University of Southampton
Faculty of Engineering
Department of Electronics and Computer Science
Southampton S017 1BJ

Radial Basis Function Network Based Multilevel Channel Equalisation Techniques for Wireless Communications

by
Mong Suan Yee
B. Eng.

A doctoral thesis submitted in partial fulfilment of
requirements for the award of Doctor of Philosophy
at the University of Southampton

June 2000

SUPERVISOR: *Professor Lajos Hanzo*
Dipl Ing, MSc, PhD, SMIEEE
Chair of Telecommunications
Department of Electronics and Computer Science
University of Southampton
Southampton SO17 1BJ
United Kingdom

University of Southampton

Abstract

Faculty of Engineering

Department of Electronics and Computer Science

Doctor of Philosophy

Radial Basis Function Network Based Multilevel Channel Equalisation Techniques for Wireless Communications

by Mong Suan Yee

This thesis investigates the employment of Radial Basis Function (RBF) networks in the context of multilevel channel equalisation. The RBF-based Decision Feedback Equaliser (DFE) was found to outperform the conventional DFE at the cost of an increased computational complexity. The RBF DFE was studied in the context of a wideband Burst-by-Burst (BbB) Adaptive Quadrature Amplitude Modulation (AQAM) scheme, where the modulation modes of no transmission (NO TX), Binary Phase Shift Keying (BPSK), 4-QAM, 16-QAM and 64-QAM were invoked by the transmitter, depending on the prevalent channel quality. The 'short-term BER' of the received burst quantifies the channel quality, which was used as the modem mode switching criterion in order to switch between different modulation modes. The Bit Per Symbol (BPS) throughput improvement for the proposed AQAM scheme designed for a target Bit Error Rate (BER) of 1% was up to a factor of two in comparison to the fixed constituent modulation modes.

The logarithmic version of the RBF equaliser referred to as the Jacobian RBF equaliser was derived, which has a reduced computational complexity. Turbo codes were invoked for improving the BER and BPS performance of the BbB AQAM scheme. The Jacobian RBF equaliser provides a logarithmic-domain output, which can be used to provide soft outputs for the channel decoder. We proposed employing the average magnitude of the Log-Likelihood Ratio (LLR) of the bits in the received burst as the channel quality measure for our adaptive scheme. The system exhibited a better BPS performance, when compared with the uncoded AQAM/RBF DFE system at low to medium channel SNRs and also showed an improved coded BER performance at higher channel SNRs.

The recently developed family of iterative equalisation and channel decoding techniques termed as turbo equalisation were explored and we investigated the employment of RBF-based turbo equalisers. A reduced-complexity RBF DFE turbo equaliser (TEQ) was proposed where the RBF DFE skips the evaluation of the symbol LLRs in the current iteration when the symbol becomes sufficiently reliable after channel decoding in the previous iteration. The proposed scheme provided an equivalent BER performance to the RBF DFE TEQ with a complexity reduction of approximately 21% at an SNR of 6dB.

Acknowledgements

I would like to express my warmest thanks to Professor Lajos Hanzo, for his encouragement, guidance and patience throughout this long walk, while completing this thesis. He has been my source of inspiration and I have greatly benefited from his advice, not only concerning my research, but also on life as a whole.

Many thanks to my collegues and friends, for their continual support, help and numerous contributions. In particular, I would like to thank Jon Blogh, Spyros Vlahoyiannatos, Yang Lie-Liang, Peter Cherriman, Choi Byoung Jo, How Hee Thong, Liew Tong Hooi alias Telly Tubby, Thomas Keller, Jason Woodard, Yen Kai, Yeap Bee Leong a.k.a the Rabbit, Kuan Ee Lin, Moustachio Wong Choong Hin and Michael Ng for lively and sometimes humourous discussions. Special thanks are also due to Denise Harvey for her help with the administrative matters.

The financial support of the European Union under the auspices of the Pan-European FIRST project and that of Motorola ECID, Swindon UK is also gratefully acknowledged.

Finally, special thanks to my family back in Malaysia for their love and encouragement and also to my family in Christ for being there during my difficult times.

List of Publications

1. Multi-level Radial Basis Function Network Based Equalisers for Rayleigh Channel, M. S. Yee, L. Hanzo, Proceedings of IEEE Vehicular Technology Conference, Houston, Texas, USA, May 16-19, 1999, pp.707-711.

2. Upper Bound Performance of Radial Basis Function Decision Feedback Equalised Burst-by-burst Adaptive Modulation, M. S. Yee, L. Hanzo, ECMCS'99, Krakow, Poland, 24-26th June 1999, CD-ROM.

3. Block Turbo Coded Burst-By-Burst Adaptive Radial Basis Function Decision Feedback Equaliser Assisted Modems, M. S. Yee, T. H. Liew, L. Hanzo, Proceedings of IEEE Vehicular Technology Conference, Amsterdam, Netherlands, pp.1600-1604, September 1999.

4. Radial Basis Function Decision Feedback Equaliser Assisted Burst-by-burst Adaptive Modulation , M. S. Yee, L. Hanzo, Proceedings of IEEE Globecom'99, Rio de Janeiro, Brazil, pp.2183-2187, 5-9 December 1999.

5. Radial Basis Function Assisted Turbo Equalisation, M. S. Yee, B. L. Yeap, L. Hanzo, to be published in the Proceedings of IEEE Vehicular Technology Conference, Tokyo, Japan, pp.640-644, 15-18 May 2000.

6. A Wideband Radial Basis Function Decision Feedback Equaliser Assisted Burst-by-burst Adaptive Modem, M. S. Yee and L. Hanzo, submitted to IEEE Transactions on Communications.

7. Burst-by-burst Adaptive Turbo Coded Radial Basis Function Assisted Decision Feedback Equalisation, M. S. Yee, T. H. Liew and L. Hanzo, submitted to IEEE Transactions on Communications.

8. Radial Basis Function Assisted Turbo Equalisation, M. S. Yee, B. L. Yeap and L. Hanzo, submitted to IEEE Transactions on Communications.

Contents

Chapter 1

Introduction

The popularity and penetration of mobile cellular communication systems and wireless networking technologies is growing at an ever-faster rate and this is certainly set to continue in the foreseeable future. Second-generation mobile radio systems, which use digital technology in contrast to the analogue first-generation systems, are proving successful worldwide in providing communications services to users. The customer base is increasing faster than expected. These second-generation systems are dominated by the Pan-European Global System of Mobile Communnications (GSM) [1], and the Pan-American IS-136 [2], IS-95 [2] schemes while in Japan by the Personal Digital Cellular (PDC) and Personal Handyphone System (PHS) technology. However, these second-generation systems provide only voice and low-data-rate services. The predicted market requirements and service needs were translated to technical requirements for the definition of third-generation (3G) mobile radio systems [3, 4] in North America, Asia and Europe. The 3G mobile technology will not only deliver a wide variety of wireless services – speech, video and data at various bit rates but it also will herald a new era of services that combine high-speed mobile access with mobile multimedia and Internet Protocol (IP) based services. The International Mobile Telecommunications system in the year 2000 [4] (IMT-2000) – formerly known as Future Public Land Mobile Telecommunications Systems (FPLMTS) – is the term used by the International Telecommunications Union (ITU) as the specification for 3G services, based upon a 'family' of compatible standards, of which a GSM-based evolution is set to be the most widespread. The 3G mobile systems have to satisfy the following requirements [5]:

- Support a wide range of bearer services from voice and low-rate to high-rate data services with at least 144 kbit/s in vehicular, 384 kbit/s in outdoor-to-indoor and up to 2 Mbit/s in indoor and picocell environments.
- Capacity and coverage improvement over that of the second generation systems, while achieving a higher quality transmission.

- Circuit-switched and packet-switched oriented operation.

Over recent years a significant amount of efforts has been devoted, both in industry and academia, towards the definition of the IMT-2000 third-generation wireless communication systems, standards, technologies and applications. Several regional standard organizations - led by the European Telecommunications Standards Institute (ETSI) in Europe, by the Association of Radio Industries and Businesses (ARIB) in Japan and by the Telecommunications Industry Association (TIA) in the United States have been dedicating their efforts to specifying the standards for IMT-2000. A total of 15 Radio Transmission Technology (RTT) IMT-2000 proposals were submitted to ITU-R in June 1998, five of which are satellite based solutions, while the rest are terrestrial solutions. Although the standard has now reached maturity, there is still a large set of important research and development problems to be addressed and resolved. At the time of writing research is already under way towards identifying the fundamental challenges and issues in the field of the fourth generation broadband wireless communication systems, which are expected to provide further improvements in terms of services, data rate, capacity, flexibility, power consumption and cost.

The new generation of mobile communication systems have to develop novel multiple-access techniques, along with the associated coding, equalisation and compression technologies, in order to meet the requirements for a reliable and secure transfer of large volumes of information at speeds commensurate with those of the fixed broadband networks. The overall objective is an improved efficiency of information transfer per unit bandwidth, higher protection against interference, efficient frequency re-use and channel allocation techniques, along with a concomitant decrease in implementational cost.

1.1 Research Motivation

In order to cope with the expected traffic and service quality demands of mobile communications, further research into transceiver design is required. Our research will focus on the equalisation aspects of dispersive fading mobile channels using neural network techniques [6].

Artificial Neural Networks (ANN) draw their inspiration from the structure of the human brain and from its functions. An ANN does not attempt to model faithfully the neurobiology of the human brain, but rather it employs the abstract notions of how the brain function. One of the key characteristics of the brain is its ability to learn and adapt appropriately to changing circumstances. Thus, for ANNs to learn from their environment, they must adopt a training or learning algorithm. Here we embark on making use of different ANN structures and their learning algorithms, which are applied to the wireless communications field, employing neural network based channel equalisation. Our objective

is to use new approaches from an ANN perpective, in order to improve the performance of conventional channel equalisers.

Due to the scarcity and price of radio spectrum available for mobile services, spectral efficiency is of primary concern for future mobile communication systems. The third generation system must accommodate a wide range of services and channel types, from pico- to macro-cellular, supporting data rates up to 2 Mbits/s. In this respect, the challenge is to provide a high transmission throughput, given the limited radio spectrum as well as to maintain a certain target transmission quality. The system may adapt to a range of factors related to both the channel and service quality, depending on whether speech, data or multimedia services are concerned. Adaptation may apply to the modulation mode and to the Forward Error Correction (FEC) coding scheme employed, among a range of other potential system features. For example, we can adapt the modulation modes based on the prevalent channel conditions, where a higher-order modulation mode is utilized, when the channel quality is favourable. By contrast, when the channel quality is low, the transmission integrity can be maintained by invoking a more robust modulation mode. In this treatise, we will explore the implementation and performance of the neural network based equalisers in a similar adaptive modulation mode based scheme.

In conjunction with equalisation, channel decoding can be employed in order to further improve the performance of the communications system. Powerful error correction schemes – such as turbo codes [7] – have been shown to yield performances close to Shannon's performance limits. Instead of performing the equalisation and decoding separately, higher performance gains can be achieved by implementing the equalisation and decoding operations jointly and iteratively. This technique is also known as turbo equalisation and has been shown to combat the effects of the channel-induced intersymbol interference successfully. Motivated by these trends, in this treatise we set out to amalgamate these powerful performance enhancement techniques with neural network based equalisation.

1.2 Organisation of Thesis

Below, we present the layout of the thesis:

- In Chapter 2, we cast channel equalisation as a classification problem. We give a brief overview of neural networks and present the design of some neural network based equalisers. In this thesis, we choose to examine a neural network structure referred to as the Radial Basis Function (RBF) network in more detail for channel equalisation, since it has an equivalent structure to the so-called optimal Bayesian equalisation solution [8]. The structure and properties of the RBF network are described, followed by the implementation of a RBF network as an equaliser. We will discuss the

computational complexity issues of the RBF equaliser with respect to that of conventional linear equalisers and provide some complexity reduction methods. Finally, performance comparisons between the RBF equaliser and the conventional equaliser are given over various channel scenarios.

- Chapter 3 commences by summarising the concept of adaptive modulation that adapts the modem mode according to the channel quality in order to maintain a certain target bit error rate and an improved bits per symbol throughput performance. The RBF based equaliser is introduced in a wideband Adaptive Quadrature Amplitude Modulation (AQAM) scheme in order to mitigate the effects of the dispersive multipath fading channel. We introduce the short-term Bit Error Rate (BER) as the channel quality measure. Lastly, a comparative study is conducted between the constituent fixed mode, the conventional DFE based AQAM scheme and the RBF based AQAM scheme in terms of their BER and throughput performance.

- In Chapter 4 we incorporate turbo channel coding in the proposed wideband AQAM scheme. A novel reduced-complexity RBF equaliser utilizing the so-called Jacobian logarithmic relationship [9] is proposed and the turbo-coded performance of the Jacobian RBF equaliser is presented for the various fixed QAM modes. Furthermore, we investigate using various channel quality measures – namely the short-term BER and the average Log-Likelihood Ratio (LLR) magnitude of the data burst generated either by the RBF equaliser or the turbo decoder – in order to control the modem mode-switching regime for our adaptive scheme.

- Chapter 5 introduces the principles of iterative, joint equalisation and decoding techniques known as turbo equalisation [10]. We present a novel turbo equalisation scheme, which employs a RBF equaliser instead of the conventional trellis-based equaliser. The structure and computational complexity of both the RBF equaliser and trellis-based equaliser are compared and we characterise the performance of these RBF and trellis-based turbo-equalisers. We then propose a reduced-complexity RBF assisted turbo equaliser, which exploits the fact that the RBF equaliser computes its output on a symbol-by-symbol basis and the symbols of the decoded transmission burst, which are sufficiently reliable, need not be equalised in the next turbo equalisation iteration.

- Chapter 6 summarises our main findings and conclusions. Suggestions for future work are also presented.

The novel contributions of the thesis are as follows:

- A wideband AQAM scheme was implemented with the aid of a RBF based equaliser. The modem mode switching metric was based on the short-term BER of the equalised

burst as estimated by the RBF equaliser. The short-term BER switching thresholds were obtained by estimating the BER improvement/degradation upon switching to a lower/higher order modulation mode [11, 12].

- Based on the approach often used in turbo codes, we proposed generating the output of the RBF network in logarithmic form by invoking the so-called Jacobian logarithm [13, 14], in order to reduce the computational complexity of the RBF equaliser. The proposed RBF equaliser using the Jacobian logarithm was referred to as the Jacobian logarithmic RBF equaliser [9].

- Turbo coding was incorporated into the wideband AQAM scheme, where Burst-by-Burst (BbB) based decoding was achieved. The average burst LLR amplitude was proposed as the channel quality measure. The performance of the RBF DFE assisted BbB AQAM in conjunction with turbo coding was assessed using different modem mode switching criteria – namely the short-term BER and the average burst LLR magnitude before and after turbo channel decoding [9].

- We presented a novel turbo equalisation scheme, which employs a RBF equaliser instead of the conventional trellis-based equaliser. The proposed turbo equaliser was shown to achieve identical BER performance to the conventional turbo equaliser, while incurring a lower complexity [15]. The computational complexity of the RBF turbo equaliser was further reduced by refraining from feeding back those decoded symbols which were deemed sufficiently reliable to the equaliser in the next iteration for further iterative equalisation.

Having presented an overview of the thesis, let us now commence our discussions with a brief overview of neural network based equalisation.

Chapter 2

Neural Network Based Equalisation

In this chapter, we will give an overview of neural network based equalisation. Channel equalisation can be viewed as a classification problem. The optimal solution to this classification problem is inherently nonlinear. Hence we will discuss, how the nonlinear structure of the artificial neural network can enhance the performance of conventional channel equalisers and examine various neural network designs amenable to channel equalisation, such as the so-called multilayer perceptron network [16, 17, 18, 19, 20], polynomial perceptron network [21, 22, 23, 24] and radial basis function network [25, 26, 8, 27]. We will examine a neural network structure referred to as the Radial Basis Function (RBF) network in detail in the context of equalisation. As further reading, the contribution by Mulgrew [28] provides an insightful briefing on applying RBF network for both channel equalisation and interference rejection problems. Originally RBF networks were developed for the generic problem of data interpolation in a multi-dimensional space [29, 30]. We will describe the RBF network in general and motivate its application. Before we proceed, our forthcoming section will describe the discrete time channel model inflicting intersymbol interference that will be used throughout this thesis.

2.1 Discrete Time Model for Channels Exhibiting Intersymbol Interference

A band-limited channel that results in intersymbol interference (ISI) can be represented by a discrete-time transversal filter having a transfer function of:

$$F(z) = \sum_{n=0}^{L} f_n z^{-n}, \tag{2.1}$$

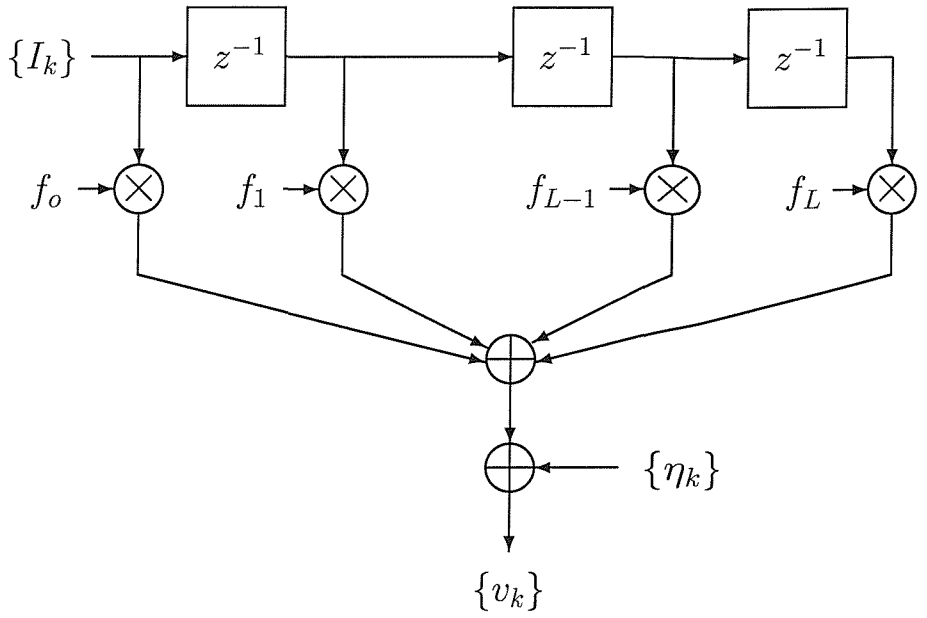


Figure 2.1: Equivalent discrete-time model of a channel exhibiting intersymbol interference and experiencing additive white Gaussian noise

where f_n is the nth impulse response tap of the channel and $L+1$ is the length of the channel impulse response (CIR). In this context, the channel represents the convolution of the impulse responses of the transmitter filter, the transmission medium and the receiver filter. In our discrete-time model discrete symbols I_k are transmitted to the receiver at a rate of $\frac{1}{T}$ symbols per second and the output v_k at the receiver is also sampled at a rate of $\frac{1}{T}$ per second. Consequently, as depicted in Figure 2.1, the passage of the input sequence $\{I_k\}$ through the channel results in the channel output sequence $\{v_k\}$ that can be expressed as

$$v_k = \sum_{n=0}^{L} f_n I_{k-n} + \eta_k \qquad -\infty \leq k \leq \infty, \tag{2.2}$$

where $\{\eta_k\}$ is a white Gaussian noise sequence with zero mean and variance σ_η^2. The number of interfering symbols contributing to the ISI is L. In general, the sequences $\{v_k\}$, $\{I_k\}$, $\{\eta_k\}$ and $\{f_n\}$ are complex-valued. Again, Figure 2.1 illustrates the model of the equivalent discrete-time system corrupted by Additive White Gaussian Noise (AWGN).

2.2 Equalisation as a Classification Problem

In this section we will show that the characteristics of the transmitted sequence can be exploited by capitalising on the finite state nature of the channel and by considering the equalisation problem as a geometric classification problem. This approach was first expounded by Gibson, Siu and Cowan [17], who investigated utilizing nonlinear structures

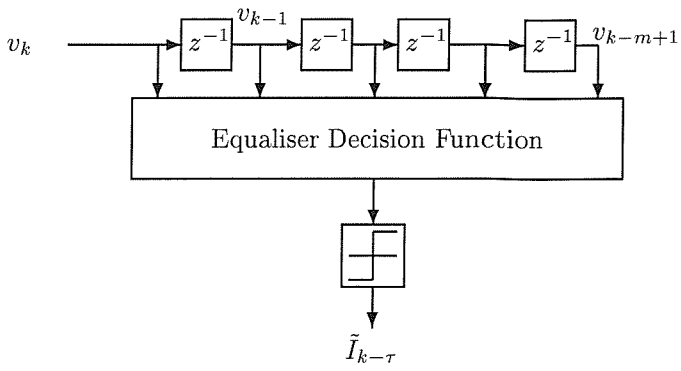


Figure 2.2: Linear m-tap equaliser schematic

offered by Neural Networks (NN) as channel equalisers.

We assume that the transmitted sequence is binary with equal probability of logical ones and zeros in order to simplify the analysis. Referring to Equation 2.2 and using the notation of Section 2.1, the symbol-spaced channel output is defined by

$$\begin{aligned} v_k &= \sum_{n=0}^{L} f_n I_{k-n} + \eta_k \\ &= \tilde{v}_k + \eta_k \qquad -\infty \leq k \leq \infty, \end{aligned} \tag{2.3}$$

where $\{\eta_k\}$ is the additive Gaussian noise sequence, $\{f_n\}$, $n = 0, 1, \ldots, L$ is the CIR, $\{I_k\}$ is the channel input sequence and $\{\tilde{v}_k\}$ is the noise-free channel output.

