5.3.18), represents an enhancement and generalization of the $(p,1)$-DFE comprising equations (5.3.4), (5.3.9) and (5.3.7). The simplification of the MAP detection criterion (5.2.2) due to the use of decision feedback, was exploited in both devices to reduce the dimension of the space on which the decisions $\widehat{u}_k$ are formed (*i.e.*, $\dim(Z_k) = p$ as compared with $\dim\{y_0, \ldots, y_{k+d}\} = k+p$). These new decision procedures are essentially classification algorithms in the sense that the appropriate (Z_k)-space is partitioned by the decision rule into sets corresponding to $\widehat{u}_k = \pm 1$.

The difference between the $(p,1)$-DFE (chapter 3) and the MAP decision feedback detector arises from the different decision rules: (5.3.7) and (5.3.18). The block DFE's criterion has a relatively simple closed form, whereas (5.3.18) represents a superior rule or partition on Z_k-space which requires numerical evaluation (since the behaviour of the error state probabilities with SNR is not easy to model in general),

but is more closely related to the fixed-delay, symbol-by-symbol minimum bit error rate criterion (5.2.2). At the design stage, the independence of certain variables had to be assumed in arriving at an implementable form of the $(p,1)$-detector's decision rule. The same approximations were necessary in deriving the sliding-window block DFE's decision rule, so this does not come as a surprise. What is important, and has been achieved here, is to extract a decision function that can be realised on-line from the more ambitious minimum bit error rate criterion (5.2.1).

Unlike the block DFE, an adaptive MAP decision feedback detector could not operate in a blind (decision-directed) mode. This is because the (steady state) probabilities $Pr(E_k)$ of the error states E_k (5.3.12) are required in the decision device, and the data (training) sequence must be known to establish and/or update these.

The asymptotic performance of the MAP decision feedback detector for high signal-to-noise ratios needs to be investigated. As we mentioned in section 5.4.2, the detector's behaviour for high SNRs is not easy to ascertain even by numerical simulation owing to the extremely small error probabilities that are involved. It is an open problem as to whether the detector increases its performance gain over the block DFE for higher SNRs or becomes equivalent to the latter.

It is not clear that the $(p,1)$-detector's criterion (5.3.18) gives the minimum probability of error over the class of detectors with criteria of the form

$$\widehat{u}_k = \arg\max_{u_k \in \mathbb{B}} \sum_{E_k \in \mathbb{E}^L} p(E_k) \sum_{V_k \in \mathbb{B}^{p-1}} e^{-\frac{1}{2\sigma^2}\left\| Z_k - D\left[\begin{smallmatrix} u_k \\ V_k \end{smallmatrix}\right] - HE_k \right\|^2},$$

for some weights $p(E_k) \in [0,1]$. This suspicion arises from the assumption that the conditional probability densities are Gaussian, which may in some cases be a bad approximation. Perhaps some tweaking of the weights $p(E_k)$ in the above equation, away from their nominal values $p(E_k) = Pr(E_k)$ for the $(p,1)$-detector, would yield a further reduction in the bit error rate.

It should be stressed that the generic complexity of the MAP decision feedback detector is of order $2^{p-1}3^L$, where p is the output dimension of the feedback filter and L is the effective channel length (of the remaining ISI). A similar exponential complexity problem is also encountered in both [26] and [19]. However in a MAP decision feedback detector, this complexity is confined solely to the decision function

(5.3.18) which, conceivably, could be replaced by a simpler feedforward non-linear processor, for instance an MLP neural network [40], while maintaining good performance. It seems that added complexity is the price we pay for achieving low bit error rates with only a small decision delay. The alternative to this is sequential decoding, such as the Viterbi algorithm, which requires a large delay (and can still suffer from a complexity problem).

In flagging the possible application of artificial neural networks here, we are alluding to the capacity for functional representation of these devices. It is known that a three-layer perceptron neural network can classify arbitrary disconnected regions of its input space [42], and something akin to this is what is needed to implement the MAP decision feedback detector's decision function. The reader should be aware that one of the major problems associated with MLP neural networks is their training. Certainly, back propagation training (a gradient descent strategy) [50] has its drawbacks in terms of the size of the data set required for adaptation of the neural network parameters and the time required for convergence. Unlike the emulation of the decision feedback equalizer by a non-linear feedforward processor (chapter 2), there is no obvious connection between the MAP decision feedback detector and a feedforward neural network, although the block DFE's decision function (at least for small block sizes—*e.g.*, (3.3.3)) is reminiscent of a neural network with hard-limiting nodes.

Chapter 6

Conclusions and Further Work

6.1 Further Work

We wrap up the thesis in this section with a discussion of some possible extensions and further work, sparing the reader from a restatement of the conclusions we have already presented in the respective chapters.

Feedforward Emulation of the DFE

We flag here some open problems concerning the work in chapter 2. Firstly, for the non-adaptive feedforward emulator, the calculation of a bound primary error probability needs to be addressed. This entails the inclusion of noise into the finite-state Markov process model of section 2.3. The aggregated state description would likely lead to a very conservative bound. The alternative to this is the computation of the full order 5^L transition matrix (L is the channel order), whose elements are dependent on the particular channel as well as the signal-to-noise ratio. It should be provable that the error probability of the feedforward emulator is lower bounded by the corresponding (tuned) DFE's error probability.

Next, there are questions concerning non-tuned performance. The effect of static mistuning (the departure of the weights from their ideal values) could be examined. The motivating idea here is the need to test the robustness of the system, or its sensitivity to changes in parameters. The adaptive performance of the feedforward emulator could be more fully examined. Theoretical results such as local stability or convergence of the adaptive algorithm might not be difficult to obtain and the

question of existence of local (*e.g.*, delay-type[1]) equilibria could equally be investigated. Delay-type equilibria of the FFE under the sign algorithm have already been observed in simulations. These considerations are essentially questions about the shape of the mean cost function surface [25].

Another possibility is the use of blind adaptive algorithms for the FFE, as suggested by existing modulus-restoral algorithms for the DFE [9]. A possible blind adaptive error measure for binary signalling is

$$J_k = \frac{1}{2}\{\widetilde{u}_k - \mathrm{sgn}(\widetilde{u}_k)\}^2,$$

where $\widetilde{u}_k$ is the output of a FFE with sigmoid nodes. We take the opportunity here to mention a interesting, but as yet unexploited, connection between neural network-like structures with hard and soft-limiting (sigmoid) nodes. If we include a random dither signal n as an input to a hard-limiting node, then by choosing the signal's probability density $p_n(\cdot)$ appropriately, we can make the action of such a node sigmoidal on average. For instance, suppose we want to choose $p_n(\cdot)$ such that

$$\mathcal{E}\{\mathrm{sgn}(x+n)\} = \frac{1}{1+e^{-x}}.$$

Then it is easily shown that the random variable n must have a probability density given by

$$p_n(\nu) = \frac{e^{-\nu}}{(1+e^{-\nu})^2} = \frac{d}{d\nu}\frac{1}{1+e^{-\nu}}.$$

A broader line of enquiry is the application of FSMP modelling techniques to standard MLP neural networks, although, without the special structure inherent in the FFE, it is harder to see how one would progress. What is needed in this area are solid results, analogous to those of chapter 2, that justify the use of neural network techniques in equalization.

Extensions to Block Decision Feedback Equalization

The work in chapter 5 pointed to the possibility of improving the performance of block DFEs by modifying the decision device. We already mentioned the need for

[1] An equilibrium of the form $\widetilde{u}_k = u_{k-\delta}$ for some $\delta > 0$.

research into simplified vector quantizers that maintain good performance for the MAP decision feedback detector. The same comment applies equally to the block DFE. Indeed, it may be feasible to apply low dimensional block DFEs to channels having long impulse responses, thereby obviating complexity problems, although this is something which needs to be tested by simulation.

The complexity involved in designing minimum bit error rate detectors need not preclude the realisation of implementable devices that achieve similar performance. The training of MLP neural networks to replace more computationally intensive decision devices is another topic worthy of investigation. An interesting heuristic approach might be to try and train a multi-layer neural network decision device as a Viterbi decoder.

The important area of coding should be researched in relation to block decision feedback equalization. We noted earlier that the device is naturally suited to the use of block codes [20]. We also saw in section 3.5.2 that quadrature amplitude modulation signalling with independent data could be incorporated into the block DFE framework. However, the problem of how to modify a block DFE for the important practical technique of trellis-coded modulation [2] remains open.