The mth order equaliser, as illustrated in Figure 2.2, has m taps as well as a delay of τ, and it produces an estimate $\tilde{I}_{k-\tau}$ of the transmitted signal $I_{k-\tau}$. The delay τ is due to the precursor section of the CIR, since it is necessary to facilitate the causal operation of the equaliser by supplying the past and future received samples, when generating the delayed detected symbol $I_{k-\tau}$. Hence the required length of the decision delay is typically the length of the CIR's precursor section, since outside this interval the CIR is zero and therefore the equaliser does not have to take into account any other received symbols. The channel output observed by the linear mth order equaliser can be written in vectorial form as

$$\mathbf{v}_k = \begin{bmatrix} v_k & v_{k-1} & \ldots & v_{k-m+1} \end{bmatrix}^T, \tag{2.4}$$

and hence we can say that the equaliser has an m-dimensional channel output observation space. For a CIR of length $L+1$, there are hence $n_s = 2^{L+m}$ possible combinations of the binary channel input sequence

$$\mathbf{I}_k = \begin{bmatrix} I_k & I_{k-1} & \ldots & I_{k-m-L+1} \end{bmatrix}^T \tag{2.5}$$

that produce $n_s = 2^{L+m}$ different possible noise-free channel output vectors

$$\tilde{\mathbf{v}}_k = \begin{bmatrix} \tilde{v}_k & \tilde{v}_{k-1} & \ldots & \tilde{v}_{k-m+1} \end{bmatrix}^T. \tag{2.6}$$

The possible noise-free channel output vectors $\tilde{v}_k$ or particular points in the observation space will be referred to as the desired channel states. Expounding further, we denote each of the $n_s = 2^{L+m}$ possible combinations of the channel input sequence $\mathbf{I}_k$ of length $L+m$ symbols as $\mathbf{s}_i, 1 \leq i \leq n_s = 2^{L+m}$, where the channel input state $\mathbf{s}_i$ determines the desired channel output state $\mathbf{r}_i$, $i = 1, 2, \ldots, n_s = 2^{L+m}$. This is formulated as:

$$\tilde{\mathbf{v}}_\mathbf{k} = \mathbf{r}_i \qquad \text{if } \mathbf{I}_k = \mathbf{s}_i, \qquad i = 1, 2, \ldots, n_s. \tag{2.7}$$

The desired channel output states can be partitioned into two classes according to the binary value of the transmitted symbol $I_{k-\tau}$, as seen below:

$$\begin{aligned} V_{m,\tau}^{+} &= \{\tilde{\mathbf{v}}_k | I_{k-\tau} = +1\}, \\ V_{m,\tau}^{-} &= \{\tilde{\mathbf{v}}_k | I_{k-\tau} = -1\}, \end{aligned} \tag{2.8}$$

and

$$V_{m,\tau} = V_{m,\tau}^{+} \bigcup V_{m,\tau}^{-}. \tag{2.9}$$

We can denote the desired channel output states according to these two classes as follows:

$$\begin{aligned} \mathbf{r}_i^{+} &\in V_{m,\tau}^{+} \qquad i = 1, 2, \ldots, n_s^{+}, \\ \mathbf{r}_j^{-} &\in V_{m,\tau}^{-} \qquad j = 1, 2, \ldots, n_s^{-}, \end{aligned} \tag{2.10}$$

where the quantities n_s^{+} and n_s^{-} represent the number of channel states $\mathbf{r}_i^{+}$ and $\mathbf{r}_j^{-}$ in the set $V_{m,\tau}^{+}$ and $V_{m,\tau}^{-}$, respectively.

The relationship between the transmitted symbol I_k and the channel output v_k can also be written in a compact form as:

$$\begin{aligned} \mathbf{v}_k &= \mathbf{F}\mathbf{I}_k + \boldsymbol{\eta}_k \\ &= \tilde{\mathbf{v}}_k + \boldsymbol{\eta}_k, \end{aligned} \tag{2.11}$$

where $\boldsymbol{\eta}_k$ is an m-component vector that represents the AWGN sequence, $\tilde{\mathbf{v}}_k$ is the noise-free channel output vector and $\mathbf{F}$ is an $m \times (m+L)$ CIR-related matrix in the form of:

$$\mathbf{F} = \begin{bmatrix} f_0 & f_1 & \cdots & f_L & \cdots & 0 \\ 0 & f_0 & \cdots & f_{L-1} & \cdots & 0 \\ \vdots & \vdots & & & & \vdots \\ 0 & 0 & f_0 & \cdots & f_{L-1} & f_L \end{bmatrix}, \tag{2.12}$$

with $f_j, j = 0, \ldots, L$ being the CIR taps.

Below we demonstrate the concept of finite channel states in a two-dimensional output observation space ($m = 2$) using a simple two-coefficient channel ($L = 1$), assumming the CIR of:

$$F(z) = 1 + 0.5z^{-1}. \tag{2.13}$$

Thus, $\mathbf{F} = \begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0.5 \end{bmatrix}$, $\tilde{\mathbf{v}}_k = \begin{bmatrix} \tilde{v}_k & \tilde{v}_{k-1} \end{bmatrix}^T$ and $\mathbf{I}_k = \begin{bmatrix} I_k & I_{k-1} & I_{k-2} \end{bmatrix}^T$. All the possible combinations of the transmitted binary symbol I_k and the noiseless channel outputs $\tilde{v}_k$, $\tilde{v}_{k-1}$, are listed in Table 2.1.

I_k	I_{k-1}	I_{k-2}	$\tilde{v}_k$	$\tilde{v}_{k-1}$
-1	-1	-1	-1.5	-1.5
-1	-1	+1	-1.5	-0.5
-1	+1	-1	-0.5	+0.5
-1	+1	+1	-0.5	+1.5
+1	-1	-1	+0.5	-1.5
+1	-1	+1	+0.5	-0.5
+1	+1	-1	+1.5	+0.5
+1	+1	+1	+1.5	+1.5

Table 2.1: Transmitted signal and noiseless channel states for the CIR of $F(z) = 1+0.5z^{-1}$ and an equaliser order of $m = 2$.

Figure 2.3 shows the 8 possible noiseless channel states $\tilde{\mathbf{v}}_k$ for a BPSK modem and the noisy channel output $\mathbf{v}_k$ in the presence of zero mean AWGN with variance $\sigma_\eta^2 = 0.05$. It is seen that the observation vector $\mathbf{v}_k$ forms clusters and the centroids of these clusters are the noiseless channel states $\mathbf{r}_i$. The equalisation problem hence involves identifying the regions within the observation space spanned by the noisy channel output $\mathbf{v}_k$ that correspond to the transmitted symbol of either $I_k = +1$ or $I_k = -1$.

A linear equaliser performs the classification in conjunction with a decision device, which is often a simple sign function. The decision boundary, as seen in Figure 2.3, is constituted by the locus of all values of $\mathbf{v}_k$, where the output of the linear equaliser is zero as it is demonstrated below. For example, for a two tap linear equaliser having tap coefficients c_1 and c_2, at the decision boundary we have:

$$v_k c_1 + v_{k-1} c_2 = 0 \tag{2.14}$$

and

$$v_{k-1} = -(\frac{c_1}{c_2}) v_k \tag{2.15}$$

gives a straight line decision boundary as shown in Figure 2.3, which divides the observation space into two regions corresponding to $I_k = +1$ and $I_k = -1$. In general, the linear equaliser can only implement a hyperplane decision boundary, which in our two-dimensional example was constituted by a line. This is clearly a non-optimum classification strategy, as our forthcoming geometric visualisation will highlight. For example, we can see in Figure 2.3 that the point $\tilde{\mathbf{v}} = \begin{bmatrix} 0.5 & -0.5 \end{bmatrix}$ associated with the $I_k = +1$ decision is closer to the

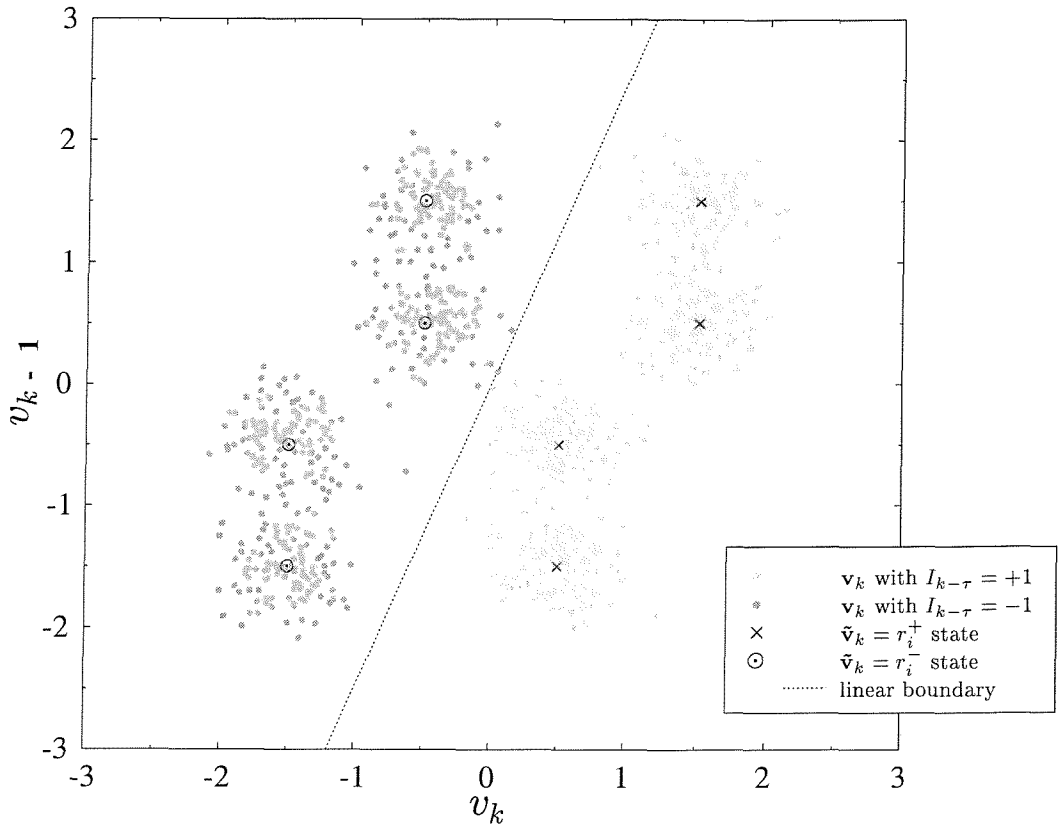


Figure 2.3: The noiseless BPSK-related channel states $\tilde{\mathbf{v}}_k = \mathbf{r}_i$ and the noisy channel outputs $\mathbf{v}_k$ of a Gaussian channel having a CIR of $F(z) = 1 + 0.5z^{-1}$ in a two-dimensional observation space. The noise variance $\sigma_\eta^2 = 0.05$, the number of noisy received $\mathbf{v}_k$ samples output by the channel and input to the equaliser is 2000 and the decision delay is $\tau = 0$. The linear decision boundary separates the noisy received $\mathbf{v}_k$ clusters that correspond to $I_{k-\tau} = +1$ from those that correspond to $I_{k-\tau} = -1$.

decision boundary than the point $\tilde{\mathbf{v}} = \begin{bmatrix} -1.5 & -0.5 \end{bmatrix}$ associated with the $I_k = -1$ decision. Therefore, in the presence of noise, there is a higher probability of the channel output centred at point $\tilde{\mathbf{v}} = \begin{bmatrix} 0.5 & -0.5 \end{bmatrix}$ to be wrongly detected as $I_k = -1$, than that of the channel output centred around $\tilde{\mathbf{v}} = \begin{bmatrix} -1.5 & -0.5 \end{bmatrix}$ being incorrectly detected as $I_k = +1$. Gibson *et. al.* [17] have shown examples of linearly non-separable channels, when the decision delay is zero and the channel is of non-minimum phase nature. The linear separability of the channel depends on the equaliser order, m, on the delay τ and in situations where the channel characteristics are time varying, it may not be possible to specify values of m and τ, which will guarantee linear separability.

According to Chen, Gibson and Cowan [21], the above shortcomings of the linear equaliser are circumvented by a Bayesian approach [31] to obtaining an optimal equalisation solution. In this spirit, for an observed channel output vector $\mathbf{v}_k$, if the probability that it was caused by $I_{k-\tau} = +1$ exceeds the probability that it was caused by $I_{k-\tau} = -1$, then we should decide in favour of $+1$ and vice versa. Thus, the optimal Bayesian equaliser solution is defined as [21]:

$$\tilde{I}_{k-\tau} = sgn(f_{Bayes}(\mathbf{v}_k)) = \begin{cases} +1 & \text{if } f_{Bayes}(\mathbf{v}_k) \geq 0 \\ -1 & \text{if } f_{Bayes}(\mathbf{v}_k) < 0, \end{cases} \tag{2.16}$$

where the optimal Bayesian decision function $f_{Bayes}(\cdot)$, based on the difference of the associated conditional density functions is given by [8]:

$$\begin{aligned} f_{Bayes}(\mathbf{v}_k) &= P(\mathbf{v}_k | I_{k-\tau} = +1) - P(\mathbf{v}_k | I_{k-\tau} = -1) \\ &= \sum_{i=1}^{n_s^+} p_i^+ p(\mathbf{v}_k - \mathbf{r}_i^+) - \sum_{j=1}^{n_s^-} p_j^- p(\mathbf{v}_k - \mathbf{r}_j^-), \end{aligned} \tag{2.17}$$

where p_i^+ and p_i^- is the *a priori* probability of appearance of each desired state $\mathbf{r}_i^+ \in V_{m,\tau}^+$ and $\mathbf{r}_i^- \in V_{m,\tau}^-$, respectively and $p(\cdot)$ denotes the associated probability density function. The quantities n_s^+ and n_s^- represent the number of desired channel states in $V_{m,\tau}^+$ and $V_{m,\tau}^-$, respectively, which are defined implicitly in Figure 2.3. If the noise distribution is Gaussian, Equation 2.17 can be rewritten as:

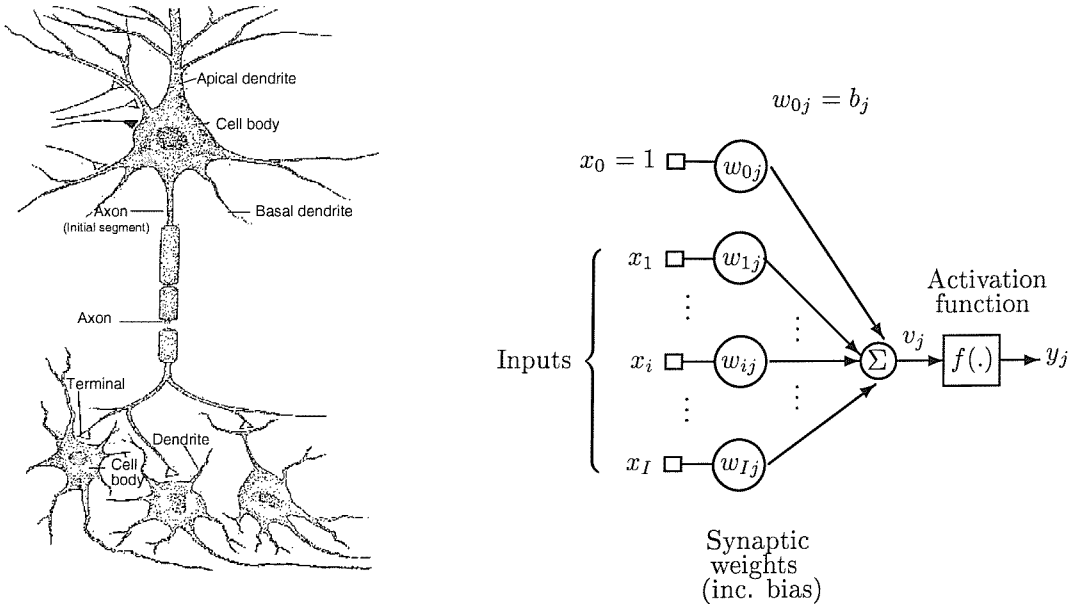
$$\begin{aligned} f_{Bayes}(\mathbf{v}_k) &= \sum_{i=1}^{n_s^+} p_i^+ (2\pi\sigma_\eta^2)^{-m/2} exp(-\|\mathbf{v}_k - \mathbf{r}_i^+\|^2 / 2\sigma_\eta^2) \\ &\quad - \sum_{j=1}^{n_s^-} p_j^- (2\pi\sigma_\eta^2)^{-m/2} exp(-\|\mathbf{v}_k - \mathbf{r}_j^-\|^2 / 2\sigma_\eta^2). \end{aligned} \tag{2.18}$$

Again, the optimal decision boundary is the locus of all values of $\mathbf{v}_k$, where the probability $I_{k-\tau} = +1$ given a value $\mathbf{v}_k$ is equal to the probability $I_{k-\tau} = -1$ for the same $\mathbf{v}_k$.

In general, the optimal Bayesian decision boundary is a hyper-surface, rather than just a hyper-plane in the m-dimensional observation space and the realisation of this nonlinear boundary requires a nonlinear decision capability. Neural networks provide this capability and the following section will discuss the various neural network structures that have been investigated in the context of channel equalisation, while also highlighting the learning algorithms used.

2.3 Introduction to Neural Networks

2.3.1 Biological and Artificial Neurons



(a) Anatomy of a typical biological neuron, from Kandel [32]

(b) An artificial neuron (jth-neuron)

Figure 2.4: Comparison between biological and artificial neurons

The human brain consists of a dense interconnection of simple computational elements referred to as neurons. Figure 2.4(a) shows a network of biological neurons. As seen in the figure, the neuron consists of a cell body – which provides the information-processing functions – and of the so-called axon with its terminal fibres. The dendrites seen in the figure are the neuron's 'inputs', receiving signals from other neurons. These input signals may cause the neuron to *fire*, i.e. to produce a rapid, short-term change in the potential difference across the cell's membrane. Input signals to the cell may be excitatory, increasing the chances of neuron firing, or inhibitory, decreasing these chances. The axon is the

neuron's transmission line that conducts the potential difference away from the cell body towards the terminal fibres. This process produces the so-called *synapses*, which form either excitatory or inhibitory connections to the dendrites of other neurons, thereby forming a neural network. Synapses mediate the interactions between neurons and enable the nervous system to adapt and react to its surrounding environment.

In Artificial Neural Networks (ANN), which mimic the operation of biological neural networks, the processing elements are artificial neurons and their signal processing properties are loosely based on those of biological neurons. Referring to Figure 2.4(b), the jth-neuron has a set of I synapses or connection links. Each link is characterized by a synaptic weight $w_{ij}, i = 1, 2, \ldots, I$. The weight w_{ij} is positive, if the associated synapse is excitatory and it is negative, if the synapse is inhibitory. Thus, signal x_i at the input of synapse i, connected to neuron j, is multiplied by the synaptic weight w_{ij}. These synaptic weights that store 'knowledge' and provide connectivity, are adapted during the learning process.

The weighted input signals of the neuron are summed up by an adder. If this summation exceeds a so-called firing threshold θ_j, then the neuron fires and issues an output. Otherwise it remains inactive. In Figure 2.4(b) the effect of the firing threshold θ_j is represented by a bias, arising from an input which is always 'on', corresponding to $x_0 = 1$, and weighted by $w_{0,j} = -\theta_j = b_j$. The importance of this is that the bias can be treated as just another weight. Hence, if we have a training algorithm for finding an approriate set of weights for a network of neurons, designed to perform a certain function, we do not need to consider the biases separately.

The activation function $f(\cdot)$ of Figure 2.5 limits the amplitude of the neuron's output to some permissible range and provides nonlinearities. Haykin [6] identifies three basic types of activation functions:

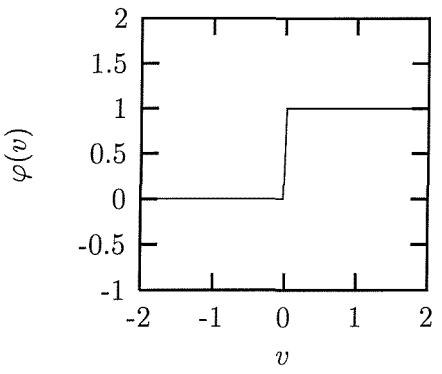
1. *Threshold Function.* For the threshold function shown in Figure 2.5(a), we have

$$f(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} . \tag{2.19}$$

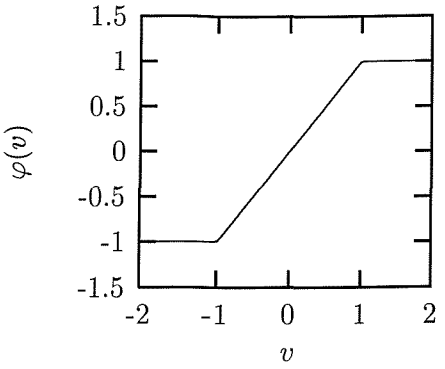
   Neurons using this activation function are referred to in the literature as the *McCulloch-Pitts model* [6]. In this model, the output of the neuron gives the value of 1 if the total internal activity level of that neuron is nonnegative and 0 otherwise.

2. *Piecewise-Linear Function.* This neural activation function, portrayed in Figure 2.5(b), is represented mathematically by:

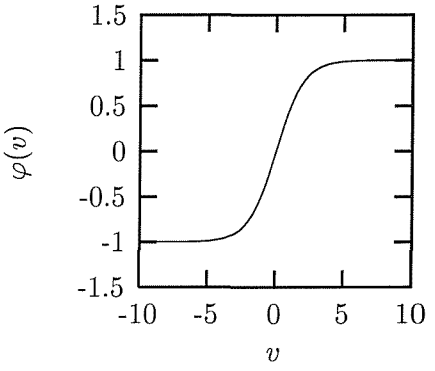
$$f(v) = \begin{cases} 1, & v \geq 1 \\ v, & -1 > v > 1 \\ -1, & v \leq -1 \end{cases} , \tag{2.20}$$



(a) Threshold activation function



(b) Piecewise-linear activation function



(c) Sigmoid activation function

Figure 2.5: Various neural activation functions $f(v)$

where the amplification factor inside the linear region is assumed to be unity. This activation function approximates a nonlinear amplifier.

3. *Sigmoid Function.* A commonly used neural activation function in the construction of artificial neural networks is the sigmoid activation function. It is defined as a strictly increasing function that exhibits smoothness and asymptotic properties, as seen in Figure 2.5(c). An example of the sigmoid function is the hyperbolic tangent function, which is shown in Figure 2.5(c) and it is defined by [6]:

$$f(v) = \frac{1 - exp(-v)}{1 + exp(-v)}. \tag{2.21}$$

This activation function is differentiable, which is an important feature in neural network theory [6].