As far as analysis of block DFE performance is concerned, techniques for bounding the error probability (including the effects of error propagation) need to be devised that do not rely on explicit knowledge of the decision boundary. Since, as mentioned in section 3.3.4, finding expressions for the decision rule of even the $(3,1)$-DFE appears to be a hard problem. Bounding the block DFE's primary error probability should be possible using trellis-based techniques [13, 14], at least for high signal-to-noise ratios. If a way for obtaining explicit formulae for the block DFE's decision rule can be found, then a recursive unwrapping procedure, similar to the one used in generating the FFE of chapter 2, could be applied to obtain a specific feedforward structure which may in turn suggest other appropriate neural network architectures for equalization.

We seize[2] this opportunity to suggest the possibility of applying input-output stability techniques (*e.g.*, Lyapunov stability, passivity theory) to analyse block DFE error recovery (in the sense of section 4.5). Preliminary results have already been ob-

[2]French for sixteen—another lucky number.

tained in this area—in particular, the stability of the high SNR (2,1)-DFE on a first order channel has been treated using passivity arguments (similar to the treatment of the DFE in [16, 24]). These arguments, however, seem quite difficult to generalise to higher order block DFEs and rely on knowledge of the decision boundary which might not be obtainable.

In the immediate future, the topics in section 3.5 concerning extensions to block decision feedback equalization should be investigated numerically. More specifically, the effect of coloured noise on the BER performance of the block DFE could be compared by simulation with the Viterbi algorithm. Although the Viterbi decoder is the optimal equalization system for white noise, it suffers a degradation in performance in the presence of coloured noise. It would be interesting to see if the block DFE's performance is as sensitive to noise colouration and whether the decision criterion can be modified in the way described in section 3.5.3 to allow for this situation.

Using the theoretical framework developed in sections 3.2.1 and 3.2.2, it is straightforward to devise recursive realisations of block DFEs for ARMA (IIR) channels. In the interests of completeness, some numerical examples of block DFE performance on ARMA channels should be catalogued. From an analysis point of view, however, the FSMP-based analysis adopted in chapter 4 (dealing with FIR channels) would need extension, since the state space is generally a continuum in the ARMA case.

The application of block processing to the equalization of non-linear channels (section 3.5.1) also seems worthy of investigation via simulation, as non-linear channels do arise in practice. The techniques of that section may be a good starting point for the application of block DFEs to non-linear modulation schemes like frequency shift keying (FSK) [20] in which the transmitted signal is a non-linear function of the data.

Perhaps the most important aspect for further work is the subject of adaptation in block DFEs. This is a major practical concern, and the algorithms derived in section 3.5.4 need to be tested numerically to establish the practical feasibility of block decision feedback equalization. Such experiments could also indicate the block DFE's sensitivity to mistuning of parameters in its feedback filter and deci-

sion device (*i.e.*, its robustness). In addition, algorithms for blind adaptation and statistical output tests for convergence could be tried, building on the existing techniques for DFEs [25, 66]. Although the theoretical treatment of such topics would tend to be rather difficult, as many open problems of this nature remain even for the conventional decision feedback equalizer.

While on the subject of adaptation, we recall the trellis interpretation of the block DFE (section 3.4.2) which establishes a direct link between the latter and the Viterbi decoder and the other Viterbi-based schemes of [10, 11, 13, 14]. In the FIR case at least, it should be possible to make these trellis-based equalizers adaptive by incorporating a stochastic gradient descent algorithm (based on the instantaneous cost or metric) to update the channel estimates. Preliminary investigations have revealed the feasibility of this approach for adaptive Viterbi decoding and M-algorithm decoding when a good enough initial estimate of the channel is available. Similar work on adaptive Viterbi decoding has also been undertaken in [67].

Venturing even further afield, we can contemplate the marriage of block decision feedback equalization and the reduced-state sequence estimation schemes of [11, 10]. In the same way that the RSSE is a structured simplification of Viterbi decoding, the block DFE is a structured generalisation of the decision feedback equalizer, and both approaches have the same extremes of performance. We appeal to the use of decision feedback in the RSSE to make the tentative suggestion that this mechanism may be amenable to block processing and vector quantization (as found in the block DFE). In so doing, it may be possible to develop adaptive nonlinear equalizers that are computationally simpler than the Viterbi decoder and yet yield comparable performance. We conclude this thesis with the observation that such a marriage would indeed be a happy one.

Bibliography

[1] C. E. Shannon, and W. Weaver, *The Mathematical Theory of Communication*, Univ. of Illinois Press, Chicago, 1963.

[2] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals", *IEEE Trans. Info. Th.*, vol. IT-28, no. 1, pp. 55-67, Jan. 1982.

[3] G. D. Forney Jr., "The Viterbi Algorithm", *Proc. IEEE*, vol. 61, no. 3, pp. 268-278, Mar. 1973.

[4] N. Seshadri, C.-E. Sundberg, "Generalized Viterbi Algorithms for Error Detection with Convolutional Codes, *IEEE GLOBECOM'89*, pp. 1534-1537, 1989.

[5] R. W. Lucky, "A Survey of the Communication Theory Literature: 1968-1973", *IEEE Trans. Inform. Theory*, Vol. IT-19, pp. 725-739, Nov. 1973.

[6] S. U. H. Qureshi, "Adaptive Equalization", *Proc. IEEE*, vol.73, pp.1349-1387, Sept. 1985.

[7] R. A. Howard, *Dynamic Probabilistic Systems: Markov Models*, John Wiley, N.Y., 1971.

[8] L. Isaacson, and R. W. Madsen, *Markov Chains, Theory and Applications*, John Wiley and Sons Inc., 1985.

[9] J. R. Treichler, and M. G. Larimore, "New Processing Techniques Based on the Constant Modulus Adaptive Algorithm", *IEEE Trans. ASSP*, ASSP-33, pp. 420-431, April 1985.

[10] A. Duel-Hallen, and C. Heegard, "Delayed Decision-Feedback Sequence Estimation", *IEEE Trans. Commun.*, Vol. COM-37, No.5, pp.428-436, May 1989.

[11] M. V. Eyuboğlu, and S. U. H. Qureshi, "Reduced-State Sequence Estimation with Set Partitioning and Decision Feedback", *IEEE Trans. Commun.*, Vol. COM-36, No.1, pp.13-20, Jan. 1988.

[12] G. J. Foschini, "A Reduced State Variant of Maximum Likelihood Sequence Detection Attaining Optimum Performance for High Signal-to-Noise Ratios", *IEEE Trans. Inform. Theory*, Vol. IT-23, no. 5, Sept. 1977.

[13] J. J. Moon, and L. R. Carley, "Fast Sequence Detection for High Density Storage Channels", submitted to *IEEE Trans. Commun.*, 1989.

[14] J. G. Proakis, and A. Khazen-Terezia, "A Decision-Feedback Tree-Search Algorithm for Digital Communication through Channels with Intersymbol Interference", *Proc. ICC'86*, Toronto, CA, pp.657-661, June 1986.

[15] D. Williamson, R. A. Kennedy, and G. W. Pulford, "Block Decision Feedback Equalization", *IEEE Transactions on Communications*, February 1992, to appear.

[16] A. Cantoni, and P. Butler, "Stability of Decision Feedback Inverses", *IEEE Trans. Commun.*, Vol. COM-24, pp.970-977, Sept. 1976. Also: Univ. Newcastle Tech. Report EE7505, Mar. 1975.

[17] D. L. Duttweiler, J. E. Mazo, and D. G. Messerschmitt, "An Upper Bound on the Error Probability in Decision Feedback Equalization", *IEEE Trans. Inform. Theory*, Vol. IT-20, pp.490-497, July 1974.

[18] A. M. de Oliveira Duarte, and J. J. O'Reilly, "Simplified Techniques for Bounding Error Statistics for DFB Receivers", *Proc. F, Commun., Radar and Signal Process.*, vol. 132, no. 7, pp. 567-575, 1985.

[19] G. D. Forney Jr., "Maximum Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference", *IEEE Trans. Inform. Theory*, Vol. IT-18, pp. 363-378, May 1972.

[20] E. A. Lee, and D. G. Messerschmitt, *Digital Communications*, Kluwer Academic Press, Boston 1988.

[21] R. A. Kennedy, G. W. Pulford, B. D. O. Anderson, and R. R. Bitmead, "When Has a Decision-Directed Equalizer Converged ?", *IEEE Transactions on Communications*, Vol. 37, no. 8, pp. 879-884, Aug. 1989.

[22] M. S. Mueller, and J. Salz, "A Unified Theory of Data-Aided Equalization", *Bell Syst. Tech. J.*, vol. 60. no. 9, pp. 2023-2038, Nov. 1981.

[23] R. A. Kennedy, and B. D. O. Anderson, "Recovery Times of Decision Feedback Equalizers on Noiseless Channels", *IEEE Trans. Commun.*, Vol. COM-35, pp.1012-1021, Oct. 1987.

[24] R. A. Kennedy, B. D. O. Anderson, and R. R. Bitmead, "Channels leading to rapid error recovery for decision feedback equalizers", *IEEE Trans. Commun.*, vol. COM-37, pp. 1146-1155, Nov. 1989.

[25] R. A. Kennedy, B. D. O. Anderson, and R. R. Bitmead, "Stochastic Dynamics of Blind Decision Feedback Equalizer Adaptation", *Proc. IFAC Symposium on Adaptive Systems in Contr. and Signal Processing*, pp. 65-70, Glasgow, April 1989.

[26] K. Abend, and B. D. Fritchman, "Statistical Detection for Communication Channels with Intersymbol Interference", *Proc. IEEE*, Vol. 58, pp. 779-785, May 1970.

[27] R. W. Chang, and J. C. Hancock, "On Receiver Structures for Channels having Memory", *IEEE Trans. Inform. Theory*, Vol. IT-12, pp. 463-468, Oct. 1966.

[28] R. A. Gonsalves, "Maximum-likelihood Receiver for Digital Data Transmission", *IEEE Trans. Commun. Technol.*, Vol. COM-16, pp.392-398, June 1968.

[29] J. F. Hayes, T. M. Cover, and J. B. Riera, "Optimal Sequence Detection and Optimal Symbol-by-symbol Detection: Similar Algorithms", *IEEE Trans. Commun.*, Vol. COM-30, pp. 152-157, Jan. 1982.

[30] K. Abend, T. J. Harley, Jr., B. D. Fritchman, and C. Gumacos, "On Optimum Receivers for Channels having Memory", *IEEE Trans. Inform. Theory* (Corresp.), Vol. IT-14, pp. 818-819, Nov. 1968.

[31] R. R Bowen, "Bayesian Decision Procedure for Interfering Digital Signals", *IEEE Trans. Inform. Theory* (Corresp.), Vol. IT-15, pp. 506-507, July 1969.

[32] A. J. Viterbi, "Error Bounds for Convolutional Codes an and Asymptotically Optimum Decoding Algorithm", *IEEE Trans. Info. Th.*, vol. IT-13, pp. 260-269, Apr. 1967.

[33] R. Bellman, and R. Kalaba, "On the Role of Dynamic Programming in Statistical Communication Theory", *IRE Trans. Info. Th.*, vol. IT-3, pp. 197-203, Sept. 1957.

[34] A. P. Clark, M. Slater, and K. Parama Raj, "Simple Nonlinear Equalisers for Binary Baseband Signals", *IEE Proc.*, Vol. 130, Pt. F, no. 6, pp. 495-505, Oct. 1983.

[35] D. D. Falconer, and F. R. Magee, "Adaptive Channel Memory Truncation for Maximum Likelihood Sequence Estimation", *Bell Syst. Tech. J.*, Vol. 52, no. 9, pp. 1541-1562, Nov. 1973.

[36] A. Gersho, and T. L. Lim, "Adaptive Cancellation of Intersymbol Interference for Data Transmission", *Bell Syst. Tech. J.*, vol. 60, no. 11, pp. 1997-2021, Nov. 1981.

[37] W. U. Lee, and F. S. Hill, "A Maximum Likelihood Sequence Estimator with Decision-Feedback Equalization", *IEEE Trans. Commun.*, vol. COM-25, No. 9, Sept. 1977.

[38] S. U. H. Qureshi, and E. E. Newhall, "An Adaptive Receiver for Data Transmission over Time-Dispersive Channels", *IEEE Trans. Info. Th.*, vol. IT-19, no. 4, Jul. 1973.

[39] S. U. H. Qureshi, "An Adaptive Decision-Feedback Receiver using Maximum Likelihood Sequence Estimation", *Proc. Int. Conf. Commun.*, vol. 14, pp. 10-15, 1973.

[40] R. P. Lippmann, "An Introduction to Computing with Neural Nets", *IEEE ASSP Mag.*, pp. 4-22, Apr. 1987.

[41] T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1980.

[42] Yoh-Han Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, Mass., 1989.

[43] A. P. Clark, L. H. Lee, and R. S. Marshall, "Developments of the Conventional Nonlinear Equaliser", *IEE Proc.*, Vol. 129, Pt. F, no. 2, pp. 85-94, Apr. 1982.

[44] H. C. Guren, and D. Williamson, "Channel Equalization in the Presence of Parameter Errors", *Proc. 1989 Australian Symposium on Signal Processing and Applications, ASSPA'89*, Adelaide Australia, pp.196-200, Apr. 1989.

[45] C. Burrus, "Block Implementation of Digital Filters", *IEEE Trans. Circuit Theory*, Vol. CT-18.pp. 697-701, Nov. 1971.

[46] C. Burrus, "Block Realization of Digital Filters", *IEEE Trans. Audio Electroacoust.*, Vol. AU-20, pp. 230-235, Oct. 1972.

[47] S. K. Mitra, and R. Gnanasekaran, "Block Implementation of Recursive Digital Filters—New Structures and Properties", *IEEE Trans. Circuit and Syst.*, Vol. CAS-25, pp. 200-207, Apr. 1978.

[48] D. Clements, and B. D. O. Anderson, "A Nonlinear Fixed-Lag Smoother for Finite State Markov Processes", *IEEE Trans. Info. Th.*, vol. IT-21, no. 4, Jul. 1975.

[49] S. Benedetto, E. Biglieri, and V. Castellani, *Digital Transmission Theory*, Prentice-Hall Inc., Englewood Cliffs, N.J., 1987.

[50] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Adaptive Equalization of Finite Non-Linear Channels Using Multilayer Perceptrons", *Signal Processing*, Vol. 20,pp. 107-119, Elsevier, 1990.

[51] P. A. Galko, and S. Pasupathy, "Optimization of Linear Receivers for Data Communication Signals", *IEEE Trans. Info. Th.*, vol. 34, no. 1, Jan. 1988.

[52] G. J. Gibson, S. Siu, and C. F. N. Cowan, "Application of Multilayer Perceptrons as Adaptive Channel Equalisers", *IFAC Adaptive Systems in Control and Signal Processing*, Glasgow, 1989.

[53] G. W. Pulford, R. A. Kennedy, D. Williamson, and B. D. O. Anderson, "Error Analysis of the Block DFE using Finite State Markov Processes", *Proc. The Second Int. Sym. on Signal Processing and its Applic., ISSPA'90*, Vol. 2, pp. 687-690, Gold Coast, Australia, Aug. 1990.

[54] G. W. Pulford, and R. A. Kennedy, "Maximum A Posteriori Decision Feedback Detection", *Proc. Int. Sym. on Information Theory and its Applic., (ISITA'90)*, Waikiki Hawaii, pp. 287-290, Nov. 1990.
Also submitted to *IEE Proceedings Part I Communications, Speech and Vision*, Nov. 1991.

[55] R. A. Kennedy, B. D. O. Anderson, and R. R. Bitmead, "Tight Bounds on the Error Probabilities of Decision Feedback Equalizers", *IEEE Trans. Commun.*, vol. COM-35, pp. 1022-1028, Oct. 1987.

[56] G. J. Gibson, S. Siu, and C. F. N. Cowan, "The Application of Nonlinear Structures to the Reconstruction of Binary Signals", *IEEE Trans. Signal Proc.*, Vol. 39, No. 8, pp. 1877-1884, Aug. 1991.

[57] G. W. Pulford, R. A. Kennedy, and B. D. O. Anderson, "A Neural Net Structure for Decision Feedback Equalisers", *Proc. Second Aust. Conf. Neural Networks ACNN'91*, pp. 223-226, Sydney, Australia, Feb. 1991.

[58] G. W. Pulford, R. A. Kennedy, and B. D. O. Anderson, "A Neural Net Structure for Emulating Decision Feedback Equalisers", *Proc. Int. Conf. Acoust. Speech Sig. Process. ICASSP'91*, Toronto, Canada, Vol.3, pp. 1517-1520, May 1991.

[59] G. W. Pulford, R. A. Kennedy, and B. D. O. Anderson, "A Neural Network Structure for Decision Feedback Equalization", submitted to *IEEE Transactions on Information Theory*, July 1991.

[60] F. R. Gantmacher, *The Theory of Matrices*, Vol. 2, Chelsea Publishing Company, N.Y., 1974.

[61] C. R. Johnson Jr., *Lectures on Adaptive Parameter Estimation*, Prentice Hall, Englewood Cliffs, NJ 07632.

[62] R. Rosenblatt, *Principles of Neurodynamics*, Spartan Books, N.Y., 1959.