The model of the jth artificial neuron, shown in Figure 2.4(b) can be described in mathematical terms by the following pair of equations:

$$y_j = f(v_j), \tag{2.22}$$

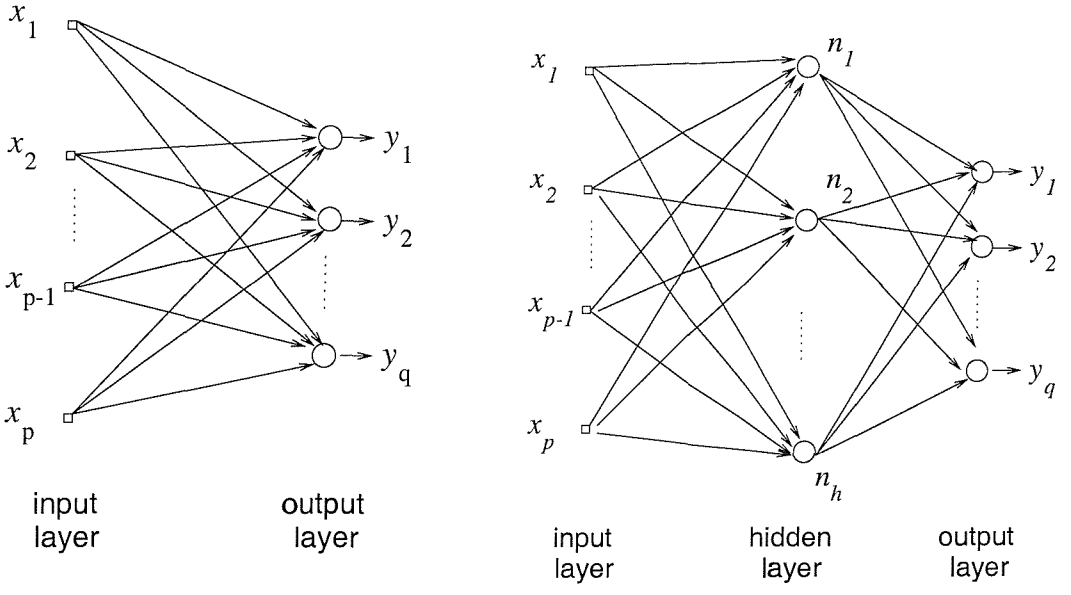
where:

$$v_j = \sum_{i=0}^{I} w_{ij} x_i. \tag{2.23}$$

Having introduced the basic elements of neural networks, we will focus next on the associated network structures or architectures. The different neural network structures yield different functionalities and capabilities. The basic structures will be described in the following section.

2.3.2 Neural Network Architectures

The network's architecture defines the neurons' arrangement in the network. Various neural network architectures have been investigated for different applications, including for example channel equalisation. Distinguishing the different structures can assist us in their design, analysis and implementation.We can identify three different classes of network architectures, which are the subjects of our forthcoming deliberations.



(a) Single-Layer Perceptron (SLP)

(b) Multi-Layer Perceptron (MLP)

Figure 2.6: Layered feedforward networks

The so-called *layered feedforward networks* of Figure 2.6 exhibit a layered structure, where all connection paths are directed from the input to the output, with no feedback. This

implies that these networks are unconditionally stable. Typically, the neurons in each layer of the network have only the output signals of the preceding layer as their inputs.

Two types of layered feedforward networks are often invoked, in order to introduce neural networks, namely the

- *Single-Layer Perceptrons* (SLP) which have a single layer of neurons.
- *Multi-Layer Perceptrons* (MLP) which have multiple layers of neurons.

Again, these structures are shown in Figure 2.6. The MLP distinguishes itself from the SLP by the presence of one or more *hidden layers* of neurons. Figure 2.6(b) illustrates the layout of a MLP having a single hidden layer. It is referred to as a p-h-q network, since it has p source nodes, h hidden neurons and q neurons in the output layer. Similarly, a layered feedforward network having p source nodes, h_1 neurons in the first hidden layer, h_2 neurons in the second hidden layer, h_3 neurons in the third layer and q neurons in the output layer is referred to as a p-h_1-h_2-h_3-q network. If the SLP has a differentiable activation function, such as the sigmoid function given in Equation 2.21, the network can learn by optimizing its weights using a variety of gradient-based optimization algorithms, such as the *gradient descent* method, described briefly in Appendix A. The interested reader can refer to the monograph by Bishop [33] for further gradient-based optimization algorithms used to train neural networks.

The addition of hidden layers of nonlinear nodes in MLP networks enables them to extract or learn nonlinear relationships or dependencies from the data, thus overcoming the restriction that SLP networks can only act as linear discriminators. Note that the capabilities of MLPs stem from the nonlinearities used within neurons. If the neurons of the MLP were linear elements, then a SLP network with appropriately chosen weights could carry out exactly the same calculations, as those performed by any MLP network. The downside of employing MLPs however, is that their complex connectivity renders them more implementationally complex and they need nonlinear training algorithms. The so-called *error back propagation* algorithm popularized in the contribution by Rumelhart *et. al.* [34, 35] is regarded as the standard algorithm for training MLP networks, against which other learning algorithms are often benchmarked [6].

Having considered the family of layered feedforward networks we note that a so-called *recurrent neural network* [6] distinguishes itself from a layered feedforward network by having at least one *feedback* loop.

Lastly, lattice structured neural networks [6] consist of networks of a one-dimensional, two-dimensional or higher-dimensional array of neurons. The lattice network can be viewed as a feedforward network with the output neurons arranged in rows and columns. For

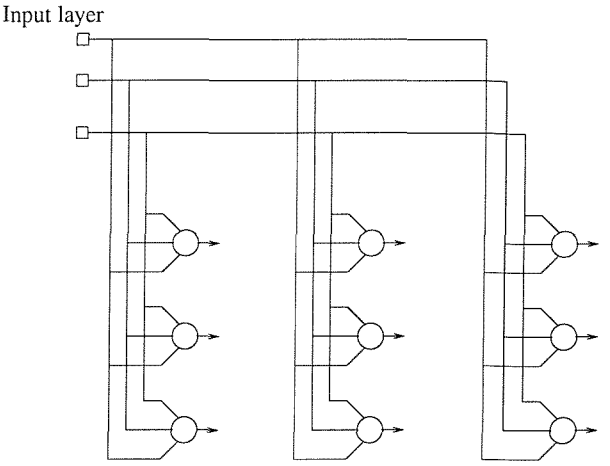


Figure 2.7: Two-dimensional lattice of 3-by-3 neurons

example, Figure 2.7 shows a two-dimensional lattice of 3-by-3 neurons fed from a layer of 3 source nodes.

Neural network models are specified by the nodes' characteristics, by the network topology, and by their training or learning rules, which set and adapt the network weights appropriately, in order to improve performance. Both the associated design procedures and training rules are the topic of much current research [36]. The above rudimentary notes only give a brief and basic introduction to neural network models. For a deeper introduction to other neural network topologies and learning algorithms, please refer for example to the review by Lippmann [37]. Let us now provide a rudimentary overview of the associated equalisation concepts in the following section.

2.4 Equalisation Using Neural Networks

A few of the neural network architectures that have been investigated in the context of channel equalisation are the so-called Multilayer Perceptron (MLP) advocated by Gibson, Siu and Cowan [16, 17, 18, 19, 20], as well as the Polynomial-Perceptron (PP) studied by Chen, Gibson, Cowan, Chang, Wei, Xiang, Bi, L.-Ngoc *et. al.* [21, 22, 23, 24]. Furthermore, the RBF was investigated by Chen, McLaughlin, Mulgrew, Gibson, Cowan, Grant *et. al.* [25, 26, 8, 27], the recurrent network [38] was proposed by C.-Sueiro, A.-Rodriguez and F.-Vidal, the Functional Link (FL) technique was introduced by Gan, Hussain, Soraghan and Durrani [39, 40, 41] and the Self-Organizing Map (SOM) was proposed by Kohonen *et. al.* [42].

Various neural network based equalisers have also been implemented and investigated for transmission over satellite mobile channels [43, 44, 45]. The following section will present and summarise some of the neural network based equalisers found in literature. We will

investigate the RBF structure in the context of equalisation in more detail during our later discourse in the next few sections.

2.5 Multilayer Perceptron Based Equaliser

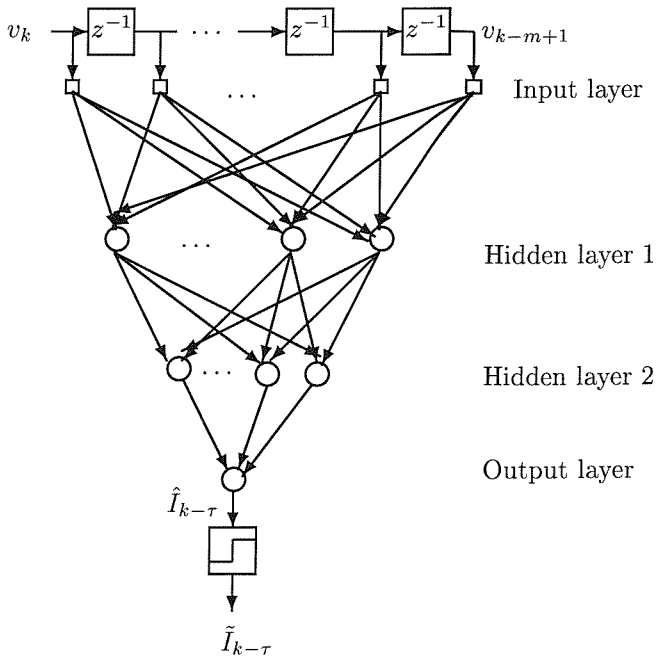


Figure 2.8: Multilayer perceptron model of the m-tap equaliser of Figure 2.2

Multilayer perceptrons (MLPs), which have three layers of neurons, i.e. two hidden layers and one output layer, are capable of forming any desired decision region for example in the context of modems, which was noted by Gibson and Cowan [46]. This property renders them attractive as nonlinear equalisers. The structure of a MLP network has been described in Section 2.3.2 as a layered feedforward network. As an equaliser, the input of the MLP network is the sequence of the received signal samples $\{v_k\}$ and the network has a single output, which gives the estimated transmitted symbol $\tilde{I}_{k-\tau}$, as shown in Figure 2.8. Figure 2.8 shows the $m - h_1 - h_2 - 1$ MLP network as an equaliser. Referring to Figure 2.9,

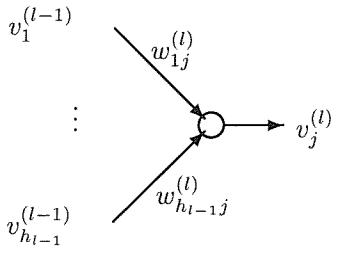


Figure 2.9: The jth neuron in the mth layer of the MLP

the jth neuron ($j = 1, \ldots, h_l$) in the lth layer ($l = 0, 1, 2, 3$, where the 0th layer is the input layer and the third layer is the output layer) accepts inputs $\mathbf{v}^{(l-1)} = [v_1^{(l-1)} \ldots v_{h_{l-1}}^{(l-1)}]^T$ from the $(l-1)$th layer and returns a scalar $v_j^{(l)}$ given by

$$v_j^{(l)} = f(\sum_{i=1}^{h_{l-1}} w_{ij}^{(l)} v_i^{(l-1)}) \qquad j = 1, \ldots, h_l, \qquad l = 0, 1, 2, 3, \tag{2.24}$$

where $h_0 = m$ is the number of nodes at the input layer, which is equivalent to the equaliser order and h_3 is the number of neurons at the output layer, which is one according to Figure 2.8. The output value $v_j^{(l)}$ serves as an input to the $(l+1)$th layer. Since the transmitted binary symbol taken from the set {+1,-1} has a bipolar nature, the sigmoid type activation function $f(\cdot)$ of Equation 2.21 is chosen to provide an output in the range of [-1,+1], as shown in Figure 2.5(c). The MLP equaliser can be trained adaptively by the so-called error back propagation algorithm described for example by Rumelhart, Hinton and Williams [34].

The major difficulty associated with the MLP is that training or determining the required weights is essentially a nonlinear optimisation problem. The mean squared error surface corresponding to the optimisation criterion is multi-modal, implying that the mean squared error surface has local minima as well as a global minimum. Hence it is extremely difficult to design gradient type algorithms, which guarantee finding the global error minimum corresponding to the optimum equaliser coefficients under all input signal conditions. The error back propagation algorithm to be introduced during our further discourse does not guarantee convergence, since the gradient descent might be trapped in a local minimum of the error surface. Furthermore, due to the MLP's typically complicated error surface, the MLP equaliser using the error back propagation algorithm has a slower convergence rate than the conventional adaptive equaliser using the Least Mean Square (LMS) algorithm described in Appendix A. This was illustrated for example by Siu *et al.* [20] using experimental results. The introduction of the so-called momentum term was suggested by Rumelhart *et al.* [35] for the adaptive algorithm to improve the convergence rate. The idea is based on sustaining the weight change moving in the same direction with a 'momentum' to assist the back propagation algorithm in moving out of a local minimum. Nevertheless, it is still possible that the adaptive algorithm may become trapped at local minima. Furthermore, the above-mentioned momentum term may cause oscillatory behaviour close to a local or global minimum. Interested readers may wish to refer to the excellent monograph by Haykin [6] that discusses the virtues and limitations of the error back propagation algorithm invoked to train the MLP network, highlighting also various methods for improving its performance. Another disadvantage of the MLP equaliser with respect to conventional equaliser schemes is that the MLP design incorporates a three-layer perceptron structure,

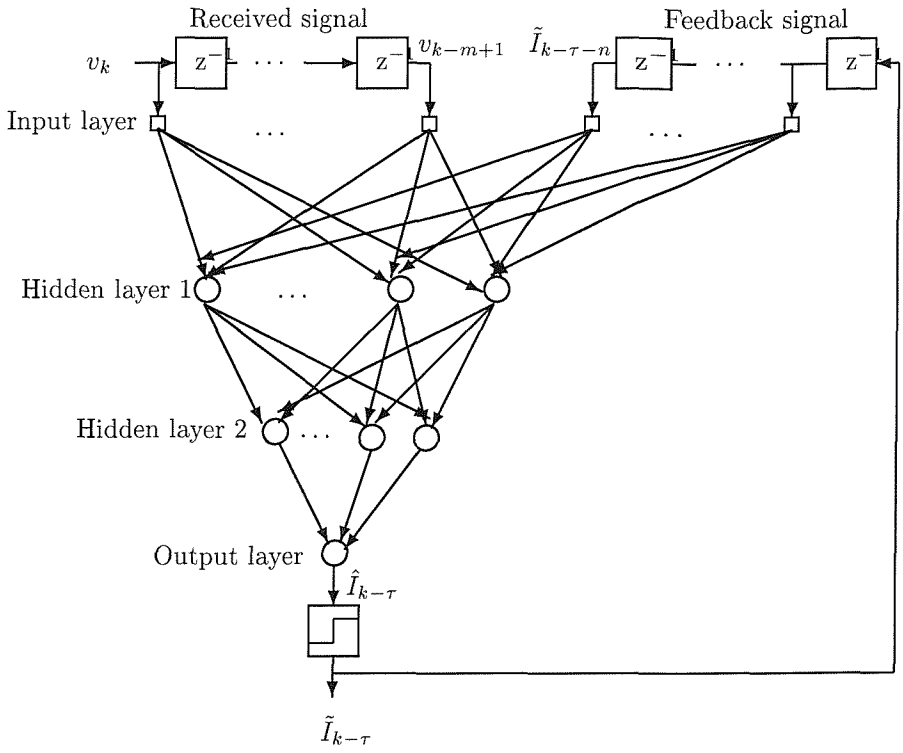


Figure 2.10: Multilayer perceptron equaliser with decision feedback

which is considerably more complex.

Siu *et al.* [20] incorporated decision feedback into the MLP structure, as shown in Figure 2.10 with a feedforward order of m and a feedback order of n. The authors provided simulation results for binary modulation over a dispersive Gaussian channel, having an impulse response of $F(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$. Their simulations show that the MLP DFE structure offers superior performance in comparison to the LMS DFE structure. They also provided a comparative study between the MLP equaliser with and without feedback. The performance of the MLP equaliser was improved by about 5dB at a BER of 10^{-4} relative to the MLP without decision feedback and having the same number of input nodes. Siu, Gibson and Cowan also demonstrated that the performance degradation due to decision errors is less dramatic for the MLP based DFE, when compared to the conventional LMS DFE, especially at poor signal-to-noise ratio (SNR) conditions. Their simulations showed that the MLP DFE structure is less sensitive to learning gain variation and it is capable of converging to a lower mean square error value. Despite providing considerable performance improvements, MLP equalisers are still problematic in terms of their convergence performance and due to their more complex structure relative to conventional equalisers.

2.6 Polynomial Perceptron Based Equaliser

The so-called PP or Volterra series structure was proposed for channel equalisation by Chen, Gibson and Cowan [21]. The PP equaliser has a simpler structure and a lower computational complexity, than the MLP structure, which makes it more attractive for equalisation. A perceptron structure is employed, combined with polynomial approximation techniques, in order to approximate the optimal nonlinear equalisation solution. The design is justified by the so-called *Stone-Weierstrass theorem* [47], which states that any continuous function can be approximated within an arbitrary accuracy by a polynomial of a sufficiently high order. The model of the PP was investigated in detail by Xiang *et al.* [24]. The nonlinear equaliser is constructed according to [21]:

$$\begin{aligned} f_p(\mathbf{v}_k) &= \sum_{i_1=0}^{m-1} c_{i_1} v_{k-i_1} + \sum_{i_1=0}^{m-1}\sum_{i_2=i_1}^{m-1} c_{i_1 i_2} v_{k-i_1} v_{k-i_2} + \cdots \\ &\quad + \sum_{i_1=0}^{m-1} \cdots \sum_{i_l=i_{l-1}}^{m-1} c_{i_1 \ldots i_l} v_{k-i_1} \cdots v_{k-i_l}, \\ &= \sum_{i=0}^{n} w_i x_{i,k}, \qquad (2.25) \\ f_{PP}(\mathbf{v}_k) &= f(f_p(\mathbf{v}_k)), \qquad (2.26) \\ \tilde{I}_{k-\tau} &= sgn[f_{PP}(\mathbf{v}_k)], \qquad (2.27) \end{aligned}$$

where l is the polynomial order, m is the equaliser order, $x_{i,k}$ are the so-called monomials (polynomial with a single power term) corresponding to the power terms of the equalizer inputs from v_{k-i_1} to $v_{k-i_1} \ldots v_{k-i_l}$, w_i are the corresponding polynomial coefficients c_{i_1} to $c_{i_1 \ldots i_l}$ and n is the number of terms in the polynomial. Here, the term w_i and $x_{i,k}$ of Equation 2.25 correspond to the synaptic weights and inputs of the perceptron/neuron described in Figure 2.4(b), respectively.

The function $f_p(\mathbf{v}_k)$ in Equation 2.26 is the polynomial that approximates the Bayesian decision function $f_{Bayes}(\mathbf{v}_k)$ of Equation 2.17 and the function $f_{PP}(\mathbf{v}_k)$ in Equation 2.26 is the PP decision function. The activation function of the perceptron $f(\cdot)$ is the sigmoid function given by Equation 2.21. The reasons for applying the sigmoidal function were highlighted by Chen, Gibson and Cowan [21], which are briefly highlighted below. In theory the number of terms in Equation 2.25 can be infinite. However, in practice only a finite number of terms can be implemented, which has to be sufficiently high to achieve a low received signal mis-classification probability, i.e. a low decision error probability. The introduction of the sigmoidal activation function $f(x)$ is necessary, since it allows a moderate polynomial degree to be used, while having an acceptable level of mis-classification of the equaliser input vector corresponding to the transmitted symbols. This was demonstrated

by Chen *et. al.* [21] using a simple classifier example. Chen *et. al.* [21] reported that a polynomial degree of $l = 3$ or 5 was sufficient with the introduction of the sigmoidal activation function judging from their simulation results for the experimental circumstances stipulated.

From a conceptual point of view, the PP structure expands the input space of the equaliser, which is defined by the dimensionality of $\{\mathbf{v}_k\}$, into an extended nonlinear space and then employs a neuron element in this space. Consider a simple polynomial perceptron

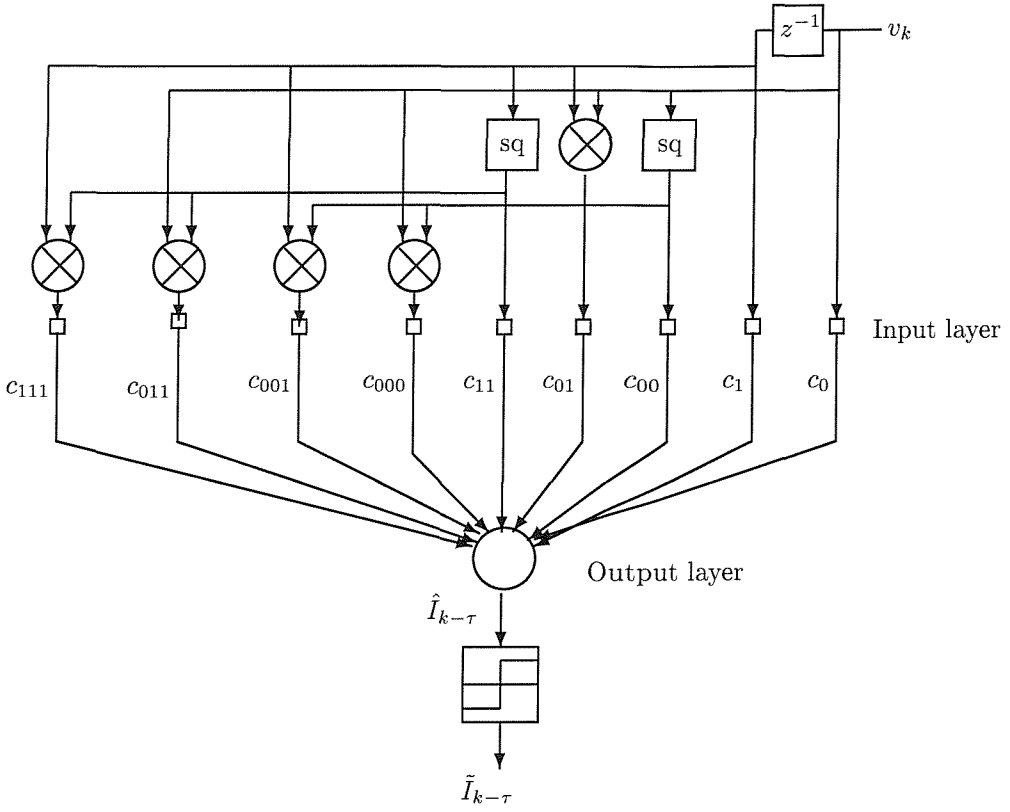


Figure 2.11: Polynomial perceptron equaliser using an equaliser order of $m = 2$ and polynomial order of $l = 3$

based equaliser, where the equaliser order is $m = 2$ and the polynomial order is $l = 3$. Then the polynomial decision function is given by:

$$\begin{aligned} f_{PP}(\mathbf{v}_k) &= f(c_0 v_k + c_1 v_{k-1} + c_{00} v_k^2 + c_{01} v_k v_{k-1} + c_{11} v_{k-1}^2 + \\ &\quad c_{000} v_k^3 + c_{001} v_k^2 v_{k-1} + c_{011} v_k v_{k-1}^2 + c_{111} v_{k-1}^3). \end{aligned} \tag{2.28}$$

The structure of the equaliser defined by Equation 2.28 is illustrated in Figure 2.11. The simulation results of Chen *et. al.* [21] using binary modulation show close agreement with the bit error rate performance of the MLP equaliser. However, the training of the PP equaliser is much easier compared to the MLP equaliser, since only a single-layer perceptron is involved in the PP equaliser. The nonlinearity of the sigmoidal activation function

introduces local minima to the error surface of the otherwise linear perceptron structure. Thus, the stochastic gradient algorithm [34, 35] assisted by the previously mentioned momentum term [35] can be invoked in their scheme in order to adaptively train the equaliser. The decision feedback structure of Figure 2.10 can be incorporated into Chen's design [21] in order to further improve the performance of the equaliser.