[63] A. P. Clark, "Adaptive Detection With Intersymbol-Interference Cancellation for Distorted Digital Signals", *IEEE Trans. Commun.*, Vol. COM-20, pp.350-361, June 1972.

[64] G. H. Golub, C. F. Van Loan, *Matrix Computations*, J. Hopkins Univ. Press, Maryland, 1990.

[65] R. R. Bitmead, M. Gevers, and V. Wertz, "Adaptive Optimal Control: The Thinkings Man's GPC", Prentice Hall, Sydney 1990.

[66] A. Benveniste, "Recursive Blind Equalizers", *IEEE Trans. Comms.*, COM-32, pp. 871-883, Aug. 1984.

[67] N. Seshadri, "Joint Data and Channel Estimation Using Blind Trellis Search Techniques", preprint submitted to *IEEE Trans. Commun.*, 1991.

Appendix A

Appendix To Chapter 1

A.1 Trellis Interpretation of Viterbi Algorithm

The Viterbi algorithm for the MLSE is easily visualised in terms of a M^L-state *trellis* [3] where M is the alphabet size and L the channel order. In the case of binary signalling ($M = 2$), only two transitions are possible from each trellis state at each time instant, as depicted in Fig.A-1 for a second order channel. The inputs u_{k-1} causing the state transitions $x_{k-1} \rightarrow x_k$ are marked on the branches (x_k defined in section 1.1.3). Rows in the trellis correspond to states and columns index the time. Each path through the trellis corresponds with a unique input sequence. The Gaussian and Markov properties of channel model allow the total cost of a path to be expressed as the sum of costs (log likelihoods) of its individual transitions. This "incremental cost" is the squared distance with respect to the noisy received signal. The algorithm extends paths from each state according to the transition which incurs the minimum incremental cost.

Not all the paths through the trellis survive the minimisation process. The path

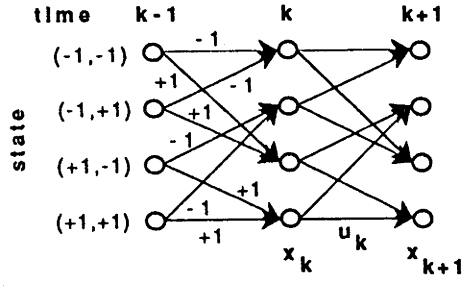

Figure A-1: Binary 4-State Viterbi Trellis.

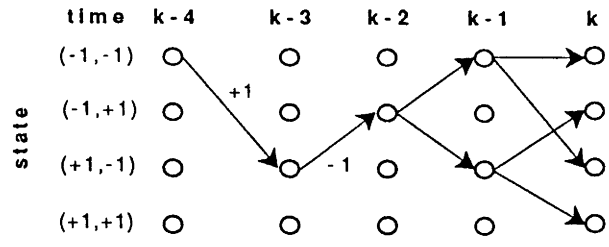

Figure A-2: Merging.

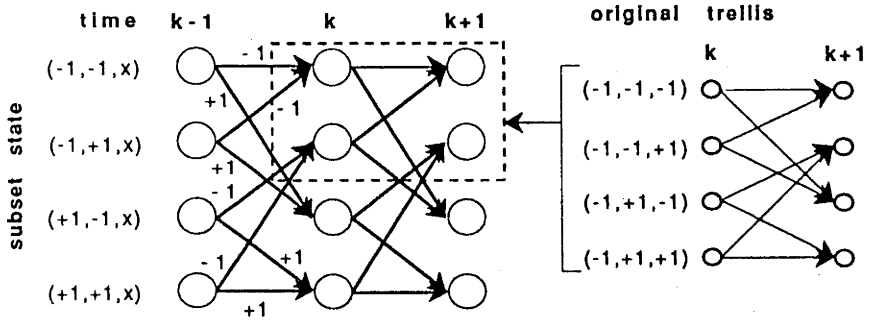

Figure A-3: Trellis with 4 Subset States.

terminating at a particular state at time k is called the *survivor* for that state. The *path history* is the sequence of states that a path has moved through. Whenever two paths meet at the same state, the path with the minimum total cost (calculated from its path history) is selected. The various paths leading to states x_k at time k will generally share a common initial portion terminating at some state $x_{k-\delta}$ ($\delta > 0$), after which some diverge. This configuration is known as *merging* and is depicted in Fig.A-2 for $\delta = 2$ on the 4-state binary trellis. A merged path corresponds to a subsequence of maximum likelihood decisions ($u_{k-4} = 1$, $u_{k-3} = -1$ in Fig.A-2). Merging is a nondeterministic phenomenon (see ref 34 in [19]) and results in an unbounded variable delay δ in the output of (groups of) MLSE decisions. In practice, a maximum decision delay (usually several times the channel length) is fixed by design. This may incur a small performance penalty with respect to MLSE detection with infinite delay.

A.2 Reduced-State Sequence Estimation

At the heart of the decision feedback sequence estimation (DFSE) and reduced-state sequence estimation (RSSE) schemes is the idea of a *subset state*. The Viterbi trellis

has M^L states for m-ary signalling. In [11] rules are given that allow the definition of a *hierarchy of partitions* of the set of channel states and signal alphabet. The resulting subset states retain the state transition properties of a trellis. So, given any subset state and knowing which of the finest partitions of the signal set contains the current symbol (estimate), the transition to the next subset state is uniquely determined. An example DFSE subset trellis for $M = 2$, $L = 3$ is shown in Fig.A-3. In this example, the full trellis has 8 states and the reduced trellis has 4 states. The inset (expanded on the right of the figure) shows how the reduction has been achieved. For instance, states $(-1,-1,-1)$ and $(-1,-1,1)$ have been aggregated into the $(-1,-1,$x$)$ subset state, where x is either 1 or -1.

The cost of a transition is the same as in the Viterbi algorithm, and is obtained from knowledge of the current input symbol estimate and the L most recent past estimates, which are stored in the path history of the survivor sequence to the destination subset state. Parallel transitions between states within the same subset transition are resolved by choosing the privileged state in the subset which minimises the cost metric for that transition. Decision feedback enters into the metric calculation by the use of past decisions in place of the true data, but the overall algorithm, in a manner of speaking, can be made to incorporate more of the Viterbi trellis and less decision feedback.

By judicious choice of the partitions that determine the subset states, a gradual tradeoff of performance for complexity is achievable, ranging from the stock DFE—the coarsest partition (a one subset state trellis) to the MLSE—the finest partition (the full M^L-state trellis).

The decision feedback sequence estimator [10] is a special case of RSSE. A reduction in the number of states in the Viterbi trellis is obtained by *aggregating* states whose first b $(1 \leq b \leq L)$ components are identical (for binary signalling states with the same b most significant bits are aggregated). The truncated states form a trellis (Fig.A-3 depicting the $b = 2$ case). Each subset transition is represented by its state transition of minimum metric (corresponding to a decision). Paths in the reduced order trellis are compared on the basis of metrics of these privileged transitions.

Appendix B

Appendix To Chapter 2

B.1 Proof of Lemma 2.5.1

We calculate the derivatives in the order $i = \Lambda - j, \Lambda - j - 1, \ldots, 2, 1$.

For $i = \Lambda - j$:

$$\begin{aligned}
\frac{\partial \tilde{u}}{\partial w_{\Lambda-j,\Lambda}} &= \frac{\partial f(\sum_{l=1}^{\Lambda-1} w_{l,\Lambda} y_l^{\Lambda-1} + y_k)}{\partial w_{\Lambda-j,\Lambda}} \\
&= \phi(y^\Lambda) \sum_{l=1}^{\Lambda-1} y_l^{\Lambda-1} \frac{\partial w_{l,\Lambda}}{\partial w_{\Lambda-j,\Lambda}} \\
&= \phi(y^\Lambda) y_{\Lambda-j}^{\Lambda-1}.
\end{aligned}$$

For $i = \Lambda - j - 1$:

$$\begin{aligned}
\frac{\partial \tilde{u}}{\partial w_{\Lambda-j-1,\Lambda-1}} &= \phi(y^\Lambda) \frac{\partial w_{\Lambda-1,\Lambda} y^{\Lambda-1}}{\partial w_{\Lambda-j-1,\Lambda-1}} \\
&= \phi(y^\Lambda) w_{\Lambda-1,\Lambda} \frac{\partial f(\sum_{l=1}^{\Lambda-2} w_{l,\Lambda-1} y_l^{\Lambda-2} + y_{k-1})}{\partial w_{\Lambda-j-1,\Lambda-1}} \\
&= \phi(y^\Lambda) w_{\Lambda-1,\Lambda} \phi(y^{\Lambda-1}) y_{\Lambda-j-1}^{\Lambda-2},
\end{aligned}$$

(with an application of the chain rule in line 1).