The PP equaliser is attractive, since it has a simpler structure than that of the MLP. The PP equaliser also has a multi-modal error surface – exhibiting a number of local minima and a global minimum – and thus still retains some problems associated with its convergence performance, although not as grave as the MLP structure. Another drawback is that the number of terms in the polynomial of Equation 2.25 increases exponentially with the polynomial order l and with the equaliser order m, resulting in an exponential increase of the associated computational complexity.

2.7 Radial Basis Function Networks

2.7.1 Introduction

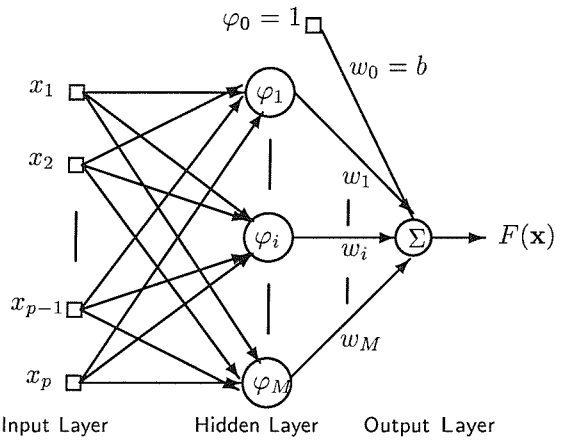


Figure 2.12: Architecture of a radial basis function network

In this section, we will introduce the concept of the so-called *Radial Basis Function* (RBF) networks and highlight their architecture. The RBF network [6] consists of three different layers, as shown in Figure 2.12. The input layer is constituted by p source nodes. A set of M nonlinear activation functions $\varphi_i, i = 1, \ldots, M$, constitutes the hidden second layer. The output of the network is provided by the third layer, which is comprised of output nodes. Figure 2.12 shows only one output node, in order to simplify our analysis. This construction is based on the basic neural network design. As suggested by the terminology, the activation functions in the hidden layer take the form of radial basis functions [6]. Radial functions are characterized by their responses that decrease or increase monotonically with distance from

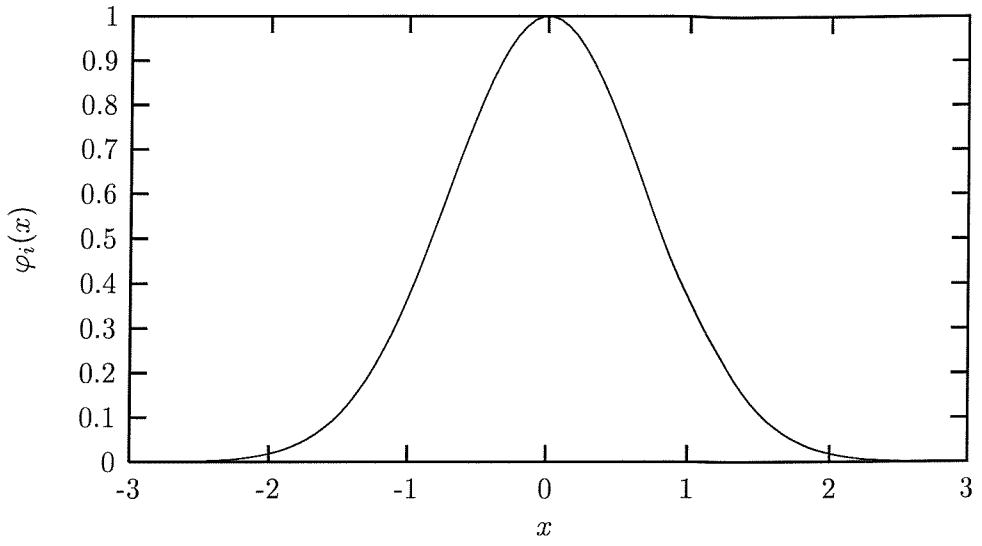


Figure 2.13: Gaussian radial basis function described by Equation 2.30 with centre $c_i = 0$ and spread of $2\sigma_i^2 = 1$.

a central point, $\mathbf{c}$, i.e. as the Euclidean norm $\|\mathbf{x}-\mathbf{c}\|$ is increased, where $\mathbf{x} = [x_1\ x_2\ \ldots\ x_p]^T$ is the input vector of the RBF network. The central points in the vector $\mathbf{c}$ are often referred to as the RBF centres. Therefore, the radial basis functions take the form

$$\varphi_i(\mathbf{x}) = \varphi(\|\mathbf{x} - \mathbf{c}_i\|), \qquad i = 0, \ldots, M, \tag{2.29}$$

where M is the number of independent basis functions in the RBF network. This justifies the 'radial' terminology. A typical radial function is the Gaussian function which assumes the form:

$$\varphi_i(\mathbf{x}) = exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right), \qquad i = 0, \ldots, M, \tag{2.30}$$

where $2\sigma_i^2$ is representative of the 'spread' of the Gaussian function that controls the radius of influence of each basis function. Figure 2.13 illustrates a Gaussian RBF, in the case of a scalar input, having a scalar centre of $c = 0$ and a spread or width of $2\sigma_i^2 = 1$. Gaussian-like RBFs are localized , i.e. they give a significant response only in the vicinity of the centre and $\varphi(x) \to 0$ as $x \to \infty$. As well as being localized, Gaussian basis functions have a number of useful analytical properties, which will be highlighted in our following discourse.

Referring to Figure 2.12, the RBF network can be represented mathematically as follows:

$$F(\mathbf{x}) = \sum_{i=0}^{M} w_i \varphi_i(\mathbf{x}). \tag{2.31}$$

The bias b in Figure 2.12 is absorbed into the summation as w_0 by including an extra basis function φ_0, whose activation function is set to 1. Bishop [33] gave an insight into the

role of the bias w_0 when the network is trained by minimizing the sum-of-squared error between the RBF network output vector and the desired output vector. The bias is found to compensate for the difference between the mean of the RBF network output vector and the corresponding mean of the target data evaluated over the training data set.

Note that the relationship between the RBF network and the Bayesian equalisation solution expressed in Equation 2.18, can be given explicitly. The RBF network's bias is set to $b = w_0 = 0$. The RBF centres $\mathbf{c}_i, i = 1, \ldots, M$, are in fact the noise-free dispersion-induced channel output vectors $\mathbf{r}_i, i = 1, \ldots, n_s$ indicated by circles and crosses, respectively, in Figure 2.3 and the number of hidden nodes M of Figure 2.12 corresponds to the number of desired channel output vectors, n_s, i.e. $M = n_s$. The RBF weights $w_i, i = 1, \ldots, M$, are all known from Equation 2.18 and they correspond to the scaling factors of the conditional probability density functions in Equation 2.18. Section 2.9.1 will provide further exposure to these issues.

Having described briefly the RBF network architecture, the next few sections will present its design in detail and also motivate its employment from the point of view of classification problems, interpolation theory and regularization. The design of the hidden layer of the RBF is justified by Cover's Theorem [48] which will be described in Section 2.7.2. In Section 2.7.3, we consider the so-called interpolation problem in the context of RBF networks. Then, we discuss the implications of sparse and noisy training data in Section 2.7.4. The solution to the problem of using regularization theory is also presented there. Lastly, in Section 2.7.5, the generalized RBF network is described, which concludes this section.

2.7.2 Cover's Theorem

The design of the radial basis function network is based on a curve-fitting (*approximation*) problem in a high-dimensional space, a concept, which was augmented for example by Haykin [6]. Specifically, the RBF network solves a complex pattern-classification problem, such as the one described in Section 2.2 in the context of Figure 2.3 for equalisation, by first transforming the problem into a high-dimensional space in a nonlinear manner and then by finding a surface in this multi-dimensional space that best fits the training data, as it will be explained below. The underlying justification for doing so is provided by *Cover's theorem* on the *separability of patterns*, which states that [48]:

> a complex pattern-classification problem non-linearly cast in a high-dimensional space is more likely to become linearly separable, than in a low-dimensional space.

We commence our discourse by highlighting the pattern-classification problem. Consider a surface that separates the space of the noisy channel outputs of Figure 2.3 into two

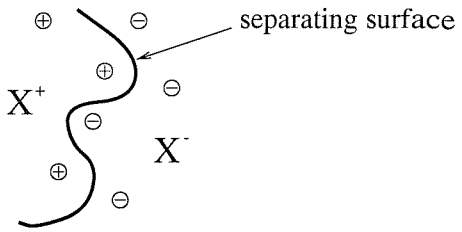


Figure 2.14: Pattern-classification into two dimensions, where the patterns are linearly non-separable, since a line cannot separate all the X^+ and X^- values, but the non-linear separating surface can – hence the term nonlinearly separable

regions or classes. Let X denote a set of N patterns or points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, each of which is assigned to one of two classes, namely X^+ and X^-. This dichotomy or binary partition of the points with respect to a surface becomes successful, if the surface separates the points belonging to the class X^+ from those in the class X^-. Thus, to solve the pattern-classification problem, we need to provide this *separating surface* that gives the decision boundary, as shown in Figure 2.14.

We will now non-linearly cast the problem of separating the channel outputs into a high-dimensional space by introducing a vector constituted by a set of real-valued functions $\varphi_i(\mathbf{x})$, where $i = 1, 2, \ldots, M$, for each input pattern $\mathbf{x} \in X$, as follows:

$$\boldsymbol{\varphi}(\mathbf{x}) = [\varphi_1(\mathbf{x}) \ \varphi_2(\mathbf{x}) \ \ldots \ \varphi_M(\mathbf{x})]^T, \tag{2.32}$$

where pattern $\mathbf{x}$ is a vector in a p-dimensional space and M is the number of real-valued functions. Recall that in our approach M is the number of possible channel output vectors for Bayesian equalisation solution. The vector $\boldsymbol{\varphi}(\mathbf{x})$ maps points of $\mathbf{x}$ from the p-dimensional input space into corresponding points in a new space of dimension M, where $p < M$. The function $\varphi_i(\mathbf{x})$ of Figure 2.12 is referred to as a *hidden function*, which plays a role similar to a hidden unit in a feedforward neural network, such as that in Figure 2.6(b). A dichotomy X^+, X^- of X is said to be *φ-separable*, if there exists an M-dimensional vector $\mathbf{w}$, such that for the scalar product $\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x})$ we may write

$$\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}) \geq 0, \qquad \text{if } \mathbf{x} \in X^+ \tag{2.33}$$

and

$$\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}) < 0, \qquad \text{if } \mathbf{x} \in X^-. \tag{2.34}$$

The hypersurface defined by the equation

$$\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}) = 0 \tag{2.35}$$

describes the separating surface in the φ space. The inverse image of this hypersurface is

$$\{\mathbf{x} : \mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}) = 0\}, \tag{2.36}$$

which defines the separating surface in the input space.

Below we give a simple example in order to visualise the concept of Cover's theorem in the context of the separability of patterns. Let us consider the XOR problem of Table 2.2, which is not linearly separable since the XOR = 0 and XOR = 1 points of Figure 2.15(a) cannot be separated by a line. The XOR problem is transformed into a linearly separable

x_1	x_2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Table 2.2: XOR truth table

problem by casting it from a two-dimensional input space into a three-dimensional space by the function $\varphi(\mathbf{x})$, where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ and $\boldsymbol{\varphi} = \begin{bmatrix} \varphi_1 & \varphi_2 & \varphi_3 \end{bmatrix}^T$. The hidden functions of Figure 2.12 are given in our example by:

$$\varphi_1(\mathbf{x}) = x_1, \tag{2.37}$$

$$\varphi_2(\mathbf{x}) = x_2, \tag{2.38}$$

$$\varphi_3(\mathbf{x}) = x_1 x_2. \tag{2.39}$$

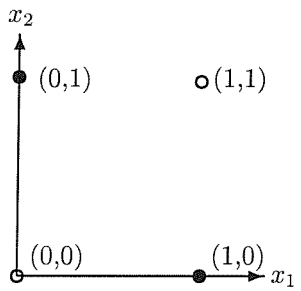
The higher-dimensional φ-inputs and the desired XOR output are shown in Table 2.3.

φ_1	φ_2	φ_3	XOR
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

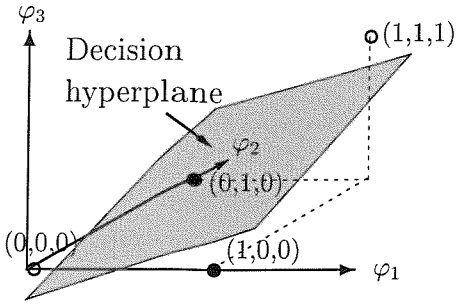
Table 2.3: XOR truth table with inputs of φ_1, φ_2 and φ_3.

Figure 2.15(b) illustrates, how the higher-dimensional XOR problem can be solved with the aid of a linear separating surface. Note that $\varphi_i, i = 1, 2, 3$ given in the above example are not of the radial basis function type described in Equation 2.29. They are invoked as a simple example to demonstrate the general concept of Cover's theorem.

Generally, we can find a non-linear mapping $\varphi(\mathbf{x})$ of sufficiently high dimension M, such that we have linear separability in the φ-space. It should be stressed, however that in some cases the use of nonlinear mapping may be sufficient to produce linear separability without having to increase the dimensionality of the hidden unit space [6].



(a) XOR problem, which is not linearly separable.



(b) XOR problem mapped to the three-dimensional space by the function $\varphi(\mathbf{x})$. The mapped XOR problem is linearly separable.

Figure 2.15: The XOR problem solved by $\varphi(\mathbf{x})$ mapping. Bold dots represent XOR = 1, while hollow dots correspond to XOR = 0.

2.7.3 Interpolation Theory

From the previous section, we note that the RBF network can be used to solve a nonlinearly separable classification problem. In this section, we highlight the use of the RBF network for performing *exact interpolation* of a set of data points in a multi-dimensional space. The exact interpolation problem requires every input vector to be mapped exactly onto the corresponding target vector, and forms a convenient starting point for our discussion of RBF networks. In the context of channel equalisation we could view the problem as attempting to map the channel output vector of Equation 2.4 to the corresponding transmitted symbol.

Consider a feedforward network with an input layer having p inputs, a single hidden layer and an output layer with a single output node. The network of Figure 2.12 performs a nonlinear mapping from the input space to the hidden space, followed by a linear mapping from the hidden space to the output space. Overall, the network represents a mapping from the p-dimensional input space to the one-dimensional output space, written as

$$s : \mathbb{R}^p \rightarrow \mathbb{R}^1, \tag{2.40}$$

where the mapping s is described by a continuous hypersurface $\mathbf{\Gamma} \subset \mathbb{R}^{p+1}$. The continuous surface $\mathbf{\Gamma}$ is a multi-dimensional plot of the output as a function of the input. Figure 2.16 illustrates the mapping $F(x)$ from a single-dimensional input space x to a single-dimensional output space and the surface $\mathbf{\Gamma}$. Again, in the case of an equaliser, the mapping surface $\mathbf{\Gamma}$ maps the channel output to the transmitted symbol.

In practical situations, the continuous surface $\mathbf{\Gamma}$ is unknown and the training data might be contaminated by noise. The network undergoes a so-called *learning process*, in order to find the specific surface in the multi-dimensional space that provides the best fit to the

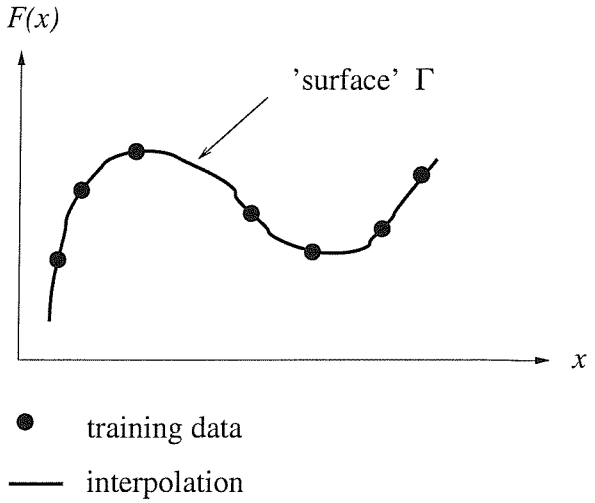


Figure 2.16: Stylised exact interpolation between the known input-output pairs by the continuous surface $\mathbf{\Gamma}$

training data d_i where $i = 1, 2, \ldots, N$. The 'best fit' surface is then used to interpolate the test data or for the specific case of an equaliser, the estimated transmitted symbol. Formally, the learning process can be categorized into two phases, the training phase and the generalisation phase. During the training phase, the fitting procedure for the surface $\mathbf{\Gamma}$ is optimised based on N known data points presented to the neural network in the form of input-output pairs $[\mathbf{x}_i, d_i], i = 1, 2, \ldots N$. The generalization phase constitutes the interpolation between the data points, where the interpolation is performed along the constrained surface generated by the fitting procedure, as the optimum approximation to the true surface $\mathbf{\Gamma}$.

Thus, we are led to the theory of multivariable interpolation in high-dimensional spaces. Assuming a single-dimensional output space, the interpolation problem can be stated as follows:

> Given a set of N different points $\mathbf{x}_i \in \mathbb{R}^p, i = 1, 2, \ldots, N$, in the p-dimensional input space and a corresponding set of N real numbers $d_i \in \mathbb{R}^1, i = 1, 2, \ldots, N$, in the one-dimensional output space, find a function $F : \mathbb{R}^p \rightarrow \mathbb{R}^1$ that satisfies the interpolation condition:

$$F(\mathbf{x}_i) = d_i, \qquad i = 1, 2, \ldots, N, \tag{2.41}$$

implying that for $i = 1, 2, \ldots, N$ the function $F(\mathbf{x})$ interpolates between the values d_i. Note that for exact interpolation, the interpolating surface is constrained to pass through all the training data points $\mathbf{x}_i$. The RBF technique is constituted by choosing a function $F(x)$

that obeys the following form:

$$F(\mathbf{x}) = \sum_{i=1}^{N} w_i \varphi(\|\mathbf{x} - \mathbf{x}_i\|), \tag{2.42}$$

where $\varphi_i(\mathbf{x}) = \varphi(\|\mathbf{x} - \mathbf{x}_i\|), i = 1, 2, \ldots, N$, is a set of N nonlinear functions, known as the radial basis function, and $\|.\|$ denotes the distance *norm* that is usually taken to be Euclidean. The known training data points $\mathbf{x}_i \in \mathbb{R}^p, i = 1, 2, \ldots, N$ constitute the centroids of the radial basis functions. The unknown coefficients w_i represent the weights of the RBF network of Figure 2.12. In order to link Equation 2.42 with Equation 2.31 we note that the number of radial basis functions M is now set to the number of training data points N and the RBF centres $\mathbf{c}_i$ of Equation 2.29 are equivalent to the training data points $\mathbf{x}_i$, i.e., $\mathbf{c}_i = \mathbf{x}_i, i = 1, 2, \ldots N$. The term associated with $i = 0$ was not included in Equation 2.42, since we argued above that the RBF bias was $w_0 = 0$.

Upon inserting the interpolation conditions of Equation 2.41 in Equation 2.42, we obtain the following set of simultaneous linear equations for the unknown weights w_i:

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1N} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{N1} & \varphi_{N2} & \cdots & \varphi_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix}, \tag{2.43}$$

where

$$\varphi_{ji} = \varphi(\|\mathbf{x}_j - \mathbf{x}_i\|), \qquad j, i = 1, 2, \ldots, N. \tag{2.44}$$

Let

$$\mathbf{d} = [d_1, d_2, \ldots, d_N]^T \tag{2.45}$$

$$\mathbf{w} = [w_1, w_2, \ldots, w_N]^T, \tag{2.46}$$

where the N-by-1 vectors $\mathbf{d}$ and $\mathbf{w}$ represent the equaliser's desired response vector and the linear weight vector, respectively. Let $\mathbf{\Phi}$ denote an N-by-N matrix with elements of $\varphi_{ji}, j, i = 1, 2, \ldots, N$, which we refer to as the *interpolation matrix*, since it generates the interpolation $F(\mathbf{x}_i) = d_i$ through Equation 2.41 and Equation 2.42 using the weights w_i. Then Equation 2.43 can be written in the compact form of:

$$\mathbf{\Phi w} = \mathbf{d}. \tag{2.47}$$

We note that if the data points d_i are all distinct and the interpolation matrix $\mathbf{\Phi}$ is positive definite, implying that all of its elements are positive and hence $\mathbf{\Phi}$ is invertible, then we can solve Equation 2.47 to obtain the weight vector $\mathbf{w}$, which is formulated as:

$$\mathbf{w} = \mathbf{\Phi}^{-1}\mathbf{d}, \tag{2.48}$$

where $\mathbf{\Phi}^{-1}$ is the inverse of the interpolation matrix $\mathbf{\Phi}$.

From *Light's theorem* [49], there exists a class of radial basis functions that generates an interpolation matrix, which is positive definite. Specifically, Light's theorem applies to a range of functions, which include the *Gaussian functions* [49] of:

$$\varphi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right), \tag{2.49}$$

$$\varphi_{ji} = \exp(\frac{-\|\mathbf{x}_j - \mathbf{x}_i\|^2}{2\sigma^2}), \qquad j,i = 1,2,\ldots,N, \tag{2.50}$$

where σ^2 is the variance of the Gaussian function. Hence the elements φ_{ji} of $\mathbf{\Phi}$ can be determined from Equation 2.50. Since $\mathbf{\Phi}$ is invertible, it is always possible to generate the weight vector $\mathbf{w}$ for the RBF network from Equation 2.48, in order to provide the interpolation through the training data.

In an equalisation context, exact interpolation can be problematic. The training data are sparse and are contaminated by noise. This problem will be addressed in the next section.

2.7.4 Regularization Theory

The partitioning hyper-surface and the interpolation hyper-surface mentioned in the previous sections was reconstructed or approximated from a given set of data points that may be sparse or noisy during learning. Therefore, the learning process used to reconstruct or approximate the classification hyper-surface can be seen as belonging to a generic class of problems referred to as *inverse problems* [6].

An inverse problem may be 'well-posed' or 'ill-posed'. In order to explain the term 'well-posed', assume that we have a domain X and a range Y taken to be spaces obeying the properties of metrics and they are related to each other by a fixed but unknown mapping $Y = F(X)$. The problem of reconstructing the mapping F is said to be *well-posed*, if the following conditions are satisfied [50]:

1. *Existence:* For every input vector $\mathbf{x} \in X$, there exists an output $y = F(\mathbf{x})$, where $y \in Y$, as seen in Figure 2.17.

2. *Uniqueness:* For any pair of input vectors $\mathbf{x}, \mathbf{t} \in X$, we have $F(\mathbf{x}) = F(\mathbf{t})$ if, and only if, $\mathbf{x} = \mathbf{t}$.

3. *Continuity:* The mapping is continuous.

If these conditions are not satisfied, the inverse problem of identifying x giving rise to y is said to be ill-posed.

Learning, where the partitioning or interpolation hyper-surface is approximated, is in general an ill-posed inverse problem . This is because the uniqueness criterion may be

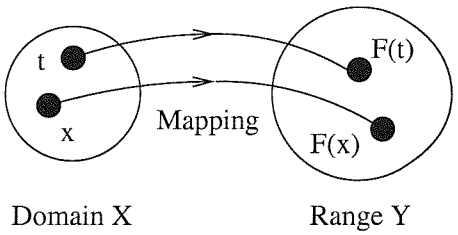


Figure 2.17: The mapping of the input domain X onto the output range Y

violated, since there may be insufficient information in the training data to reconstruct the input-output mapping uniquely. Furthermore, the presence of noise or other impairments in the input data adds uncertainty to the reconstructed input-output mapping. This is the case in the context of the equalisation problem.

Tikhonov [51] proposed a method referred to as *regularization for solving ill-posed problems*. The basic idea of regularization is to *stabilize* the solution by means of some auxiliary non-negative function that imposes prior restrictions such as, smoothness or correlation constraints on the input-output mapping and thereby converting an ill-posed problem into a well-posed problem. This approach was treated in depth by Poggio and Girosi [52].

According to Tikhonov's regularization theory [51], the previously introduced function F is determined by minimising a *cost function* $\mathcal{E}(F)$, defined by

$$\mathcal{E}(F) = \mathcal{E}_s(F) + \lambda \mathcal{E}_c(F), \tag{2.51}$$

where λ is a positive real number referred to as the *regularization parameter* and the two terms involved are [51]:

1. *Standard Error Term:* This term, denoted by $\mathcal{E}_s(F)$, quantifies the standard error between the desired response d_i and the actual response y_i for training samples $i = 1, 2, \ldots, N$. It is defined by

$$\begin{aligned} \mathcal{E}_s(F) &= \frac{1}{2}\sum_{i=1}^{N}(d_i - y_i)^2 \\ &= \frac{1}{2}\sum_{i=1}^{N}[d_i - F(\mathbf{x}_i)]^2. \end{aligned} \tag{2.52}$$

2. *Regularizing Term:* This term, denoted by $\mathcal{E}_c(F)$, depends on the geometric properties of the approximation function $F(\mathbf{x})$. It provides the so-called *a priori* smoothness constraint and it is defined by

$$\mathcal{E}_c(F) = \frac{1}{2}\|\mathcal{P}F\|^2 \tag{2.53}$$

where $\mathcal{P}$ is a linear (pseudo) differential operator, referred to as a *stabilizer* [6], which stabilizes the solution F, rendering it smooth and therefore continuous.

The regularization parameter λ indicates, whether the given training data set is sufficiently extensive in order to specify the solution $F(\mathbf{x})$. The limiting case $\lambda \to 0$ implies that the problem is unconstrained. Here, the solution $F(\mathbf{x})$ is completely determined from the given data set. The other limiting case, $\lambda \to \infty$, implies that the *a priori* smoothness constraint is sufficient to specify the solution $F(\mathbf{x})$. In other words, the training data set is unreliable. In practical applications the regularization parameter λ is assigned a value between the two limiting conditions, so that both the sample data and the *a priori* information contribute to the solution $F(\mathbf{x})$.

The minimisation of the cost function $\mathcal{E}(F)$ by evaluating the derivative of $\mathcal{E}(F)$ in Equation 2.51 provides the following solution to $F(\mathbf{x})$ [6]:

$$\begin{aligned} F(\mathbf{x}) &= \frac{1}{\lambda}\sum_{i=1}^{N}[d_i - F(\mathbf{x}_i)]G(\mathbf{x};\mathbf{x}_i) \\ &= \sum_{i=1}^{N} w_i G(\mathbf{x};\mathbf{x}_i), \end{aligned} \tag{2.54}$$

where $G(\mathbf{x};\mathbf{x}_i)$ denotes the so-called Green function centred at $\mathbf{x}_i$ and $w_i = \frac{1}{\lambda}[d_i - F(\mathbf{x}_i)]$. Equation 2.54 states that the solution $F(\mathbf{x})$ to the regularization problem is a linear superposition of N number of Green functions centred at the training data points $x_i, i = 1, 2, \ldots, N$. The weights w_i are the *coefficients of the expansion* of $F(\mathbf{x})$ in terms of $G(\mathbf{x};\mathbf{x}_i)$ and x_i are the *centres of the expansion* for $i = 1, 2, \ldots, N$. The centres $\mathbf{x}_i$ of the Green functions used in the expansion are the given data points used in the training process.

We now have to determine the unknown expansion cofficients w_i denoted by

$$w_i = \frac{1}{\lambda}[d_i - F(\mathbf{x}_i)], \qquad i = 1, 2, \ldots, N. \tag{2.55}$$

Let

$$\mathbf{F} = [F(\mathbf{x}_1), F(\mathbf{x}_2), \ldots, F(\mathbf{x}_N)]^T, \tag{2.56}$$

$$\mathbf{d} = [d_1, d_2, \ldots, d_N]^T, \tag{2.57}$$

$$G = \begin{bmatrix} G(\mathbf{x}_1;\mathbf{x}_1) & G(\mathbf{x}_1;\mathbf{x}_2) & \ldots & G(\mathbf{x}_1;\mathbf{x}_N) \\ G(\mathbf{x}_2;\mathbf{x}_1) & G(\mathbf{x}_2;\mathbf{x}_2) & \ldots & G(\mathbf{x}_2;\mathbf{x}_N) \\ \vdots & \vdots & & \vdots \\ G(\mathbf{x}_N;\mathbf{x}_1) & G(\mathbf{x}_N;\mathbf{x}_2) & \ldots & G(\mathbf{x}_N;\mathbf{x}_N) \end{bmatrix}, \tag{2.58}$$

$$\mathbf{w} = [w_1, w_2, \ldots, w_N]^T. \tag{2.59}$$

Rewriting Equation 2.55 and Equation 2.54 in matrix form, we obtain respectively:

$$\mathbf{w} = \frac{1}{\lambda}(\mathbf{d} - \mathbf{F}) \tag{2.60}$$

and

$$\mathbf{F} = \mathbf{G}\mathbf{w}. \tag{2.61}$$

Upon substituting Equation 2.61 into Equation 2.60, we get

$$(\mathbf{G} + \lambda\mathbf{I})\mathbf{w} = \mathbf{d}, \tag{2.62}$$

where $\mathbf{I}$ is the N-by-N identity matrix.

Invoking Light's Theorem [49] from Section 2.7.3, we may state that the matrix $\mathbf{G}$ is positive definite for certain classes of Green functions, provided that the data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ are distinct. The classes of Green functions covered by Light's theorem include the so-called multi-quadrics and Gaussian functions [6]. In practice, λ is chosen to be sufficiently large to ensure that $\mathbf{G} + \lambda\mathbf{I}$ is positive definite and therefore, invertible. Hence, the linear Equation 2.62 will have a unique solution given by

$$\mathbf{w} = (\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{d}. \tag{2.63}$$

The set of Green functions used is characterized by the specific form adopted for the stabilizer $\mathcal{P}$ and the associated boundary conditions [6]. By defination, if the stabilizer $\mathcal{P}$ is translationally invariant, then the Green function $G(\mathbf{x}; \mathbf{x}_i)$ centred at $\mathbf{x}_i$ will depend only on the difference between the argument $\mathbf{x}$ and $\mathbf{x}_i$, i.e.:

$$G(\mathbf{x}; \mathbf{x}_i) = G(\mathbf{x} - \mathbf{x}_i). \tag{2.64}$$

If the stabilizer $\mathcal{P}$ is to be both *translationally and rotationally invariant*, then the Green function $G(\mathbf{x}; \mathbf{x}_i)$ will depend only on the *Euclidean norm* of the difference vector $\mathbf{x} - \mathbf{x}_i$, formulated as:

$$G(\mathbf{x}; \mathbf{x}_i) = G(\|\mathbf{x} - \mathbf{x}_i\|). \tag{2.65}$$

Under these conditions, the Green function must be a *radial basis function.* Therefore, the regularized solution of Equation 2.54 takes on the form:

$$F(\mathbf{x}) = \sum_{i=1}^{N} w_i G(\|\mathbf{x} - \mathbf{x}_i\|). \tag{2.66}$$

An example of a Green function, whose form is characterized by the differential operator $\mathcal{P}$ that is both translationally and rotationally invariant is the *multivariate Gaussian function* that obeys the following form

$$G(\mathbf{x}; \mathbf{x}_i) = exp\left(-\frac{1}{2\sigma_i^2}\|\mathbf{x} - \mathbf{x}_i\|^2\right), \qquad i = 1, \ldots, N. \tag{2.67}$$

Equation 2.67 is characterized by a *mean vector* $\mathbf{x}_i$ and common *variance* σ_i^2.

It is important to realize that the solution described by Equation 2.66 differs from that of Equation 2.42. The solution of Equation 2.66 is *regularized* by the definition given in Equation 2.63 for the weight vector $\mathbf{w}$. The two solutions are the same only if the regularization parameter λ is equal to zero. The regularization parameter λ provides the smoothing effect in constructing the partition or interpolation hyper-surface during the learning process.

Typically, the number of training data symbols is higher than the number of basis functions required for the RBF network to give an acceptable approximation to the interpolation solution. The generalized RBF network is introduced to address this problem and its structure is discussed in the following section.

2.7.5 Generalized Radial Basis Function Networks

The one-to-one correspondence between the training input data $\mathbf{x}_i$ and the Green function $G(\mathbf{x};\mathbf{x}_i)$ for $i = 1, 2, \ldots, N$ is prohibitively expensive to implement in computational terms for large N values. Especially the computation of the linear weights w_i is computationally demanding, which requires the inversion of an N-by-N matrix according to Equation 2.63. In order to overcome these computational difficulties, the complexity of the RBF network would have to be reduced and this requires an approximation to the regularized solution.

The approach followed here involves seeking a suboptimal solution in a lower-dimensional space that approximates the regularized solution described by Equation 2.54. This can be achieved using *Galerkin's method* [6]. According to this technique, the approximated solution $F^*(\mathbf{x})$ is expanded using a reduced $M \leq N$ number of basis functions, as follows:

$$F^*(\mathbf{x}) = \sum_{i=1}^{M} w_i \varphi_i(\mathbf{x}), \tag{2.68}$$

where $\varphi_i(\mathbf{x}), i = 1, 2, \ldots, M$, is a new set of basis functions. The number M of the basis functions M is typically less than the number of data points N and the coefficients w_i constitute a new set of weights. Using radial basis functions, we set

$$\varphi_i(\mathbf{x}) = G(\|\mathbf{x} - \mathbf{c}_i\|), \qquad i = 1, 2, \ldots, M, \tag{2.69}$$

where $\mathbf{c}_i, i = 1, 2, \ldots, M$, is the set of RBF centres to be determined. Thus, with the aid of Equation 2.68 and Equation 2.69 we have

$$\begin{aligned} F^*(\mathbf{x}) &= \sum_{i=1}^{M} w_i G(\mathbf{x};\mathbf{c}_i) \\ &= \sum_{i=1}^{M} w_i G(\|\mathbf{x} - \mathbf{c}_i\|). \end{aligned} \tag{2.70}$$

Now the problem we have to address is the determination of the new set of weights $w_i, i = 1, 2, \ldots, M$, based on a reduced number of $M \leq N$ basis functions so as to minimize the new cost function $\xi(F^*)$ according to Tikhonov's cost function of Equation 2.51. This new cost function is defined by

$$\xi(F^*) = \sum_{i=1}^{N} \left(d_i - \sum_{j=1}^{M} w_j G(\|\mathbf{x}_i - \mathbf{c}_j\|) \right)^2 + \lambda \|\mathcal{P}F^*\|^2. \tag{2.71}$$

Minimizing Equation 2.71 with respect to the weight vector $\mathbf{w}$ yields [6]:

$$(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{G}_0)\mathbf{w} = \mathbf{G}^T\mathbf{d}, \tag{2.72}$$

where

$$\mathbf{d} = [d_1, d_2, \ldots, d_N]^T, \tag{2.73}$$

$$\mathbf{G} = \begin{bmatrix} G(\mathbf{x}_1;\mathbf{c}_1) & G(\mathbf{x}_1;\mathbf{c}_2) & \ldots & G(\mathbf{x}_1;\mathbf{c}_M) \\ G(\mathbf{x}_2;\mathbf{c}_1) & G(\mathbf{x}_2;\mathbf{c}_2) & \ldots & G(\mathbf{x}_2;\mathbf{c}_M) \\ \vdots & \vdots & & \vdots \\ G(\mathbf{x}_N;\mathbf{c}_1) & G(\mathbf{x}_N;\mathbf{c}_2) & \ldots & G(\mathbf{x}_N;\mathbf{c}_M) \end{bmatrix}, \tag{2.74}$$

$$\mathbf{w} = [w_1, w_2, \ldots, w_M]^T, \tag{2.75}$$

$$\mathbf{G}_0 = \begin{bmatrix} G(\mathbf{c}_1;\mathbf{c}_1) & G(\mathbf{c}_1;\mathbf{c}_2) & \ldots & G(\mathbf{c}_1;\mathbf{c}_M) \\ G(\mathbf{c}_2;\mathbf{c}_1) & G(\mathbf{c}_2;\mathbf{c}_2) & \ldots & G(\mathbf{c}_2;\mathbf{c}_M) \\ \vdots & \vdots & & \vdots \\ G(\mathbf{c}_M;\mathbf{c}_1) & G(\mathbf{c}_M;\mathbf{c}_2) & \ldots & G(\mathbf{c}_M;\mathbf{c}_M) \end{bmatrix}. \tag{2.76}$$

Here, the matrix $\mathbf{G}$ is a non-symmetric N-by-M matrix and the matrix $\mathbf{G}_0$ is a symmetric M-by-M matrix. Thus, upon solving Equation 2.72 to obtain the weights $\mathbf{w}$, we get:

$$\mathbf{w} = (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{G}_0)^{-1}\mathbf{G}^T\mathbf{d}. \tag{2.77}$$

Observe that the solution in Equation 2.77 is different from Tikhonov's solution in Equation 2.63. Specifically, in Equation 2.58 the matrix $\mathbf{G}$ is a symmetric N-by-N matrix, while in Equation 2.74 it is a non-symmetric N-by-M matrix.

By introducing a number of modifications to the exact interpolation procedure presented in Section 2.7.3 we obtain the generalized radial basis function network model that provides a smooth interpolating function, in which the number of basis functions is determined by the affordable complexity of the mapping to be represented, rather than by the size of the data set. The modifications which are required are as follows:

1. The number of basis functions, M, need not be equal to the number of training data points, N.

2. In contrast to Equation 2.42, the centres of the basis functions are no longer constrained to be given by N training input data points $\mathbf{x}_i$. Thus, the position of the centres of the radial basis functions $\mathbf{c}_i, i = 1, 2, \ldots, M$, in Equation 2.70 are the unknown parameters that have to be 'learned' together with the weights of the output layer $w_i, i = 1, 2, \ldots, M$. A few methods of obtaining the RBF centres are as follows: random selection from the training data, the so-called Orthogonal Least Squares (OLS) learning algorithm of Chen, Cowan, Grant *et al.* [53, 54] and the well-known K-means clustering algorithm [8]. We opted for using the K-means clustering algorithm in order to learn the RBF centres in our equalisation problem and this algorithm will be described in more detail in Section 2.8.

3. Instead of having a common RBF spread or width parameter $2\sigma^2$, as described in Equation 2.49, each basis function is given its own width $2\sigma_i^2$, as in Equation 2.67. The value of the spread or width is determined during training. Bishop [33] noted that based on noisy interpolation theory, it is a useful rule of thumb when designing the RBF network with good generalization properties to set the width $2\sigma_i^2$ of the RBF large in relation to the spacing of the RBF input data.

Here, the new set of RBF network parameters, $\mathbf{c}_i$, σ_i^2, and w_i, where $1 \leq i \leq M \leq N$, can be learnt in a sequential fashion. For example, a clustering algorithm can be used to estimate the RBF centres, $\mathbf{c}_i$. Then, an estimate of the variance of the input vector with respect to each centre provides the width parameter, σ_i^2. Finally, we can calculate the RBF weights w_i using Equation 2.77 or adaptively using the LMS algorithm [6].

Note that apart from regularization, an alternative way of reducing the number of basis functions required and thus reduce the associated complexity is to use the OLS learning procedure proposed by Chen, Cowan and Grant [53]. This method is based on viewing the RBF network as a linear regression model, where the selection of RBF centres is regarded as a problem of subset selection. The OLS method, employed as a forward regression procedure, selects a suitable set of RBF centres, which are referred to as the regressors, from a large set of candidates for the training data, yielding $M < N$. As a further advance, Chen, Chng and Alkadhimi [54] proposed a regularised OLS learning algorithm for RBFs that combines the advantages of both the OLS and the regularization method. Indeed, it was OLS training that was used in the initial application of RBF networks to the channel equalisation problem [27]. Instead of using the regularised interpolation method, we opted for invoking detection theory, in order to solve the equalisation problem with the aid of RBF networks. This will be expounded further in Section 2.9.

Having described and justified the design of the RBF network of Figure 2.12 that was

previously introduced in Section 2.7.1, in the next section the K-means clustering algorithm used to learn the RBF centres and to partition the RBF network input data into K subgroups or clusters is described briefly.

2.8 K-means Clustering Algorithm

In general, the task of the K-means algorithm [55] is to partition the domain of arbitrary vectors into K regions and then to find a centroid-like reference vector, $\mathbf{c}_i, i = 1, \ldots, K$, that best represents the set of vectors in each region or partition. In the RBF network based equaliser design the vectors to be clustered are the noisy channel state vectors $\mathbf{v}_k, k = -\infty, \ldots, \infty$ observed by the equaliser using the current tap vectors, such as those seen in Figure 2.3, where the centroid-like reference vectors are constituted by the optimal channel states $\mathbf{r}_i, i = 1, \ldots, n_s$, as described in the previous sections. Suppose that a set of input patterns $\mathbf{x}$ of the algorithm is contained in a domain $\mathbb{P}$. The K-means clustering problem is formulated as finding a partition of $\mathbb{P}$, $\mathbf{P} = [\mathbb{P}_1, \ldots, \mathbb{P}_K]$, and a set of reference vectors $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$ that minimize the cluster MSE cost function defined as follows:

$$\text{MSE}(\mathbf{P}, \mathbf{C}) = \sum_{i=1}^{K} \int_{\mathbb{P}_i} p(\mathbf{x}) \cdot \|\mathbf{x} - \mathbf{c}_i\|^2 d\mathbf{x}, \tag{2.78}$$

where $\| \ \|$ denotes the l_2 norm and $p(\mathbf{x})$ denotes the probability density function of $\mathbf{x}$.

Upon presenting a new training vector to the K-means algorithm, it repetitively updates both the reference vectors or centroids $\mathbf{c}_i$ and the partition $\mathbf{P}$. We define $\mathbf{c}_{i,k}$ and $\mathbf{x}_k$ as the ith reference vector and the current input pattern presented to the algorithm at time k. The adaptive K-means clustering algorithm computes the new reference vector $\mathbf{c}_{i,k+1}$ as

$$\mathbf{c}_{i,k+1} = \mathbf{c}_{i,k} + M_i(\mathbf{x}_k)\{\mu(\mathbf{x}_k - \mathbf{c}_{i,k})\}, \tag{2.79}$$

where μ is the learning rate governing the speed and accuracy of the adaptation and $M_i(\mathbf{x}_k)$ is the so-called membership indicator that specifies, whether the input pattern $\mathbf{x}_k$ belongs to region $\mathbb{P}_i$ and also, whether the ith neuron is active. In the traditional adaptive K-means algorithm the learning rate μ is typically a constant and the membership indicator $M_i(\mathbf{x})$ is defined as:

$$M_i(\mathbf{x}) = \begin{cases} 1 & \text{if } \|\mathbf{x} - \mathbf{c}_i\|^2 \leq \|\mathbf{x} - \mathbf{c}_j\|^2 \text{ for each } i \neq j \\ 0 & \text{otherwise.} \end{cases} \tag{2.80}$$

A serious problem associated with most K-means algorithm implementations is that the clustering process may not converge to an optimal or near-optimal configuration. The algorithm can only assure local optimality, which depends on the initial locations of the representative vectors. Some initial reference vectors get 'entrenched' in regions of the algorithm's input vector domain with few or no input patterns and may not move to where they

are needed. To deal with this problem, Rumelhart and Zipser [56] employed leaky learning, where in addition to adjusting the closest reference vector, other reference vectors are also adjusted, but inconjunction with smaller learning rates. Another approach, proposed by DeSieno and is referred to as the conscience algorithm [57] keeps track of how many times each reference vector has been updated in response to the algorithm's input vectors and if a reference vector gets updated or 'wins' too often, it will 'feel guilty' and therefore pulls itself out of the competition. Thus, the average rates of 'winning' for each region is equalized and no reference vectors can get 'entrenched' in that region. However, these two methods yield partitions that are not optimal with respect to the MSE cost function of Equation 2.78.