In general, to compute the output derivative with respect to weight $w_{i,i+j}$, we apply the chain rule repeatedly up to the derivative of the diagonal node $(i+j, i+j)$ through which arises the output's only dependence on $w_{i,i+j}$. Thus,

$$
\begin{aligned}
\frac{\partial \tilde{u}}{\partial w_{i,i+j}} &= \phi(y^{\Lambda}) w_{\Lambda-1,\Lambda} \frac{\partial y^{\Lambda-1}}{\partial w_{i,i+j}} \\
&= \phi(y^{\Lambda}) w_{\Lambda-1,\Lambda} \frac{\partial f(\sum_{l=1}^{\Lambda-2} w_{l,\Lambda-1} y_l^{\Lambda-2} + y_{k-1})}{\partial w_{i,i+j}} \\
&= \phi(y^{\Lambda}) w_{\Lambda-1,\Lambda} \phi(y^{\Lambda-1}) w_{\Lambda-2,\Lambda-1} \frac{\partial y^{\Lambda-2}}{\partial w_{i,i+j}} \\
&\vdots \\
&= \phi(y^{\Lambda}) w_{\Lambda-1,\Lambda} \phi(y^{\Lambda-1}) w_{\Lambda-2,\Lambda-1} \cdots \phi(y^{i+j+1}) w_{i+j,i+j+1} \frac{\partial y^{i+j}}{\partial w_{i,i+j}}.
\end{aligned}
$$

But from (2.5.5),

$$
\frac{\partial y^{i+j}}{\partial w_{i,i+j}} = \phi(y^{i+j}) y_i^{i+j-1}
$$

and this establishes the lemma.

Appendix C

Appendix To Chapter 3

C.1 Proof of Theorem 3.2.2

The MAP criterion for generating q $(1 \leq q \leq p)$ decisions for a block length p, under decision feedback is

$$\tilde{U}_k \triangleq \arg\max_{U \in \mathbb{B}^q} Pr(U \triangleq [u_k,\ u_{k+1},\ \ldots,\ u_{k+q-1}]' \mid Z_k,\ E_k = 0).$$

We can rewrite the above probability using Bayes' rule as

$$\begin{aligned} Pr(U|Z_k, E_k = 0) &= Pr(Z_k|\ E_k = 0,\ U)\frac{Pr(U|\ E_k = 0)}{Pr(Z_k|\ E_k = 0)} \qquad \text{(C.1.1)} \\ &= \sum_{V \in \mathbb{B}^{p-q}} Pr(Z_k|\ E_k = 0, U_k = \begin{bmatrix} U \\ V \end{bmatrix}) Pr(V|E_k = 0), \end{aligned}$$

where

$$V \triangleq [u_{k+q},\ u_{k+q+1},\ \ldots,\ u_{k+p-1}]'.$$

Now $Pr(Z_k|E_k = 0, U_k) = Pr(N_k = Z_k - DU_k|E_k = 0, U_k)$. The noise sequence is independent of the input data so U_k and N_k are independent. At this stage of the design we have not specified how the past decisions (state estimates in the ARMA case) are generated. We now assume that E_k (which depends on these past decisions or state estimates) is independent of U_k and N_k. (Observe that in the resulting detector, E_k will depend on u_j and n_j for $j \leq k + p - q - 1$ through decision feedback, so there will be a (weak) correlation between E_k and N_k if $p > q$,

and between E_k and V if $p > 2q$.) Hence, dropping the conditioning from the right hand side of the last line of (C.1.1), we have

$$\tilde{U}_k \triangleq \arg\max_{U \in \mathbb{B}^q} \sum_{V \in \mathbb{B}^{p-q}} p_N\Big(Z_k - D\begin{bmatrix} U \\ V \end{bmatrix}\Big) Pr(V), \tag{C.1.2}$$

where $p_N(\cdot)$ is the multivariate Gaussian density defined in (3.2.17). Since the inputs are equiprobable and independent, $Pr(V) = 2^{q-p}$. This constant multiplier (along with the normalising constant of the density) does not affect the maximisation, so we omit it, thus obtaining (3.2.20).

C.2 Proof of Theorem 3.2.3

We can extract the dominant term from

$$\sum_{V \in \mathbb{B}^{p-q}} e^{-\frac{1}{2\sigma^2}\left\| Z_k - D\left[\begin{smallmatrix} U \\ V \end{smallmatrix}\right]\right\|^2}$$

as:

$$\begin{aligned} & e^{-\frac{1}{2\sigma^2}\left\| Z_k - D\left[\begin{smallmatrix} U \\ V^*(U) \end{smallmatrix}\right]\right\|^2} \Big(|\Omega(U)| + \sum_{V \in \Omega^c(U)} e^{-\frac{1}{2\sigma^2}\Delta^2(U,V)} \Big) \\ = \; & O\Big(e^{-\frac{1}{2\sigma^2}\left\| Z_k - D\left[\begin{smallmatrix} U \\ V^*(U) \end{smallmatrix}\right]\right\|^2} \Big) \text{ as } \sigma \to 0, \end{aligned} \tag{C.2.1}$$

where $\Omega(U)$ is the set

$$\Omega(U) \triangleq \Big\{ V \in \mathbb{B}^{p-q} \text{ realizing } \min \Big\| Z_k - D\begin{bmatrix} U \\ V \end{bmatrix} \Big\|^2 \Big\},$$

with cardinality $|\Omega(U)|$, and

$$\Delta^2(U,V) \begin{cases} = 0 & \forall V \in \Omega(U) \\ > 0 & \forall V \in \mathbb{B}^{p-q} \setminus \Omega(U) \triangleq \Omega^c(U) \end{cases}$$

and $V^*(U) \in \Omega(U)$. We now need the following Lemma which relates the optimal problem over $U \in \mathbb{B}^q$ to an optimization over $W \in \mathbb{B}^p$ via projection.

Lemma C.2.1 *Let* $\Omega \triangleq \{W \in I\!B^p \text{ realizing } \min \|Z_k - DW\|^2\}$. *If* $V^*(U) \in \Omega(U)$ *then*

$$\arg\min_{U \in I\!B^q} \left\| Z_k - D \begin{bmatrix} U \\ V^*(U) \end{bmatrix} \right\|^2 = [I_q \vdots 0\ ]W^*$$

for some

$$W^* \triangleq \begin{bmatrix} U^* \\ V^* \end{bmatrix} \in \Omega, \quad U^* \in I\!B^q, \text{ and } V^*(U^*) = V^*.$$

Proof: Suppose (to establish a contradiction)

$$\hat{U} \triangleq \arg\min_{U \in I\!B^q} \left\{ \left\| Z_k - D \begin{bmatrix} U \\ V^*(U) \end{bmatrix} \right\|^2 \right\} \neq [I_q \vdots 0\ ]W^*$$

for any $W^* \in \Omega$. Then (by definition)

$$\left\| Z_k - D \begin{bmatrix} \hat{U} \\ V^*(\hat{U}) \end{bmatrix} \right\|^2 \leq \left\| Z_k - D \begin{bmatrix} U \\ V^*(U) \end{bmatrix} \right\|^2, \quad \forall\, U \in I\!B^q.$$

and by definition of $V^*(U) \in \Omega(U)$, for each $U \in I\!B^q$

$$\left\| Z_k - D \begin{bmatrix} U \\ V^*(U) \end{bmatrix} \right\|^2 \leq \left\| Z_k - D \begin{bmatrix} U \\ V \end{bmatrix} \right\|^2, \quad \forall\, V \in I\!B^{p-q}.$$

But this implies $\begin{bmatrix} \hat{U} \\ V^*(\hat{U}) \end{bmatrix} \in \Omega$ which is a contradiction.

□

Using (C.1.1) and (C.2.1), now observe that (3.2.20) is asymptotically equivalent to

$$\arg\min_{U \in I\!B^q} \left\{ \left\| Z_k - D \begin{bmatrix} U \\ V^*(U) \end{bmatrix} \right\|^2 \right\} \text{ as } \sigma \to 0.$$

However Lemma C.2.1 shows that the above can be rewritten as (3.2.23) which is the desired result.