The performance of the adaptive K-means algorithm depends on the learning rate μ in Equation 2.79. There is a tradeoff between the *dynamic performance* (rate of convergence) and the *steady-state performance* (residual deviation from the optimal solution or excess MSE). When using a fixed learning rate, it must be sufficiently small for the adaptation to converge. The excess MSE is smaller at a lower learning rate. However, a smaller learning rate also results in a slower convergence rate. Because of this problem, adaptive K-means algorithms having variable learning rates have been investigated [58]. The traditional adaptive K-means algorithm can be improved by incorporating two mechanisms: by biasing the clustering towards an optimal partition and by adjusting the learning rate dynamically. The justification and explanation concerning how the two mechanisms are implemented is described in more detail by Chinrungrueng *et al.* [58].

Having described the K-means clustering algorithm, which can be used as the RBF network's learning algorithm, we proceed to further explore the RBF network structure in the context of an equaliser in the following Section.

2.9 Radial Basis Function Network Based Equalisers

2.9.1 Introduction

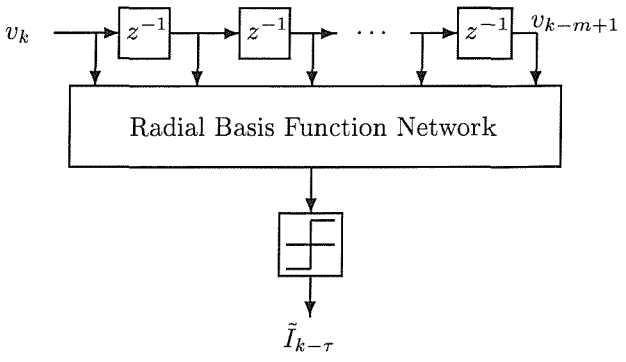


Figure 2.18: Radial Basis Function Equaliser for BPSK

The RBF network is ideal for channel equalisation applications, since it has an equivalent structure to the so-called optimal Bayesian equalisation solution of Equation 2.18 [8]. Therefore, RBF equalisers can be derived directly from theoretical considerations related to optimal detection and all our prior knowledge concerning detection problems [31] can be exploited. The neural network equaliser based on the MLP of Section 2.5, the polynomial perceptrons of Section 2.6 and on the so-called self-organizing map [42] constitutes a model-free classifier, thus requiring a long training period and large networks. The schematic of the RBF equaliser is depicted in Figure 2.18. The overall response of the RBF network of Figure 2.12, again, can be formulated as:

$$\begin{aligned} f_{RBF}(\mathbf{v}_k) &= \sum_{i=1}^{M} w_i \varphi(\|\mathbf{v}_k - \mathbf{c}_i\|), \\ \varphi(x) &= exp(-x^2/\rho), \end{aligned} \tag{2.81}$$

where $\mathbf{c}_i, i = 1, \ldots, M$ represents the RBF centres, which have the same dimensionality as the input vector $\mathbf{v}_k$, $\|\cdot\|$ denotes the Euclidean norm, $\varphi(\cdot)$ is the radial basis function introduced in Section 2.7, ρ are positive constants defined as the spread or width of the RBF in Section 2.7 (each of the RBFs has the same width, i.e., $2\sigma_i^2 = \rho$, since the received signal is corrupted by the same Gaussian noise source) and M is the number of hidden nodes of the RBF network. Note that the number of input nodes of the RBF network in Figure 2.12, p, is now equivalent to the order m of the equaliser, i.e. $p = m$, and the bias is set to $b = 0$. The detected symbol is given by:

$$\tilde{I}_{k-\tau} = sgn(f_{RBF}(\mathbf{v}_k)), \tag{2.82}$$

where the decision delay τ is introduced to facilitate causality in the equaliser and to provide the 'past' and the 'future' received samples, with respect to the 'delayed' detected symbol, for equalisation.

The relationship between the RBF network and the Bayesian equalisation solution expressed in Equation 2.18 can be established explicitly. The RBF centres $\mathbf{c}_i, i = 1, \ldots, M$ are in fact constituted by the noise-free channel output vectors $\mathbf{r}_i$ indicated by the circles and crosses in Figure 2.3, while the number of hidden nodes M in Figure 2.12 corresponds to the number of desired channel output vectors, n_s, i.e., $M = n_s$. The weights w_i correspond to the scaling factors of the conditional probability density functions in Equation 2.18 given by:

$$w_i = \begin{cases} p_i(2\pi\sigma_\eta^2)^{-m/2} & \text{if } \mathbf{r}_i \in V_{m,\tau}^+, \\ -p_i(2\pi\sigma_\eta^2)^{-m/2} & \text{if } \mathbf{r}_i \in V_{m,\tau}^-, \end{cases} \tag{2.83}$$

where p_i is the *a priori* probability of occurance for the noise-free channel output vector $\mathbf{r}_i$ and σ_η^2 is the noise variance of the Gaussian channel. For equiprobable transmitted binary

symbols the *a priori* probability of each state is identical. Therefore, the network can be simplified considerably in the context of binary signalling by fixing the RBF weights to $w_i = +1$, if the RBF centroids $\mathbf{c}_i$ correspond to a positive channel state $\mathbf{v}_i^+$ and to $w_i = -1$, if the centroids $\mathbf{c}_i$ correspond to a negative channel state $\mathbf{v}_i^-$. The widths ρ in Equation 2.81 are controlled by the noise variance and are usually set to $\rho = 2\sigma_\eta^2$, while $\varphi(\cdot)$ is the noise probability density function, which is usually Gaussian. When these conditions are met, the RBF network realizes precisely the Bayesian equalisation solution [8], a fact, which is augmented further below.

Specifically, in order to realize the optimal Bayesian solution using the RBF network, we have to identify the RBF centres or the noise-free channel output vectors. Chen *et al.* [8] achieved this using two alternative schemes. The first method identifies the channel model using standard linear adaptive CIR estimation algorithms such as for example Kalman filtering [59] and then calculates the corresponding CIR-specific noise-free vectors. The second method estimates these vectors or centres directly using so-called supervised learning – where training data are provided – and a decision-directed clustering algorithm [8, 26], which will be described in detail in Section 2.9.3.

The ultimate link between the RBF network and the Bayesian equaliser renders the RBF design an attractive solution to equalisation problems. The performance of the RBF equaliser is superior to that of the MLP and PP equalisers of Sections 2.5 and 2.6 and it needs a significantly shorter training period, than these nonlinear equalisers [8]. Furthermore, Equation 2.81 shows that RBF networks are linear in terms of the weight parameter w_i, while the non-linear RBFs $\varphi(x)$ are assigned to the hidden layer of Figure 2.12. The RBF network can be configured to have a so-called uni-modal error surface where f_{RBF} in Equation 2.81 exhibits only one minimum, namely the global minimum, with respect to its weights w_i, while also having a guaranteed convergence performance. The RBF equaliser is capable of equalising nonlinear channels, can be also adapted to non-Gaussian noise distributions. Furthermore, in a recursive form, referred to as the *recurrent RBF equaliser* [38], the equaliser can provide optimal decisions based on all the previous received samples, $v_{k-i}, i = 0, \ldots, \infty$, instead of only those previous received samples, $v_{k-i}, i = 0, \ldots, v_{k-m+1}$ which are within the equaliser's memory. The RBF equaliser can be used to compute the so-called *a posteriori* probabilities of the transmitted symbols, which are constituted by their correct detection probabilities. The advantages of using the *a posteriori* symbol probabilities for blind equalisation and tracking in time-variant environments have been discussed in several contributions [38, 60]. Furthermore, the *a posteriori* probabilities generated can be used to directly estimate the associated BER without any reference signal. The BER estimate can be used by the receiver as a measure of reliability of the data transmission process or even to control the transmission rate in variable rate digital modems or to invoke

a specific modulation in adaptive QAM systems.

Number of subractions and additions	$2n_s m - 1$
Number of multiplications	$n_s(m+1)$
Number of divisions	n_s
Number of exp()	n_s

Table 2.4: Computational complexity of a linear RBF network equaliser having m inputs and n_s hidden units per equalised output sample based on Equation 2.81. When the optimum Bayesian equaliser of Equation 2.18 is used, we have $n_s = 2^{L+m}$, while in Section 2.9.7 we will reduce the complexity of the RBF equaliser by reducing the value of n_s.

The drawback of RBF networks is, however, that their complexity, i.e. the number of neurons n_s in the hidden layer of Figure 2.12 grows dramatically, when the channel memory L and the equaliser order m increase, since $n_s = 2^{L+m}$. The vector subtraction $\mathbf{v}_k - \mathbf{c}_i$ in Equation 2.81 involves m subtraction operations, while the computation of the norm $\|\cdot\|^2$ of an m-element vector involves m multiplications and $m-1$ additions. Thus, the term $w_i\varphi(\|\mathbf{v}_k - \mathbf{c}_i\|)$ in Equation 2.81 requires $2m-1$ additions/subtractions, $m+1$ multiplications, one division and an $\exp(\cdot)$ operation. The summation $\sum_{i=1}^{M}$ in Equation 2.81 where $M = n_s$, involves $n_s - 1$ additions. Therefore the associated computational complexity of the RBF network equaliser based on Equation 2.81 is given in Table 2.4.

For non-stationary channels the values of the RBF centres, $\mathbf{c}_i$, will vary as a function of time and each centre must be re-calculated, before applying the decision function of Equation 2.81. Since $n_s = 2^{L+m}$ can be high, the evaluation of Equation 2.81 may not be practical for real-time applications. A range of methods proposed for reducing the complexity of the RBF network equaliser and to render it more suitable for realistic channel equalisation will be described in Section 2.9.7. Our simulations results will be presented in Section 2.12.

2.9.2 RBF-based Equalisation in Multilevel Modems

In the previous sections, the transmitted symbols considered were binary. In this section, based on the suggestions of Chen, McLaughlin and Mulgrew [25], we shall extend the design of the RBF equaliser to complex $\mathcal{M}$-ary modems, where the information symbols are selected from the set of $\mathcal{M}$ complex values, $\mathcal{I}_i, i = 1, 2, \ldots, \mathcal{M}$. An example is, when a Quadrature Amplitude Modulation (QAM) scheme [61] is used.

Since the delayed transmitted symbols $I_{k-\tau}$ in the schematic of Figure 2.18 may assume any of the legitimate $\mathcal{M}$ complex values, the channel input sequence $\mathbf{I}_k$, defined in Equation 2.5, produces $n_s = \mathcal{M}^{L+m}$ different possible values for the noise-free channel output

vector $\tilde{\mathbf{v}}_k$ of Figure 2.18 described in Equation 2.6, which were visualised for the binary case in Figure 2.3. The desired channel states can correspondingly be partitioned into $\mathcal{M}$ classes – rather than two – according to the value of the transmitted symbol $I_{k-\tau}$, which is formulated as follows:

$$\begin{aligned} V_{m,\tau}^i &= \{\tilde{\mathbf{v}}_k | I_{k-\tau} = \mathcal{I}_i\}, \\ &= \{\mathbf{r}_1^i, \ldots, \mathbf{r}_j^i, \ldots, \mathbf{r}_{n_s^i}^i\}, \qquad i = 1, 2, \ldots, \mathcal{M}, \end{aligned} \tag{2.84}$$

where $\mathbf{r}_j^i, j = 1, \ldots, n_s^i$, is the jth desired channel output state due to the $\mathcal{M}$-ary transmitted symbol $I_{k-\tau} = \mathcal{I}_i, i = 1, \ldots, \mathcal{M}$. More explicitly, the quantities n_s^i represent the number of channel states $\mathbf{r}_j^i$ in the set $V_{m,\tau}^i$. The number of channel states in any of the sets $V_{m,\tau}^i$ is identical for all the transmitted symbols $\mathcal{I}_i, i = 1, 2, \ldots, \mathcal{M}$, i.e. $n_s^i = n_s^j$ for $i \neq j$ and $i, j = 1, \ldots \mathcal{M}$. Lastly, we have $\sum_{i=1}^{\mathcal{M}} n_s^i = n_s$.

Thus, the optimal Bayesian decision solution of Equation 2.16 defined for binary signalling based on Bayes' decision theory [21] has to be redefined for $\mathcal{M}$-ary signalling as follows, in order to achieve the minimum error-probability:

$$\tilde{I}_{k-\tau} = \mathcal{I}_i^*, \qquad \text{if } \zeta_i^*(k) = \max\{\zeta_i(k), 1 \leq i \leq \mathcal{M}\}, \tag{2.85}$$

where $\zeta_i(k)$ is the decision variable based on the conditional density function given by:

$$\begin{aligned} \zeta_i(k) &= P(\mathbf{v}_k | I_{k-\tau} = \mathcal{I}_i) \cdot P(I_{k-\tau} = \mathcal{I}_i) \\ &= \sum_{j=1}^{n_s^i} p_j^i p(\mathbf{v}_k - \mathbf{r}_j^i), \qquad 1 \leq i \leq \mathcal{M}. \end{aligned} \tag{2.86}$$

The quantities $p_j^i, i = 1, \ldots, \mathcal{M}, j = 1, \ldots, n_s^i$ denote the *a priori* probability of appearance of each desired state $\mathbf{r}_j^i \in V_{m,\tau}^i$ associated with the transmitted $\mathcal{M}$-ary symbol $\mathcal{I}_i, i = 1, \ldots, \mathcal{M}$ and $p(\cdot)$ is the probability density function of the additive noise of the channel.

Thus, there are $\mathcal{M}$ neural 'subnets' associated with the $\mathcal{M}$ decision variables $\zeta_i(k) = P(\mathbf{v}_k | I_{k-\tau} = \mathcal{I}_i) \cdot P(I_{k-\tau} = \mathcal{I}_i), i = 1, 2, \ldots, \mathcal{M}$. The architecture of the RBF equaliser for the $\mathcal{M}$-ary multilevel modem scenario considered is shown in Figure 2.19. Note that the output of each sub-RBF network gives the corresponding conditional density function $\zeta_i(k) = P(\mathbf{v}_k | I_{k-\tau} = \mathcal{I}_i) \cdot P(I_{k-\tau} = \mathcal{I}_i)$ and this output value can be used for generating soft decision inputs in conjunction with error correction techniques. Observe that the schematic of Figure 2.19 is more explicit, than that of Figure 2.18, since for the specific case of BPSK we have $\mathcal{M} = 2$. This yields two equaliser subnets, which correspond to the transmission of a logical one as well as a logical zero, respectively.

The computational complexity of the $\mathcal{M}$-ary RBF equaliser is dependent on the order $\mathcal{M}$ of the modulation scheme, since the number of sub-RBF hidden nodes is equivalent to $n_s^i =$

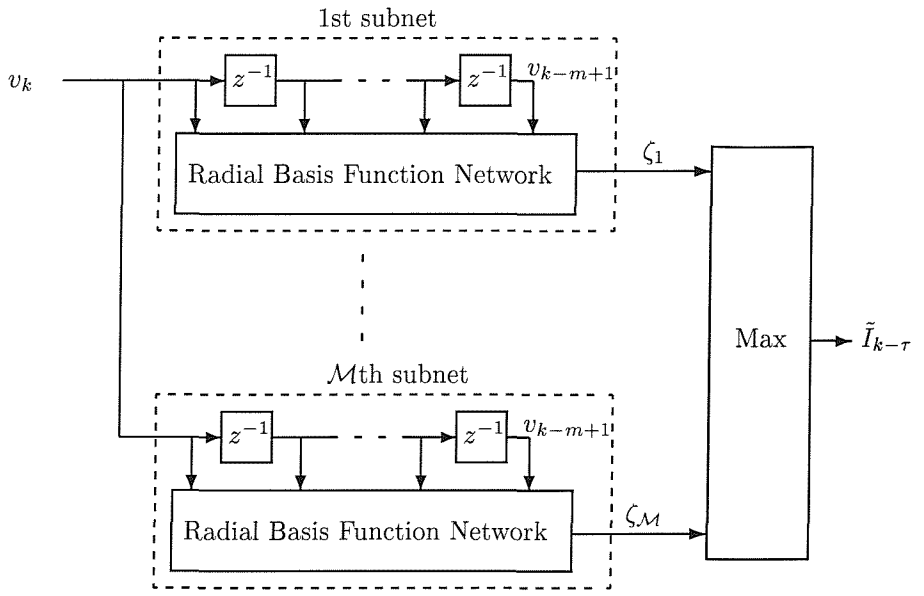


Figure 2.19: Radial Basis Function Equaliser for $\mathcal{M}$-level modems

$\mathcal{M}^{L+m}/\mathcal{M}$. Thus, its application is typically restricted to low-order $\mathcal{M}$-ary modulation schemes. The computational complexity of each subnet of the $\mathcal{M}$-ary RBF equaliser is similar to that in Table 2.4, taking into account the reduced number of hidden nodes, namely $n_s^i = n_s/\mathcal{M}$. Thus, the overall computational complexity of the $\mathcal{M}$-ary RBF equaliser described by Equation 2.85 and 2.86 is given in Table 2.5.

Number of subractions and additions	$2n_s m - \mathcal{M}$
Number of multiplications	$n_s(m+1)$
Number of divisions	n_s
Number of exp()	n_s
Number of max operation	1

Table 2.5: Computational complexity of an mth-order RBF network equaliser per equalised output sample for $\mathcal{M}$-ary modulation based on Equation 2.85 and 2.86. The total number of hidden nodes of the RBF equaliser is n_s.

2.9.3 Adaptive RBF Equalisation

The knowledge of the noise-free channel outputs is essential for the determination of the decision function associated with Equation 2.85. The channel state estimation – where the channel states were defined in Section 2.2, in particular in the context of Equation 2.8 – requires the knowledge of the CIR, but this often may not be available. Thus the channel state has to be 'learned' during the actual data transmission or inferred during the equaliser

training period, when the transmitted symbols are known to the receiver. This can be achieved typically in two ways [26]:

- By invoking CIR estimation methods [26, 25, 62]
- By employing so-called clustering algorithms [8] as described in Section 2.8

These methods will be highlighted by the following two Sections.

2.9.4 Channel Estimation Using a Training Sequence

According to our approach in this section, the channel model is first estimated using algorithms such as the Least Mean Square (LMS) algorithm [59]. With the knowledge of the CIR, the channel state can then be calculated. Let us define the CIR estimate associated with the model of Figure 2.1 as:

$$\hat{\mathbf{f}}_k = \begin{bmatrix} \hat{f}_{0,k} & \dots & \hat{f}_{L,k} \end{bmatrix}^T, \tag{2.87}$$

and introduce the $(L+1)$-element channel estimator input vector

$$\mathbf{I}_{f,k} = \begin{bmatrix} I_k & \dots & I_{k-L} \end{bmatrix}^T, \tag{2.88}$$

where $\{I_k\}$ is the transmitted channel input sequence, which is known during the training period. Then the error between the actual channel output v_k and the estimated channel output derived using the estimated CIR $\hat{\mathbf{f}}_{k-1}$ can be expressed as:

$$\varepsilon_k = v_k - \hat{\mathbf{f}}_{k-1}^T \mathbf{I}_{f,k}. \tag{2.89}$$

The CIR estimate can then be updated following the steepest descent philosophy of Equation A.3 as follows:

$$\hat{\mathbf{f}}_k = \hat{\mathbf{f}}_{k-1} + \mu_f \varepsilon_k \mathbf{I}_{f,k}^*, \tag{2.90}$$

where μ_f is the step-size defined by the channel estimator learning rule. Note however that the LMS channel estimation technique based on the channel model described in Figure 2.1 will fail, if the channel is non-linear in its nature.

During data transmission after learning, a decision-directed and delayed version of Equation 2.89 and Equation 2.90 is used, which is formulated as:

$$\begin{aligned} \varepsilon_{k-\tau} &= v_{k-\tau} - \hat{\mathbf{f}}_{k-\tau-1} \tilde{\mathbf{I}}_{f,k-\tau} \\ \hat{\mathbf{f}}_{k-\tau} &= \hat{\mathbf{f}}_{k-\tau-1} + \mu_f \varepsilon_{k-\tau} \tilde{\mathbf{I}}_{f,k-\tau}^*, \end{aligned} \tag{2.91}$$

that can be employed to track time-varying channels, where

$$\tilde{\mathbf{I}}_{f,k-\tau} = \begin{bmatrix} \tilde{I}_{k-\tau} & \dots & \tilde{I}_{k-\tau-L} \end{bmatrix}^T \tag{2.92}$$

is the channel estimator input vector associated with the CIR vector $\mathbf{f}_{k-\tau}$. Note that during data transmission, $\{\tilde{I}_{k-\tau}\}$ is the delayed symbol, detected by the equaliser. At instant $k+1$, the delayed CIR estimate $\hat{\mathbf{f}}_{k-\tau}$ is used to track the time-varying channel as though it were the most recent estimate $\hat{\mathbf{f}}_k$. The current channel model $\hat{\mathbf{f}}_{k+1}$ might have changed considerably. This tracking error owing to the inherent decision delays will degrade the performance of the channel estimator. As it will be demonstrated in Figure 2.22 at a later stage, increasing the decision delay τ first introduced in the context of Equation 2.82 improves the performance of the equaliser for a stationary channel. By contrast, this will degrade the performance of the channel estimator for a nonstationary channel environment. Thus we need to achieve a reasonable compromise and the selection of the decision delay parameter τ yielding satisfactory equaliser performance will depend on how rapidly the CIR varies.

The computational complexity of the LMS channel estimator is characterised in Table 2.6 based on Equation 2.89, which requires $L+1$ multiplication and $L+1$ addition/subtraction operations, and Equation 2.90 which involves $L+2$ multiplication and $L+1$ addition operations. On the basis of the estimated CIR $\hat{\mathbf{f}}_k$ it is straightforward to compute the estimated noise-free channel outputs $\tilde{v}_k$ using convolution and therefore to generate the channel output states $\mathbf{r}_i$. Upon substituting Equation 2.7 into the noiseless version of Equation 2.11, the channel output state $\mathbf{r}_i$ can be computed from:

$$\mathbf{r}_i = \mathbf{F}\mathbf{s}_i \tag{2.93}$$

where the elements of the CIR matrix $\mathbf{F}$ are obtained from Equation 2.90. Equation 2.93 requires $m(m+L)$ multiplication and $m(m+L-1)$ addition operations. Therefore, an additional computational load is encountered in converting the CIR estimate $\hat{\mathbf{f}}_k$ into the vector $\mathbf{r}_i$ of channel output states and this has to be added to the computational complexity calculation of the CIR estimator given in Table 2.6, in order to quantify to give the total complexity for this channel state learning method, as shown in Table 2.7.