C.3 Proof of Lemma 3.5.1

Proof: From the definitions of $Y_k, \hat{\mathcal{H}}$ and $\mathcal{U}_k$, we have

$$\begin{aligned} Y_k' \hat{\mathcal{H}} \mathcal{U}_k &= y_k(\hat{h}_L u_{k-L} + \hat{h}_{L-1} u_{k-L+1} + \cdots + \hat{h}_0 u_k) \\ &+ y_{k+1}(\hat{h}_L u_{k-L+1} + \cdots + \hat{h}_0 u_{k+1}) + \end{aligned}$$

$$\vdots \qquad \text{(C.3.1)}$$

$$+ \quad y_{k+p-1}(\hat{h}_L u_{k-L+p-1} + \cdots + \hat{h}_0 u_{k+p-1}), \qquad \text{(C.3.2)}$$

with

$$\phi_{ij} \triangleq \mathcal{E}\left\{y_{k+i-1}\, u_{k-L+j-1}\right\},\ i = 1, \cdots, p,\ j = 1, \cdots, L+p.$$

Take the expectation of (C.3.1) to get

$$\begin{aligned} \mathcal{E}\left\{Y_k' \hat{\mathcal{H}} \mathcal{U}_k\right\} &= \hat{h}_L(\phi_{11} + \phi_{22} + \cdots + \phi_{pp}) \\ &+ \hat{h}_{L-1}(\phi_{12} + \phi_{23} + \cdots + \phi_{p,p+1}) + \\ &\vdots \\ &+ \hat{h}_0(\phi_{1,L+1} + \phi_{2,L+2} + \cdots + \phi_{p,L+p}), \end{aligned} \qquad \text{(C.3.3)}$$

whence (3.5.22) follows.

Let $\hat{h}_{ij}$ (not to be confused with $\hat{h}_i$ in (3.5.20)) be the ij element of the $(L+p) \times (L+p)$ matrix $\hat{\mathcal{H}}'\hat{\mathcal{H}}$, then

$$\begin{aligned} \mathcal{E}\left\{\mathcal{U}_k' \hat{\mathcal{H}}' \hat{\mathcal{H}} \mathcal{U}_k\right\} &= \mathcal{E}\left\{\sum_{i=1}^{L+p}\sum_{j=1}^{L+p} \hat{h}_{ij}\, u_{k-L+i-1}\, u_{k-L+j-1}\right\} \\ &= \sum_{i=1}^{L+p}\sum_{j=1}^{L+p} \hat{h}_{ij} \mathcal{E}\left\{u_{k_L+i-1}\, u_{k-L+j-1}\right\}. \end{aligned} \qquad \text{(C.3.4)}$$

By assumption, $\mathcal{E}\{u_i u_j\} = \delta_{ij}$, where δ_{ij} is the Kronecker delta, so that (3.5.23) is equal to

$$\sum_{i=1}^{L+p}\sum_{j=1}^{L+p} \hat{h}_{ij}\delta_{ij} = \sum_{i=1}^{L+p} \hat{h}_{ii} = \text{tr}\ \hat{\mathcal{H}}'\hat{\mathcal{H}}. \qquad \text{(C.3.5)}$$

Consideration of the form of $\hat{\mathcal{H}}$ leads by observation to

$$\begin{aligned} \text{tr}\ \hat{\mathcal{H}}'\hat{\mathcal{H}} &= \hat{h}_L^2 + (\hat{h}_{L-1}^2 + \hat{h}_L^2) + (\hat{h}_{L-2}^2 + \hat{h}_{L-1}^2 + \hat{h}_L^2) \\ &+ \cdots + (\hat{h}_{L-p+2}^2 + \cdots + \hat{h}_L^2) \\ &+ \{(\hat{h}_{L-p+1}^2 + \cdots + \hat{h}_L^2) + (\hat{h}_{L-p}^2 + \cdots + \hat{h}_{L-1}^2) \\ &+ \cdots + (\hat{h}_0^2 + \cdots + \hat{h}_{p-1}^2)\} \\ &+ (\hat{h}_0^2 + \cdots + \hat{h}_{p-2}^2) + \cdots + (\hat{h}_0^2 + \hat{h}_1^2) + \hat{h}_0^2, \end{aligned}$$

where each of the $L - p + 2$ terms in the curly brackets is a sum of p terms. It is easy to check that each $\widehat{h}_i^2$ occurs exactly p times in the above sum, so that (3.5.23) follows.

□

C.4 Proof of Theorem 3.5.1

Proof: Starting with

$$\|\mathcal{V}_k\|^2 = Y_k'Y_k - 2Y_k'\widehat{\mathcal{H}}\mathcal{U}_k + \mathcal{U}_k'\widehat{\mathcal{H}}'\widehat{\mathcal{H}}\mathcal{U}_k,$$

we have firstly,

$$\frac{\partial}{\partial \widehat{h}_j} Y_k'Y_k = 0 \ \forall j.$$

We use (C.3.1) to re-express $Y_k'\widehat{\mathcal{H}}\mathcal{U}_k$ as

$$\sum_{l=0}^{L} \widehat{h}_{L-l} \sum_{i=1}^{p} y_{k+i-1}u_{k-L+l-1},$$

whence

$$\begin{aligned} \frac{\partial}{\partial \widehat{h}_j} Y_k'\widehat{\mathcal{H}}\mathcal{U}_k &= \sum_{l=0}^{L} \delta_{L-l,j} \sum_{i=1}^{p} y_{k+i-1}u_{k-L+l-1} \\ &= \sum_{i=1}^{p} y_{k+i-1}u_{k+i-j-1}. \end{aligned}$$

Furthermore, we have

$$\widehat{\mathcal{H}}\mathcal{U}_k = \begin{bmatrix} \widehat{h}_L u_{k-L} + \widehat{h}_{L-1}u_{k-L+1} + \cdots + \widehat{h}_0 u_k \\ \widehat{h}_L u_{k-L+1} + \widehat{h}_{L-1}u_{k-L+2} + \cdots + \widehat{h}_0 u_{k+1} \\ \vdots \\ \widehat{h}_L u_{k-L+p-1} + \widehat{h}_{L-1}u_{k-L+p} + \cdots + \widehat{h}_0 u_{k+p-1} \end{bmatrix}, \qquad \text{(C.4.1)}$$

which has squared norm

$$\sum_{l=0}^{p-1} \left(\sum_{i=0}^{L} \widehat{h}_{L-i}u_{k-L+i+l} \right)^2,$$

and derivatives

$$\frac{\partial}{\partial \widehat{h}_j}\|\widehat{\mathcal{H}}\mathcal{U}_k\|^2 = 2\sum_{l=0}^{p-1} u_{k+l-j}\left(\sum_{i=0}^{L}\widehat{h}_{L-i}u_{k-L+i+l}\right) \quad j = 1,\ldots,L.$$

Combining the above derivatives, we obtain, after a relabelling of indices in the sums

$$\begin{aligned}\frac{1}{2}\frac{\partial}{\partial \widehat{h}_j}\|\mathcal{V}_k\|^2 &= \sum_{l=0}^{p-1} u_{k+l-j}\sum_{i=0}^{L}\widehat{h}_i u_{k+l-i} \\ &- \sum_{l=0}^{p-1} u_{k+l-j}y_{k+l}.\end{aligned}$$

Evaluation at $\widehat{h}_j = \widehat{h}_j(k)$ and substitution in (3.5.27) now yields the result.

□

Appendix D

Appendix To Chapter 4

D.1 Primary Error Probability Calculation

We are considering the primary error probability of the high signal-to-noise ratio (2,1)-DFE on a first order channel with $h_0 = 1$ and $h_1 > 0$ (the $h_1 < 0$ case follows the same lines). We start by defining the sets

$$\begin{aligned}
\mathcal{H}_1 &\triangleq \{(z_1, z_2) \in \mathbb{R}^2 \mid h_1(z_2 - 1) + z_1 < 0, z_2 \geq h_1\} \\
\mathcal{H}_2 &\triangleq \{(z_1, z_2) \in \mathbb{R}^2 \mid (h_1 - 1)z_2 + z_1 < 0, z_2 < h_1\} \\
\mathcal{H}_3 &\triangleq \{(z_1, z_2) \in \mathbb{R}^2 \mid h_1(z_2 + 1) + z_1 < 0, z_2 \leq -h_1\} \\
\mathcal{H}_- &= \mathcal{H}_1 \cup \mathcal{H}_2 \cup \mathcal{H}_3 \\
\mathcal{H}_+ &= \{Z \in \mathbb{R}^2 \mid -Z \in \mathcal{H}_-\}.
\end{aligned} \tag{D.1.1}$$

Note that the sets $\mathcal{H}_i$ $(i = 1, 2, 3)$ are pairwise disjoint. Following (4.3.11) and (4.3.13) and taking advantage of symmetry, we integrate over $\mathcal{H}_-$ to obtain the exact expression for the primary error probability

$$\begin{aligned}
\mathcal{P}_0 &= \frac{1}{2}\int_1^\infty p_n(t) Q\left(\frac{1 + h_1(t + h_1 - 2)}{\sigma}\right) dt \\
&+ \frac{1}{2}\int_{2h_1 - 1}^\infty p_n(t) Q\left(\frac{1 + h_1(h_1 - t)}{\sigma}\right) dt \\
&+ \frac{1}{2}\int_{-1}^{2h_1 - 1} p_n(t) Q\left(\frac{1 + (h_1 - 1)(h_1 - 1 - t)}{\sigma}\right) dt \\
&+ \frac{1}{2}\int_{-1}^\infty p_n(t) Q\left(\frac{1 + h_1(t + h_1)}{\sigma}\right) dt
\end{aligned}$$

$$+ \quad \frac{1}{2}\int_{2h_1+1}^{\infty} p_n(t)Q(\frac{1+h_1(h_1+2-t)}{\sigma})dt$$
$$+ \quad \frac{1}{2}\int_{1}^{2h_1+1} p_n(t)Q(\frac{1+(h_1-1)(h_1+1-t)}{\sigma})dt,$$

where $p_n(\cdot)$ is a Gaussian zero-mean density with variance σ^2, and $Q(\cdot)$ is defined in (4.3.6). The weighting of each term, $2 \times \frac{1}{4} = \frac{1}{2}$, arises from the equal probability of the u_k in (4.3.6). The reader may want to verify that this expression reduces to (4.3.15) when $h_1 = 1$.