$2(L+1)+1$ multiplications
$2(L+1)$ additions or subtractions

Table 2.6: Computational complexity of the LMS CIR estimator for a channel having $L+1$ symbol-spaced taps per estimated CIR based on Equation 2.89 and Equation 2.90.

The CIR estimate can also be updated using the Recursive Least Square (RLS) algorithm [59], which has a better convergence performance compared to the LMS algorithm in most cases. However, the RLS algorithm exhibits a higher computational complexity than the LMS algorithm. For dispersive mobile radio channels the adaptive algorithm is expected to

$m(m+L)+2(L+1)+1$ multiplications
$3L+m+1$ additions or subtractions

Table 2.7: Computational complexity of the m-dimensional channel output state learning algorithm using the LMS CIR estimator for a channel having $L+1$ symbol-spaced taps per channel output state based on Equation 2.89, Equation 2.90 and Equation 2.93.

continuously operate during both the training and transmission periods in highly nonstationary environments, consequently its numerical stability is vital. Many versions of the fast RLS algorithm may not be suitable for this purpose. The CIR can also be estimated using the so-called least sum of square errors (LSSE) algorithm [63]. This algorithm is similar to the CIR estimator used in the GSM system [64] and those in [65, 66], and it exhibits a low computational complexity.

2.9.5 Channel Output State Estimation using Clustering Algorithms

Apart from training sequences, the channel states can also be estimated invoking the clustering algorithms described in Section 2.8. The computational procedures of the so-called supervised K-means clustering algorithm during the equaliser training period can be summarised as follows [8]:

$$\begin{aligned} &\text{if } \mathbf{I}_k = \mathbf{s}_i, \text{then} \\ &\qquad \mathbf{c}_{i,k} = \mathbf{c}_{i,k-1} + \mu_c \cdot (\mathbf{v}_k - \mathbf{c}_{i,k-1}), \\ &\text{otherwise} \\ &\qquad \mathbf{c}_{i,k} = \mathbf{c}_{i,k-1}, \end{aligned} \tag{2.94}$$

where μ_c is the associated learning rate, $\mathbf{s}_i, 1 \leq i \leq n_s = \mathcal{M}^{L+m}$ is the ith channel input sequence and $\mathbf{I}_k = \begin{bmatrix} I_k & \dots & I_{k-m+1-L} \end{bmatrix}^T$ is an $(m+L)$-element transmitted symbol vector, which is known during the training phase. Explicitly, according to Equation 2.94 the clustering algorithm takes into account the most recently received m-element vector $\mathbf{v}_k$ in adapting the ith RBF centre $\mathbf{c}_{i,k}$, if the current $(L+m)$-element channel input vector $\mathbf{I}_k$ is given by the specific $(L+m)$-element vector $\mathbf{s}_i$. Initially, the RBF centres are all set to 0, i.e $c_{i,0} = 0, i = 1 \leq i \leq n_s = \mathcal{M}^{L+m}$. Equation 2.94 dictates that the previous centroid $\mathbf{c}_{i,k-1}$ has to be updated according to the 'distance' $(\mathbf{v}_k - \mathbf{c}_{i,k})$ between itself and the most recent $(L+m)$-element received vector $\mathbf{v}_k$ after scaling it by the learning rate μ_c. Otherwise the ith centre is not updated based on the information of the current received vector $\mathbf{v}_k$. Referring back to Section 2.8, the membership indicator defined by Equation 2.80 differs from that of the supervised version of the K-means clustering algorithm described

by Equation 2.94. Explicitly, this modified membership indicator is defined as:

$$M_i(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{I}_k = \mathbf{s}_i \\ 0 & \text{otherwise.} \end{cases} \tag{2.95}$$

For time-varying channels we have to track the time-varying channel states during transmission after the training period. For tracking the channel-induced channel state variations, the following decision-directed clustering algorithm can be used to adjust the RBF centres, in order to take into account the current network input vector $\mathbf{v}_k$ in the updating of the centres as follows [8]:

$$\begin{aligned} &\text{if } \tilde{\mathbf{I}}_{k-\tau} = \mathbf{s}_i, \text{then} \\ &\qquad \mathbf{c}_{i,k} = \mathbf{c}_{i,k-1} + \mu_c \cdot (\mathbf{v}_{k-\tau} - \mathbf{c}_{i,k-1}), \\ &\text{otherwise} \\ &\qquad \mathbf{c}_{i,k} = \mathbf{c}_{i,k-1}, \end{aligned} \tag{2.96}$$

where $\tilde{\mathbf{I}}_{k-\tau} = \begin{bmatrix} \tilde{I}_{k-\tau} & \dots & \tilde{I}_{k-\tau-m+1-L} \end{bmatrix}^T$ represents the $(L+m)$ equalised demodulated symbols after decision and a delay of τ. Note that whilst in Equation 2.94 the transmitted vector $\mathbf{I}_k$ was used, in Equation 2.96 the vector $\tilde{\mathbf{I}}_{k-\tau}$ at the output of the decision device is used. The computational complexity of the clustering algorithm obeying Equation 2.94 is given in Table 2.8.

Local operation: Find i, $i = 1, \dots, n_s$, for which $\mathbf{I}_k = \mathbf{s}_i$.
m multiplication
$2m$ additions or subtrations

Table 2.8: Computational complexity of the clustering algorithm specified by Equation 2.94 per channel output state for a RBF network having m inputs and n_s hidden nodes.

As we mentioned previously, all the RBF centres were initially set to 0. However, the centres can be initialised to the corresponding noisy channel states, in order to improve the convergence rate, since there is a higher probability that the actual channel states are nearer to the noisy channel states, than to $\mathbf{c}_{i,0} = 0, i = 1, \dots, n_s = \mathcal{M}^{L+m}$. Thus, the algorithm described by Equation 2.94 can be adapted as follows:

$$\begin{aligned} &\textit{if } \mathbf{I}_k = \mathbf{s}_i, \text{ and } \mathbf{c}_{i,k} \text{ has not been initialised } \textit{then} \\ &\qquad \mathbf{c}_{i,k} = \mathbf{v}_k, \\ &\textit{else if } \mathbf{I}_k = \mathbf{s}_i, \text{ and } \mathbf{c}_{i,k} \text{ has been initialised } \textit{then} \\ &\qquad \mathbf{c}_{i,k} = \mathbf{c}_{i,k-1} + \mu_c \cdot (\mathbf{v}_k - \mathbf{c}_{i,k-1}). \end{aligned} \tag{2.97}$$

The achievable improvement of the convergence performance in conjunction with this algorithm will be demonstrated by our simulation results in Section 2.12.

2.9.6 Other Adaptive RBF Parameters

In the previous subsection, clustering algorithms were used for training the RBF centres. Similar procedures can be employed also for training the RBF weights as it will be outlined below. Explicitly, if our previous assumption of equiprobable symbols is violated, we have to adjust the RBF weights in order to learn the corresponding scaling factors of the conditional probability density functions in Equation 2.18 during the training period. The adaptation of the RBF weights can be achieved pursuing the approach of Chen, Mulgrew and Grant using the following supervised LMS algorithm [8]:

$$\begin{aligned} \epsilon_k &= I_{k-\tau} - f_{RBF}(\mathbf{v}_k) \\ \mathbf{w}_{i,k} &= \mathbf{w}_{i,k-1} + \mu_w \epsilon_k \varphi(\|\mathbf{v}_k - \mathbf{c}_i\|), \end{aligned} \tag{2.98}$$

where μ_w is the learning rate for the RBF weights. Explicitly, the error $\epsilon_k = I_{k-\tau} - f_{RBF}(\mathbf{v}_k)$ between the $(L+m)$-element transmitted symbol vector $\mathbf{I}_{k-\tau}$ and the RBF's output is scaled by the RBF learning rate μ_w and this product is then used to weight $\varphi(\|\mathbf{v}_k - \mathbf{c}_i\|)$, in order to update the previous RBF weight $\mathbf{w}_{i,k-1}$, where $\varphi(\|\mathbf{v}_k - \mathbf{c}_i\|)$ is the RBF evaluated at the Euclidean norm $\|\mathbf{v}_k - \mathbf{c}_i\|$ characteristic of the 'distance' between the centroids $\mathbf{c}_i, i = 1, \ldots, n_s = \mathcal{M}^{L+m}$, and the $(L+m)$-element received vector $\mathbf{v}_k$.

Furthermore, if the exact number of RBF centres is not known precisely or if there is a deliberate attempt to use a reduced set of centres to reduce the computational complexity – as it will be described in Section 2.9.7 – it may be prudent to train the weights using the LMS algorithm of Equation 2.98, in order to make best use of the actual centres that have been provided [28]. Similarly, in noisy environments, where clustering techniques may only provide fairly crude estimates of the centres, training the RBF weights will make best use of the trained centres [28]. Another method of training the RBF weights is demonstrated in Chapter 5 where the information of the coded symbols, generated by the channel decoder is used to adapt the RBF weights.

2.9.7 Reducing the Complexity of the RBF Equaliser

In an effort to reduce the RBF equaliser's complexity, Chng *et al.* [67] proposed finding a RBF centre subset model in order to approximate the Bayesian decision function's response given in Equation 2.18 for the current $(L+m)$-element input vector $\mathbf{v}_k$. This implied using only the centres which are near, in Euclidean sense, to the current input vector $\mathbf{v}_k$ for the subset model. The rationale of this approach is based on the assumption that the contribution of the RBF centres to the decision function is inversely related to their distance from the input vector, as we can observe from Equation 2.18. The decision function response using only the centres within a distance of Δ from $\mathbf{v}_k$ is very similar to the full Bayesian

RBF response, if the distance Δ is sufficiently large. Chng's results show that a distance of $\Delta = 4\sigma_\eta$ is sufficient and can reduce the number of centres required for the subset model to as small as 5-10% of the full model. Chng's paper [67] also provides a fast algorithm for identifying the specific centres, which are within a distance of Δ from the input vector $\mathbf{v}_k$ for the subset model .

Patra and Mulgrew [68] investigated the computational complexity aspects of RBF equalizers. They proposed an RBF equaliser using scalar centres, which can implement the Bayesian decision function of Equation 2.18, while allowing a lower computational complexity compared to previously reported RBF equalizers. This issue will be detailed in the next section, hence suffice to say here that the scalar centre c_{il} is the $(l+1)$th component of the RBF centroid vector $\mathbf{c}_i = [c_{i0} \ \ldots \ c_{il} \ \ldots \ c_{i(m-1)}]^T$, associated with the mth order equaliser, where c_{il} assumes the possible values of the noise-free channel output $\tilde{v}_k$ in order to realise the optimal Bayesian decision function of Equation 2.18. For binary transmission, there are 2^{L+1} possible noise-free channel output states, which correspond to each of the m elements of the equaliser's input vector $\tilde{\mathbf{v}}$, described in Equation 2.6, where $L+1$ is the length of the CIR. The mapping between the scalar centres and the scalar channel states will be expounded in more detail in Section 2.10.

The RBF equaliser described by the scalar centres can efficiently employ subset centre selection for computing the decision function of Equation 2.18, resulting in a substantial reduction in computational complexity. The algorithm proposed for subset centre selection by Patra [68] is more attractive compared to that suggested by Chng [67] *et al.* , since it is more efficient in terms of selecting a subset of the total set of centres in the one-dimensional space. This is because we only need to select a subset of centres from a total of 2^{L+1} possible scalar centres for Patra's method [68] compared to a total of $n_s = 2^{m+L}$ possible vector centres for Chng's method [67].

Another method of selecting a subset of significant RBF centres is to make use of past detected symbols. This idea, which incoporated decision feedback into the RBF network was proposed by Chen *et al.* [25, 26]. Section 2.11 will present this approach in more detail, together with our simulation results in Section 2.12.

In an effort to further reduce the complexity we invoke an approach often used in turbo codes [69]. for complexity reduction. Specifically, we proposed generating the output of the RBF equaliser in logarithmic form by invoking the Jacobian logarithm [13, 14], in order to avoid the computation of exponentials and to reduce the number of multiplictions performed. We refer to this equaliser as the Jacobian RBF equaliser, which will be introduced in Section 4.2.

2.10 Scalar Noise-free Channel Output States

In this section, we will describe in detail the scalar noise-free channel output states and relate them to the m-element noise-free channel output state vector $\tilde{\mathbf{v}}_\mathbf{i}$ and to the scalar RBF centres c_{il} that we have mentioned in Section 2.9.7. After defining the scalar noise-free channel output state, we will expound on how it is used to reduce the complexity of the RBF equaliser.

Referring back to Equation 2.6 and Equation 2.3, the lth element $\tilde{v}_{k-l}, l = 0, 1, \ldots, m-1$, of the $(L+1)$-element noise-free channel output vector $\tilde{\mathbf{v}}_k$ corresponds to the so-called block-convolution of a sequence of $L+1$ transmitted symbols and the $L+1$ CIR taps. In other words, the number of transmitted symbols contributing to the value of $\tilde{v}_{k-l}$ is $L+1$ and we represent these transmitted symbols by an $L+1$ element vector $\mathbf{I}_{f,k}$, as described by Equation 2.88. Let us now introduce the concept of scalar states using the channel-state example of Table 2.1, where the scalar channel output states are $r_1 = -1.5$, $r_2 = -0.5$, $r_3 = 0.5$ and $r_4 = 1.5$, while the number of scalar channel states is $n_{s,f} = 2^{L+1} = 4$ $(L = 1)$. Thus, the vector channel output states can be expressed with the aid of the scalar states forming the vector as $\mathbf{r}_1 = [r_1\ r_1]^T$, $\mathbf{r}_2 = [r_1\ r_2]^T, \ldots$, etc. More explicitly at every instant $-\infty < k < \infty$ the noiseless scalar channel output is given by the corresponding convolution of the input bits and the CIR. In general, the number of different possible combinations of the $(L+1)$-element transmitted symbol sequence in $\mathbf{I}_{f,k}$ is $n_{s,f} = 2^{L+1}$ for a binary modulation scheme. We represent these transmitted symbol combinations equivalently as a channel input state $\mathbf{s}_{scalar,i}$, where $i = 1, 2, \ldots, n_{s,f} = 2^{L+1}$. After convolution with the CIR, each of these channel input states $\mathbf{s}_{scalar,i}$ generates a scalar channel output state $r_i, i = 1, 2, \ldots, n_{s,f} = 2^{L+1}$. Thus, as we have seen with reference to Table 2.1 the noise-free channel output $\tilde{v}_k$ can take up any of the $n_{s,f} = 2^{L+1}$ scalar channel output states r_i, depending on $\mathbf{I}_{f,k}$, which is summarised as:

$$\tilde{v}_k = r_i \qquad \text{if } \mathbf{I}_{f,k} = \mathbf{s}_{scalar,i} \qquad i = 1, \ldots, n_{s,f}, \qquad -\infty < k < \infty. \tag{2.99}$$

Similarly to our introductory example, the scalar channel output states $r_i, i = 1, 2, \ldots, n_{s,f} = 2^{L+1}$, can be suitably combined to form the vector channel output states $\mathbf{r}_j, j = 1, 2, \ldots, n_s = 2^{m+L}$, seen in Equation 2.7.

In order to realise the optimal Bayesian decision function of Equation 2.18 , the scalar centre c_{il} – which is the $(l+1)$th component of the vector centre $\mathbf{c}_i$, where $i = 1, 2, \ldots, n_s = 2^{m+L}$ and $l = 0, 1, \ldots, m-1$, as mentioned in Section 2.9.7 – has to assume the value of these scalar channel output states r_i. The scalar centres c_{il} can be obtained from a lookup table that provides the mapping $Q : R \rightarrow C$, where $R = \{r_1, \ldots, r_i, \ldots r_{n_{s,f}}\}$ and $C = \{c_{00}, \ldots, c_{il}, \ldots, c_{n_s(m-1)}\}$. Using again the example of Table 2.1 and letting

$\mathbf{c}_i = \mathbf{r}_i, i = 1, \ldots, n_s$, the scalar centres correspond to the scalar channel output states as follows : $c_{00} = r_1, c_{01} = r_1, c_{10} = r_1, c_{11} = r_2, \ldots$, etc.

A scalar channel output state r_i is just the conditional mean of the noisy observation v_k given by $\mathbf{I}_{f,k} = \mathbf{s}_{scalar,i}$, and a clustering procedure can be used to update the scalar channel states as follows [25]:

$$\begin{aligned} &\text{if } \mathbf{I}_{f,k} = \mathbf{s}_{scalar,i}, \text{then} \\ &\qquad r_{i,k} = r_{i,k-1} + \mu_r \cdot (v_k - r_{i,k-1}), \\ &\text{otherwise} \\ &\qquad r_{i,k} = r_{i,k-1}, \end{aligned} \tag{2.100}$$

where μ_r is the associated learning rate of the scalar channel states. For time-varying channels, it is necessary to continuously update r_i during data transmission. This can be achieved using the following decision-directed version of Equation 2.100:

$$\begin{aligned} &\text{if } \tilde{\mathbf{I}}_{f,k-\tau} = \mathbf{s}_{scalar,i}, \text{then} \\ &\qquad r_{i,k} = r_{i,k-1} + \mu_r \cdot (v_{k-\tau} - r_{i,k-1}) \\ &\text{otherwise} \\ &\qquad r_{i,k} = r_{i,k-1}. \end{aligned} \tag{2.101}$$

The computational complexity of the clustering algorithm in the context of the scalar channel states is given in Table 2.9. Note that the computational load of the clustering scheme for the scalar channel states is lower than that for the vector channel states, which becomes explicit by comparing Table 2.9 and Table 2.8 of Section 2.9.5, since $n_{s,f} < n_s$. However, some additional processing is required, in order to expand the scalar states into the vector states. This is not costly, especially, if the expansion can be done via a lookup table.

Local operation: Find i, $i = 1, \ldots, n_{s,f}$, for which $\mathbf{I}_{f,k} = \mathbf{s}_{scalar,i}$.
1 multiplications
2 additions or subtractions

Table 2.9: Computational complexity of the clustering algorithm per scalar channel output state for $n_{s,f}$ number of scalar channel output states based on Equation 2.100.

As mentioned in Section 2.9.3, the channel states can be learnt by invoking channel estimation methods. Section 2.9.4 described a channel estimation method using the LMS algorithm. Since the number of channel taps $L + 1$ is lower than that of the scalar channel states $n_{s,f} = \mathcal{M}^{L+1}$, it becomes explicit that an adaptive scheme based on a channel

estimator requires a shorter training period than the clustering approach. Thus the former is better suited for time-variant channels. However, the clustering scheme does not assume the linear channel model described by Equation 2.1 and it is immune to nonlinear distortion. When significant nonlinear distortion is inflicted for example by the system's power amplifier, the estimated channel states based on a linear model will deviate from the true states, causing a performance loss. The clustering approach does not suffer form this problem and it always converges to the set of true channel output states, regardless of whether the channel is linear or nonlinear.

The scalar channel state clustering scheme provides faster convergence compared to the vector channel state clustering scheme, since the convergence performance depends on the number of clusters or channel states and the number of scalar channel states is less than the number of vector channel states. This will be demonstrated in Section 2.12, which will provide simulation results in order to characterize the performance of the scalar channel state clustering scheme.

Upon extending the scalar channel state concept to multilevel modems, we note that the number of channel states $n_{s,f} = \mathcal{M}^{L+1}$ grows exponentially with the number $\mathcal{M}$ of symbol constellation points used in the modulation scheme. Thus, the convergence rate is dependent on the type of modulation scheme used.

2.11 Decision Feedback Assisted Radial Basis Function Network Equaliser [25, 26, 62]

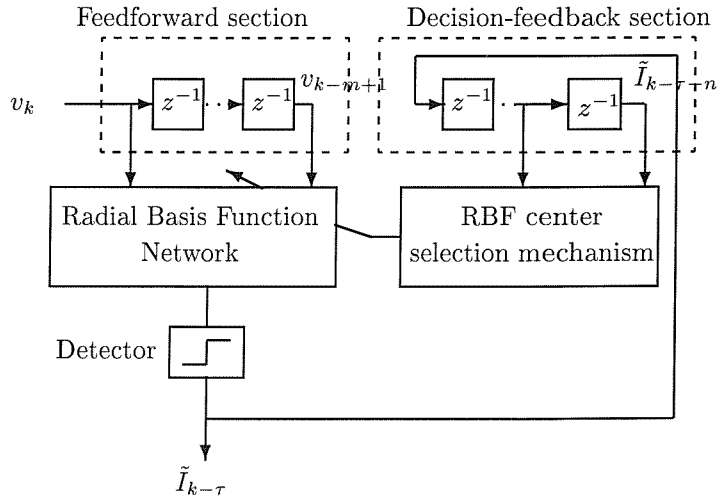


Figure 2.20: Radial basis function equaliser with decision feedback

In their seminal contribution Chen, Mulgrew and McLaughlin [25, 62, 26] introduced decision feedback into the RBF equaliser in order to reduce its computational complexity, as

mentioned earlier in Section 2.9.7. Figure 2.20 illustrates this design for a binary modulation scheme. Observe in the figure that in contrast to conventional DFEs, where the output of the feedback section is subtracted from that of the feedforward section, here the feedback section is employed to assist in the operation of the feedforward section, as it will become explicit later in this section. The structure of a decision feedback RBF equaliser is specified by the equaliser's decision delay τ, the feedforward order m and the feedback order n.