D.2 Proof of Equations (4.3.16)-(4.3.18)

We are considering the integral of a two-dimensional circular Gaussian density, centred at the origin, over the following regions in the (x, y)-plane:

$$\begin{aligned}
\mathcal{R}_1 &= \{(x,y) \in I\!R^2 \mid 1 < y < \infty, x > y\} \\
\mathcal{R}_2 &= \{(x,y) \in I\!R^2 \mid -\infty < y < \infty, x > y+2\} \\
\mathcal{R}_3 &= \{(x,y) \in I\!R^2 \mid -\infty < y < -3, x > y+4\}
\end{aligned}$$

We rely on the separability of a white two-dimensional Gaussian into two one-dimensional Gaussians (equation 4.3.9). We express the integrals over $\mathcal{R}_1$ and $\mathcal{R}_2$ as standard integrals (in cases (i) and (ii) below), and bound the integral over $\mathcal{R}_3$ (in case (iii)) using integrals over quadrants or half-spaces.

Case (i)

Let $\mathcal{R}_1' = \{(x,y) \mid y > x, 1 < x < \infty\}$, then $\mathcal{R}_1 \cap \mathcal{R}_1' = \emptyset$ and $\mathcal{R}_1 \cup \mathcal{R}_1' = (1,\infty) \times (1,\infty) \cup \emptyset$, where $\emptyset$ is a set of Lebesgue measure zero. Note that the integral of p_N over $(1,\infty)$ is simply $Q(\frac{1}{\sigma})$. By symmetry of the density $p_N(\cdot)$, $\int_{\mathcal{R}_1} dp_N = \int_{\mathcal{R}_1'} dp_N$, and therefore

$$\int_{\mathcal{R}_1 \cup \mathcal{R}_1'} dp_N = Q(\frac{1}{\sigma}).Q(\frac{1}{\sigma}) = 2\int_{\mathcal{R}_1} dp_N,$$

which yields (4.3.16).

Case (ii)

This is straightforward, as $\mathcal{R}_2$ is a half-space:

$$\int_{\mathcal{R}_2} dp_N = \frac{1}{2}Q(\frac{\sqrt{2}}{\sigma}),$$

which is (4.3.17).

Case (iii)

Define the following sets in $\mathbb{R}^2$:

$$\begin{aligned} \widetilde{\mathcal{R}}_3 &= (1,\infty) \times (-\infty,-3) \\ \mathcal{H}_1 &= \{(x,y) \mid x > y+4\}. \end{aligned}$$

Then, since $\widetilde{\mathcal{R}}_3 \supset \mathcal{R}_3 \supset \mathcal{H}_1$, we have

$$\int_{\widetilde{\mathcal{R}}_3} dp_N < \int_{\mathcal{R}_3} dp_N < \int_{\mathcal{H}_1} dp_N,$$

but

$$\int_{\widetilde{\mathcal{R}}_3} dp_N = Q(\frac{1}{\sigma})Q(\frac{3}{\sigma}),$$

and

$$\int_{\mathcal{H}_1} dp_N = Q(\frac{3}{\sigma}).$$

This yields the bounds in (4.3.18).

D.3 Reachability of the Zero-Error State

We start with the following definition.

Definition D.3.1 (Reachability) *A state E_K in the finite-state Markov process for the block DFE is* reachable *from an initial state E_k, $k < K$, if there exists a finite sequence of inputs $\{u_{k+i} \mid i = 0,\ldots,K-1\}$ (each having nonzero probability) driving the system from E_k to E_K. A state that is reachable from any initial state in the FSMP is said simply to be* reachable.

The zero-error state of the FSMP for a given channel must necessarily be reachable if the block DFE is to be able to recover from an arbitrary initial error condition. We show here that, for a high SNR (2,1)-DFE operating on an order L finite impulse response channel, the zero-error state is always reachable.

Reachability of the zero-error state is assured if we can write down a rule or recipe for choosing the terms in the sequence $\{u_{k+i} \mid i = 0, \ldots, K-1\}$ that drive the state from $E_k \neq \underline{0}$ to $E_K = \underline{0}$. Since the error state is an L-vector in this case and satisfies the shift register property (4.5.1), we need only determine a length L sequence of inputs $\{u_{k+i} \mid i = 0, \ldots, L-1\}$, each resulting in a correct decision $\hat{u}_{k+i} = u_{k+i}$.

The noiseless input to the decision device (with $h_0 = 1$) is

$$Z_k = DU_k + R_k$$

where $U_k = [u_k, u_{k+1}]'$, D is given by (4.2.6) and

$$R_k = \begin{bmatrix} h_1 e_{k-1} + h_2 e_{k-2} + \cdots + h_L e_{k-L} \\ h_2 e_{k-1} + \cdots + h_L e_{k-L+1} \end{bmatrix}.$$

In fact, the geometry of the decision boundary is such that we can always choose a u_k, with u_{k+1} unspecified at time k, so that for any R_k (determined by E_k and h_i, $i \geq 1$) we have $\hat{u}_k = u_k$. The same recipe applies at subsequent times $k+1, k+2, \ldots, k+L-1$, yielding the sequence of inputs $\{u_{k+i} \mid i = 0, \ldots, L-1\}$ that drives the system from an arbitrary state $E_k \neq \underline{0}$ to $E_{k+L} = \underline{0}$. We now show this.

Recall that the decision $\hat{u}_k$ is given by (4.2.14). We first note the following property of the decision device.

Property D.3.1 *The decision device input DU_k always lies outside the strip S defined by*

$$S = \{Z_k \in \mathbb{R}^2 \mid |h_1 z_{k,2} + z_{k,1}| < |h_1|\}$$

for all $U_k \in \mathbb{B}^2$.

This is simple to check. The point $[1, h_1 + 1]' \notin S$ since

$$h_1(h_1 + 1) + 1 > h_1, \ \forall h_1.$$

The point $[1, h_1 - 1]' \notin S$ since

$$h_1(h_1 - 1) + 1 = h_1^2 - h_1 + 1 > h_1, \text{ when } |h_1| \neq 1.$$

(We consider the $h_1 = 1$ case separately.) By symmetry, the same reasoning applies to $[-1, h_1 \pm 1]'$.

We now define the two half-spaces

$$\mathcal{H}_1 = \{(z_1, z_2) \in I\!R^2 \mid h_1 z_2 + z_1 \geq 0\}$$

$$\mathcal{H}_2 = \{(z_1, z_2) \in I\!R^2 \mid h_1 z_2 + z_1 < 0\}$$

with $\mathcal{H}_1 \cup \mathcal{H}_2 = I\!R^2$. Treating the term R_k as an arbitrary (quantized) displacement of the point DU_k, we note that if we choose u_k according to the recipe

$$u_k = 1 \quad \text{if} \quad R_k \in \mathcal{H}_1 \tag{D.3.1}$$

$$u_k = -1 \quad \text{if} \quad R_k \in \mathcal{H}_2, \tag{D.3.2}$$

then the decision device input will lie outside S, *i.e.*,

$$D \begin{bmatrix} u_k \\ \pm 1 \end{bmatrix} + R_k \notin S.$$

Consequently, with $R_k \triangleq [r_{k,1}, r_{k,2}]' \in \mathcal{H}_1$, the top line of (4.2.14) implies

$$\begin{aligned} \widehat{u}_k &= \operatorname{sgn}(h_1 z_{k,2} + z_{k,1}) \\ &= \operatorname{sgn}\{(h_1^2 + 1)u_k + h_1 u_{k+1} + \underbrace{h_1 r_{k,2} + r_{k,1}}_{\geq 0} + u_k\} \\ &= u_k \text{ regardless of } u_{k+1}, \end{aligned}$$

since $h_1^2 + 1 > |h_1| \; \forall h_1 \in I\!R$. A similar argument shows that we can choose $\widehat{u}_k = u_k$ when $r_k \in \mathcal{H}_2$.

If $h_1 = \pm 1$, we simply redefine the sets $\mathcal{H}_1$ and $\mathcal{H}_2$ as

$$\mathcal{H}_1 = \{(z_1, z_2) \in I\!R^2 \mid sz_2 + z_1 > 0\} \cup \{(z_1, z_2) \mid sz_2 + z_1 = 0, z_1 \geq 0\}$$

$$\mathcal{H}_1 = \{(z_1, z_2) \in I\!R^2 \mid sz_2 + z_1 < 0\} \cup \{(z_1, z_2) \mid sz_2 + z_1 = 0, z_1 < 0\},$$

where $s \triangleq \text{sgn}(h_1)$, then the same recipe for u_k (D.3.1) still applies.

We remark that the preceding argument would not extend so easily to the $(3,1)$-DFE as Property D.3.1 generally does not hold.

D.4 Inclusion of Noise into the FSMP Analysis

We illustrate briefly how to incorporate the effects of noise into a finite-state Markov process-based analysis, like that of section 4.5, by considering for simplicity the an ordinary tuned DFE operating on a (noisy) first order channel. The analysis automatically includes the effects of error propagation. There are more possible transitions when noise is present, for instance the zero-error state is no longer an absorbing state. The technique also extends to the two-input block DFE, although the calculation of the elements of the transition matrix p_{ij} would be considerably more elaborate (see section 4.3).

The output of the decision feedback equalizer is then

$$\hat{u}_k = \text{sgn}(u_k + h_1 e_{k-1} + n_k),$$

where there is only one element of the error state, namely $e_{k-1} = u_{k-1} - \hat{u}_{k-1}$. Consider the probability of being in the zero-error state at time $k+1$, using Bayes' rule, this is

$$Pr(e_{k+1} = 0) = \sum_{e_k \in I\!E} Pr(e_k = 0 \mid e_k) Pr(e_k). \tag{D.4.1}$$

Now, the transition probabilities of the FSMP are given by

$$p_{ij} = Pr(\langle e_{k+1} \rangle = i \mid \langle e_k \rangle = j),\ 1 \le i, j \le 3 \tag{D.4.2}$$

where we number the states according to (4.5.5) as

$$\langle e_k = -2 \rangle = 1,\ \langle e_k = 0 \rangle = 2,\ \langle e_k = 2 \rangle = 3.$$

We define the steady state transition matrix as $P_s = \lim_{k \to \infty} P^k$ with elements $\overline{p}_{ij}$.

In the steady state (for large k) we would have $Pr(e_{k+1} = 0) = Pr(e_k = 0) \triangleq \lambda$, so that (D.4.1) implies

$$\lambda = \overline{p}_{21}(1-\lambda)/2 + \overline{p}_{22}\lambda + \overline{p}_{23}(1-\lambda)/2,$$

or

$$\lambda = \frac{\frac{1}{2}(\overline{p}_{21} + \overline{p}_{23})}{1 - \overline{p}_{22} + \frac{1}{2}(\overline{p}_{21} + \overline{p}_{23})},$$

in which we have used the equiprobability of the input symbols and the odd symmetry of the decision device, *i.e.*,

$$Pr(e_{k-1} = 2) = Pr(e_{k-1} = -2) = (1 - Pr(e_{k-1} = 0))/2.$$

The probability of error (BER) is therefore

$$\text{BER} = 1 - \lambda = \frac{1 - \overline{p}_{22}}{1 + \frac{1}{2}(\overline{p}_{21} + \overline{p}_{23}) - \overline{p}_{22}}. \tag{D.4.3}$$

It remains to evaluate the elements of the transition probability matrix p_{ij}. We use Bayes' rule again to write for given e_k and e_{k-1}

$$\begin{aligned} Pr(e_k \mid e_{k-1}) &= \frac{1}{2} Pr(\hat{u}_k = 1 - e_k \mid e_{k-1}, u_k = 1) \\ &+ \frac{1}{2} Pr(\hat{u}_k = -1 - e_k \mid e_{k-1}, u_k = -1). \end{aligned} \tag{D.4.4}$$

Assuming white Gaussian noise with variance σ^2, we can show from the above that the elements of P, given by (D.4.2), are

$$Pr(e_k \mid e_{k-1}) = \begin{cases} \frac{1}{2}Q(\frac{1-h_1 e_{k-1}}{\sigma}), & e_k = -2 \\ \frac{1}{2}Q(\frac{-1-h_1 e_{k-1}}{\sigma}) + \frac{1}{2}Q(\frac{-1+h_1 e_{k-1}}{\sigma}), & e_k = 0 \\ \frac{1}{2}Q(\frac{1+h_1 e_{k-1}}{\sigma}), & e_k = 2. \end{cases} \tag{D.4.5}$$

For a particular choice of channel $[h_0, h_1]$ and signal-to-noise ratio, we first compute the transition probability matrix P numerically from (D.4.5), obtain the steady state matrix by raising P to a large power and finally compute the bit error rate using (D.4.3). For instance, with $h_0 = h_1 = 1$ and a SNR of 4dB, we obtain a theoretical

BER of 0.0966 with this method, compared with a simulated BER of 0.0970 (for a 10^6 point run).

Appendix E

Appendix To Chapter 5

E.1 Structure of the Error State FSMP

In effect, we are computing the steady state distribution of the finite-state Markov process governing the dynamics of the detector's error states. Denote by π_k the stochastic L-vector whose elements are the probabilities of the respective error states (numbered from 1 to 3^L) at time k, and by P_L the $3^L \times 3^L$-matrix whose (i,j) element p_{ij} is the probability of of a transition from state j to state i, or in symbols

$$p_{ij} = Pr(\langle E_{k+1} \rangle = i \mid \langle E_k \rangle = j).$$

The governing equation is then $\pi_{k+1} = P_L \, \pi_k$ and the (unique) steady state distribution π_s satisfies $\pi_s = P_L \, \pi_s$. Although the actual entries of P_L (transition probabilities) are dependent on the channel parameters and on the SNR, the *structure* of P_L is constrained by symmetry with respect to sign (property 5.3.1) and the following fact:

From each state at time k, only 3 transitions to states at time k+1 may have nonzero probability.

This property derives from the (shift register) definition of the error state (5.3.12). In particular, the following holds for any FIR channel:

$$Pr(E_{k+1} = \underline{0} \mid E_k = \underline{0}) = 1 - 2Pr(E_{k+1} = [0, 0, \cdots, 0, 2]' \mid E_k = \underline{0}). \qquad \text{(E.1.1)}$$

For instance, with the indexing rule for error states that assigns an integer $\langle E_k \rangle \in \{1, \cdots, 3^L\}$ to error state E_k (5.3.12) according to (see section 4.5.1):

$$\langle E_k \rangle = (1 + \frac{1}{2}e_{k-1})3^{L-1} + (1 + \frac{1}{2}e_{k-2})3^{L-2} + \cdots + (1 + \frac{1}{2}e_{k-L})3^0, \qquad \text{(E.1.2)}$$

the transition probability matrix for the detector operating on a second order channel ($L = 2$) would have the structure:

$$P_2 = \begin{bmatrix} p_{11} & p_{12} & p_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{24} & p_{25} & p_{26} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{73} & p_{72} & p_{71} \\ p_{41} & p_{42} & p_{43} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{54} & p_{55} & p_{54} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{43} & p_{42} & p_{41} \\ p_{71} & p_{72} & p_{73} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{26} & p_{25} & p_{24} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_{13} & p_{12} & p_{11} \end{bmatrix}. \qquad \text{(E.1.3)}$$

Of course, all column sums of P_L are unity.

If we use the state indexing rule given in (E.1.2), we can deduce the structure of P_{L+1} from P_L simply by replacing each nonzero element of P_L in turn by a 3×3 block having one of three possible forms (the particular form depends on the position of the element being replaced), and each zero element by a 3×3 matrix of zeros. The three block types are (with appropriate choice of subscripts)

$$\begin{bmatrix} p_{..} & p_{..} & p_{..} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ p_{..} & p_{..} & p_{..} \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ p_{..} & p_{..} & p_{..} \end{bmatrix}.$$

There is clearly a high degree of structural similarity between transition matrices for successive channel orders.