The n-symbol long binary feedback vector $\tilde{\mathbf{I}}_{feedback,k-\tau} = \left[\begin{array}{ccc} \tilde{I}_{k-\tau-1} & \ldots & \tilde{I}_{k-\tau-n} \end{array} \right]^T$ is associated with $n_f = 2^n$ states. We denote the set of $n_f = 2^n$ different feedback sequences by $\mathbf{s}_{f,j}$, $1 \leq j \leq n_f = 2^n$. The binary subset $V^+_{m,\tau}$ and $V^-_{m,\tau}$ of the channel states defined in Equation 2.8 can be further partitioned into n_f subsets, $V^+_{m,\tau,j}$ and $V^-_{m,\tau,j}$, according to the $n_f = 2^n$ possible feedback states such that the union of the $n_f = 2^n$ number of feedback states associated with the two legitimate binary transmitted symbols can be formulated as:

$$\begin{aligned} V^+_{m,\tau} &= \bigcup_{1 \leq j \leq n_f} V^+_{m,\tau,j} \\ V^-_{m,\tau} &= \bigcup_{1 \leq j \leq n_f} V^-_{m,\tau,j}, \end{aligned} \tag{2.102}$$

where $V^{\pm}_{m,\tau}$ is the set of possible $\tilde{\mathbf{v}}_k$ values associated with the delayed transmitted symbol $I_{k-\tau} = \pm 1$ and the feedback symbol sequence $\tilde{\mathbf{I}}_{feedback,k-\tau} = \mathbf{s}_{f,j}$ yields the following subsets:

$$\begin{aligned} V^+_{m,\tau,j} &= \{\tilde{\mathbf{v}}_k | I_{k-\tau} = +1 \cap \tilde{\mathbf{I}}_{feedback,k-\tau} = \mathbf{s}_{f,j}\}, \\ V^-_{m,\tau,j} &= \{\tilde{\mathbf{v}}_k | I_{k-\tau} = -1 \cap \tilde{\mathbf{I}}_{feedback,k-\tau} = \mathbf{s}_{f,j}\}, \\ & 1 \leq j \leq n_f. \end{aligned} \tag{2.103}$$

Thus the role of the feedback symbol vector $\tilde{\mathbf{I}}_{feedback,k-\tau}$ in the decision feedback structure is to select a subset of centres for a particular decision. The proportion of channel states in the sets $V^+_{m,\tau,j}$ and $V^-_{m,\tau,j}$ is $n^+_{s,j} = n^+_s/n_f$ and $n^-_{s,j} = n^-_s/n_f$, respectively. The total number of channel states associated with the feedback state $\mathbf{s}_{f,j}$ is given by $n_{s,j} = n^+_{s,j} + n^-_{s,j}$. Given the feedback vector $\tilde{\mathbf{I}}_{feedback,k-\tau} = \mathbf{s}_{f,j}$, the Bayesian decision function of Equation 2.18 can be rewritten with a reduced number of noiseless channel states as:

$$\begin{aligned} f_{Bayes}(\mathbf{v}_k | \tilde{\mathbf{I}}_{feedback,k-\tau} = \mathbf{s}_{f,j}) &= \sum_{i=1}^{n^+_{s,j}} p^+_{j,i} (2\pi\sigma^2_\eta)^{-m/2} exp(-\|\mathbf{v}_k - \mathbf{r}^+_{j,i}\|^2 / 2\sigma^2_\eta) \\ &- \sum_{l=1}^{n^-_{s,j}} p^-_{j,l} (2\pi\sigma^2_\eta)^{-m/2} exp(-\|\mathbf{v}_k - \mathbf{r}^-_{j,l}\|^2 / 2\sigma^2_\eta), \\ & j = 1, \ldots, n_f, \end{aligned} \tag{2.104}$$

where $\mathbf{r}^+_{j,i}$ and $\mathbf{r}^-_{j,i}$, $i = 1, \ldots, n^{\pm}_{s,j}, j = 1, \ldots, n_f$ are the ith noiseless channel states, when the feedback vector $\tilde{\mathbf{I}}_{feedback,k-\tau}$ is $\mathbf{s}_{f,j}$, while the superscript $^+$ and $^-$ correspond to the

transmitted symbols of $I_{k-\tau} = +1$ and $I_{k-\tau} = -1$, respectively. Explicitly, $\mathbf{r}_{j,i}^{+} \in V_{m,\tau,j}^{+}$, $\mathbf{r}_{j,l}^{-} \in V_{m,\tau,j}^{-}$, while $p_{j,i}^{+}$ and $p_{j,i}^{-}$ are the *a priori* probability of occurance for each state $\mathbf{r}_{j,i}^{+}$ and $\mathbf{r}_{j,i}^{-}$, respectively. The minimum error probability decision is thus formulated as:

$$\tilde{I}_{k-\tau} = sgn(f_{Bayes}(\mathbf{v}_k|\tilde{\mathbf{I}}_{feedback,k-\tau} = \mathbf{s}_{f,j})). \tag{2.105}$$

The relationship between the RBF network described in Equation 2.81 and the Bayesian DFE decision function expressed in Equation 2.104 can now be given explicitly. The weights w_i in Equation 2.81 correspond to the scaling factors of the conditional probability density function given by $\pm p_{j,i}^{\pm}(2\pi\sigma_\eta^2)^{-m/2}$ in Equation 2.104. This was mentioned before in the context of Equation 2.83. The RBF centres $\mathbf{c}_i$ in Equation 2.81 correspond to the noise-free channel output vectors $\mathbf{r}_{j,i}^{+}$ and $\mathbf{r}_{j,i}^{-}$. That is, if the n-element feedback symbol sequence $\tilde{\mathbf{I}}_{feedback,k-\tau}$ obtained is equivalent to $\mathbf{s}_{f,j}$, we assigned the $n_{s,j}$ number of RBF centres $\mathbf{c}_i, i = 1, \ldots, n_{s,j}$, to the channel output vector $\mathbf{r}_{j,i}^{\pm}, i = 1, \ldots, n_{s,j}^{\pm}$. The decision feedback reduces the computational complexity of the RBF equaliser, since the number of RBF hidden nodes needed to realize the Bayesian equalisation solution of Equation 2.18 is reduced from $n_s = 2^{m+L}$ to $n_{s,j} = n_s/n_f = 2^{m+L}/2^n = 2^{m+L-n}$ with the knowledge of the feedback state value. However, when the equaliser makes an incorrect decision and this decision is fed back, the wrong subset of centres is selected and this will degrade the BER performance of the RBF DFE, as it will be demonstrated in Section 2.12.

Extending the decision feedback RBF equaliser to a multilevel modem scenario is straightforward by introducing sub-RBF networks for each possible decision variable based on the conditional probability density function, as it was described in Section 2.9.2. The conditional Bayesian decision variable of Equation 2.86 can be redefined for the Bayesian DFE as:

$$\begin{aligned} \zeta_i(k) &= P(\mathbf{v}_k|I_{k-\tau} = \mathcal{I}_i \cap \tilde{\mathbf{I}}_{feedback,k-\tau} = \mathbf{s}_{f,l}) \\ &= \sum_{l=1}^{n_{s,j}^i} p_{j,l}^i p(\mathbf{v}_k - \mathbf{r}_{j,l}^i), \\ &\qquad 1 \le i \le \mathcal{M}, \\ &\qquad 1 \le j \le n_f, \end{aligned} \tag{2.106}$$

where $\mathbf{r}_{j,l}^i$ is the lth noiseless channel state, $l = 1, \ldots, n_{s,j}^i$ when the feedback vector is given by $\tilde{\mathbf{I}}_{feedback,k-\tau} = \mathbf{s}_{f,j}$ and the transmitted symbol is $I_{k-\tau} = \mathcal{I}_i$, i.e., $\mathbf{r}_{j,l}^i \in V_{m,\tau,j}^i$.

The computational complexity of the decision feedback assisted RBF equaliser is given in Table 2.10 based on Equation 2.104, which is similar to that without decision feedback given in Table 2.5, except for the reduced number of hidden units $n_{s,j} \le \mathcal{M}^{L+m}$. We conclude that in general, the complexity increase of the RBF DFE is of the order of $\mathcal{M}^L$,

Determine the feedback state	
$2n_{s,j}m - \mathcal{M}$	subtraction and addition
$n_{s,j}(m+1)$	multiplication
$n_{s,j}$	division
$n_{s,j}$	exp()
1	max evaluation

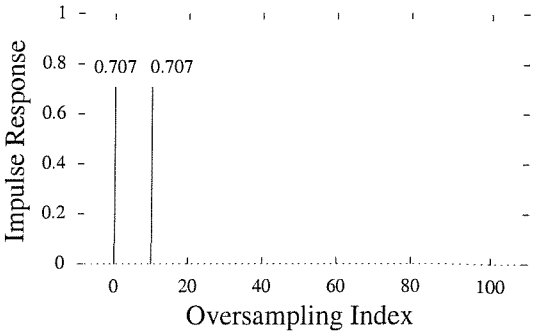
Table 2.10: Computational complexity of a decision feedback RBF network equaliser with m inputs and $n_{s,j}$ hidden units per equalised output sample based on Equation 2.104.

since as $n_{s,j} = \mathcal{M}^{m+L-n}$. Hence, its application is typically restricted to low-order $\mathcal{M}$-ary modulation schemes, such as 4-QAM and to channels, where the ISI does not extend beyond four or five symbol periods [28].

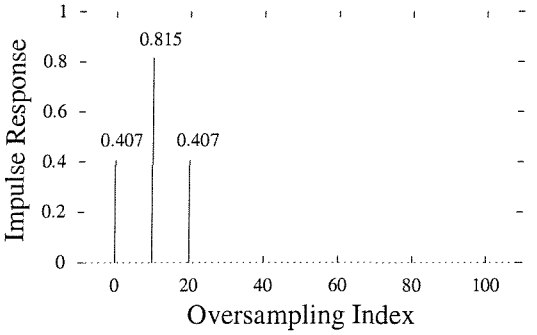
The oldest symbol that influences the decision at the kth signalling instant, which produces the detected symbol $\tilde{I}_{k-\tau}$, is $I_{k-m+1-L}$, as seen in Equation 2.5. The oldest feedback symbol is $\tilde{I}_{k-\tau-n}$. Therefore, it is sufficient to employ a feedback order of

$$n = \log_2 n_f = L + m - 1 - \tau, \tag{2.107}$$

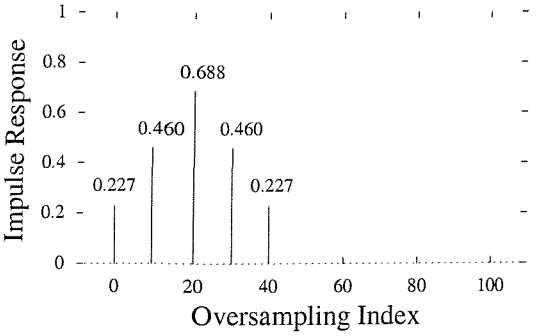
because this will enable us to influence decisions over the memory duration L of the concatenated channel and the feedforward RBF section m. Assuming hence $n = L+m-1-\tau$, Chen, Mulgrew and McLaughlin [26] mathematically proved that the Bayesian DFE of a feedforward order of $m = \tau + 1$ has the same conditional decision variables as those having a feedforward order of $m > \tau + 1$. The mathematical proof is given in Appendix B. Thus, given the delay τ – which was defined in the context of Figure 2.18 as the total decision delay of the feedforward shift-register of the RBF DFE – the feedforward order $m = \tau + 1$ is sufficient for attaining the best possible BER at the lowest possible complexity [26]. This is demonstrated in Figure 2.22 over the two-path channel environment of Figure 2.21(a). Substituting $m = \tau + 1$ in Equation 2.107 gives the corresponding feedback order of $n = \log_2 n_f = L$. Overall, the equaliser delay τ specifies the number of channel states n_s, required for computing the decision variables and thus determines the computational complexity encountered. A pragmatic rule is to set the equaliser's decision delay to $\tau = L$ [26]. However, note that increasing the decision delay τ and feedforward order m will improve the performance of the RBF equaliser, as demonstrated in Figure 2.23, at the expense of increasing the computational complexity exponentially, since the number of desired channel states $n_s = \mathcal{M}^{L+m}$ increases exponentially with m. Figure 2.24 shows the equaliser's BER performance versus its feedforward order m. The BER performance improves almost linearly with the feedforward order, before the curves reach their SNR-dependent residual BERs. The effect of increasing the feedforward order is more significant



(a) Two-path channel



(b) Three-path channel



(c) Five-path channel

Figure 2.21: Four discrete time channel impulse responses for an oversampling ratio of 10.

in BER-reduction terms at high E_b/N_0 values, as shown in Figure 2.24. For example, at an SNR of 12dB, an increase of the feedforward order from $m = 3$ to $m = 6$ improves the equaliser's BER performance by an order of magnitude, from 10^{-5} to 10^{-6}.

In the next subsection we shall further illustrate the concept of the feedback states and the redefined noiseless channel states using the same example as in Section 2.2.

2.11.1 Radial Basis Function Decision Feedback Equaliser Example

The channel impulse response used in this example was given by Equation 2.13, which is repeated here for convenience: $F(z) = 1 + 0.5z^{-1}$, implying that we have $L = 1$. We use the following equaliser parameters:

- Feedforward order of $m = 2$.
- Feedback order of $n = 1$.
- Decision delay of $\tau = 1$.

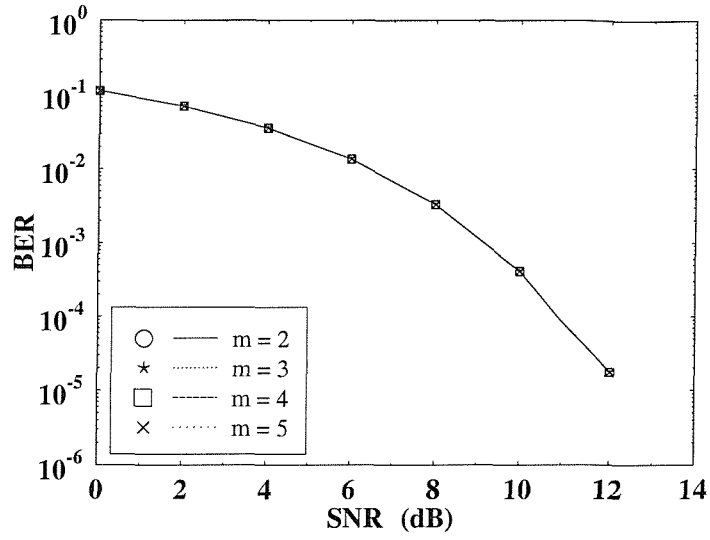


Figure 2.22: BER versus SNR performance of the RBF equaliser with correct decision feedback upon varying the feedforward order m over the dispersive two-path Gaussian channel of Figure 2.21(a). The equaliser decision delay τ was fixed to 1 symbol and the feedback order n was varied according to $n = L + m - 1 - \tau$, where $L + 1 = 2$ is the CIR length.

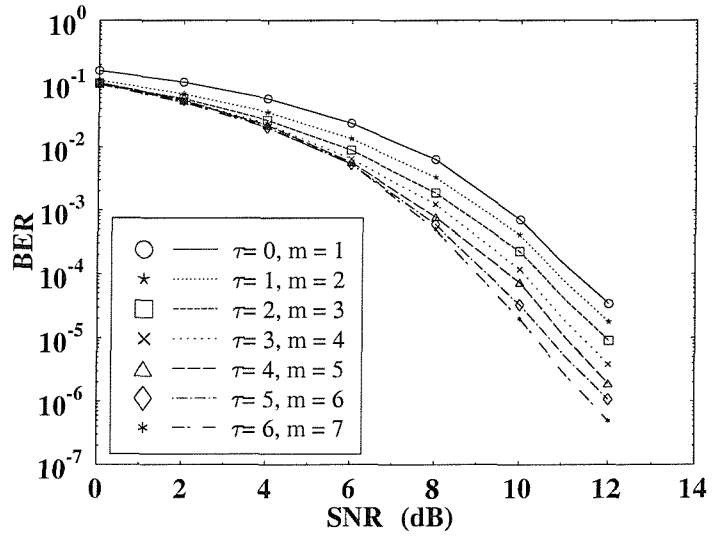


Figure 2.23: BER versus SNR performance of the RBF equaliser with correct decision feedback upon varying the decision delay τ over the dispersive two-path Gaussian channel of Figure 2.21(a). The equaliser feedforward order m was fixed to $m = \tau + 1$ and the feedback order n was varied according to $n = L + m - 1 - \tau$, where $L + 1 = 2$ is the CIR length. The equaliser's complexity increases exponentially with m, as seen in Table 2.10.

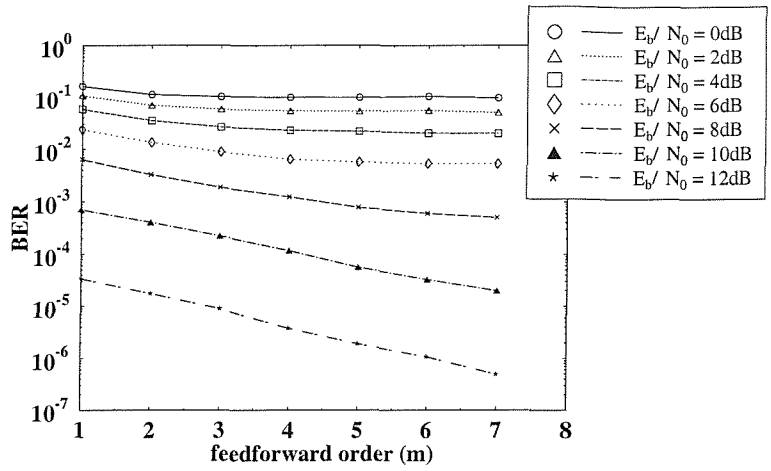


Figure 2.24: BER versus feedforward order m of the RBF equaliser with correct decision feedback for varying E_b/N_0 values over the dispersive two-path Gaussian channel of Figure 2.21(a). The equaliser decision delay τ was varied according to $\tau = m - 1$ and the equaliser complexity increases exponentially with m, as seen in Table 2.10.

Thus, in Figure 2.20 we have $\tilde{\mathbf{I}}_{feedback,k-\tau} = \left[\tilde{I}_{k-2}\right]$, since the feedforward section delays the received signal by two sampling interval durations. Furthermore, $\tilde{\mathbf{v}}_k = \begin{bmatrix} \tilde{v}_k & \tilde{v}_{k-1} \end{bmatrix}^T$ and the delayed transmitted symbol is I_{k-1}. The number of noise-free channel output states is $n_s = 2^{m+L} = 2^{2+1} = 8$ in Figure 2.25, where $n_s^+ = 4$ and $n_s^- = 4$, the number of feedback states is $n_f = 2^n = 2$, while the number of subset channel states associated with the $n_f = 2^n = 2$ feedback states is $n_{s,j}^+ = 2$ and $n_{s,j}^- = 2$. We denote the feedback states $\mathbf{s}_{f,j}$, where $j = 1, 2$ as $\mathbf{s}_{f,1} = [-1]$ and $\mathbf{s}_{f,2} = [+1]$. Assuming that the feedback symbols are correct, all the combinations of the transmitted binary symbols I_k, I_{k-1} and I_{k-2} as well as the noiseless channel outputs $\tilde{v}_k$, $\tilde{v}_{k-1}$, the noiseless channel output states $\mathbf{r}_{i,j}^+$ and $\mathbf{r}_{i,j}^-$ and the feedback states $\mathbf{s}_{f,j}$ are listed in Table 2.11. Again, Figure 2.25 shows the noiseless channel output states observed by an equaliser having a feedforward order of $m = 2$ and decision delay of $\tau = 1$. Figure 2.26(a) and Figure 2.26(b) show the noiseless channel output states of the RBF DFE using the parameters given above, when the feedback state $\mathbf{s}_{f,j}$ is equivalent to -1 and +1, respectively, as stated in Table 2.11. Following the spirit of Figure 2.3 in partitioning the decision space, at this stage we have to decide, what the transmitted bit I_{k-2} was. This decision can be carried out by evaluating Equation 2.106 and identifying the symbol $\mathcal{I}_i$, $i = 1, \ldots, \mathcal{M}$ associated with the highest probability.

Note that the number of channel states required, in order to estimate the transmitted symbol $I_{k-\tau}$ is now reduced from $n_s = 2^{m+L}$ to $n_{s,j} = n_s/n_f = 2^{m+L}/2^n = 2^{m+L-n}$, if we invoke the feedback state $\mathbf{s}_{f,j}$ in order to assist in the RBF subset selection. Explicitly, in

I_k	I_{k-1}	I_{k-2}	$\tilde{v}_k$	$\tilde{v}_{k-1}$	$\mathbf{s}_{f,j}$	$\mathbf{r}_{j,i}$
-1	-1	-1	-1.5	-1.5	$\mathbf{s}_{f,1} = -1$	$\mathbf{r}_{1,1}^{-}$
+1	-1	-1	+0.5	-1.5	$\mathbf{s}_{f,1} = -1$	$\mathbf{r}_{2,1}^{-}$
-1	+1	-1	-0.5	+0.5	$\mathbf{s}_{f,1} = -1$	$\mathbf{r}_{1,1}^{+}$
+1	+1	-1	+1.5	+0.5	$\mathbf{s}_{f,1} = -1$	$\mathbf{r}_{2,1}^{+}$
-1	-1	+1	-1.5	-0.5	$\mathbf{s}_{f,2} = +1$	$\mathbf{r}_{1,2}^{-}$
+1	-1	+1	+0.5	-0.5	$\mathbf{s}_{f,2} = +1$	$\mathbf{r}_{2,2}^{-}$
-1	+1	+1	-0.5	+1.5	$\mathbf{s}_{f,2} = +1$	$\mathbf{r}_{1,2}^{+}$
+1	+1	+1	+1.5	+1.5	$\mathbf{s}_{f,2} = +1$	$\mathbf{r}_{2,2}^{+}$

Table 2.11: Transmitted signal I_k, I_{k-1}, I_{k-2}, noiseless channel output $\tilde{v}_k, \tilde{v}_{k-1}$, feedback channel states $\mathbf{s}_{f,j}$ and noiseless channel states $\mathbf{r}_{j,i}$ for the channel impulse response of $F(z) = 1 + 0.5z^{-1}$ and equaliser feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol. The coordinates $\tilde{v}_k$ and $\tilde{v}_{k-1}$ identify the points $\mathbf{r}_{j,i}$ in Figure 2.25 and 2.26. This table is the extension of Table 2.1, where the entries were rearranged appropriately, in order to separate the entries assosiated with $s_{f,1} = -1$ and $s_{f,2} = +1$.

the example given above the number of channel states is reduced from 8 to 4, given the information of the feedback symbols. The computational complexity reduction factor owing to decision feedback is actually higher than n_f, since a DFE typically requires a reduced feedforward order m with respect to that, which is required without decision feedback. This is justified by the following arguments. Increasing the number of feedforward taps m extends the dimensionality of the observation space. This is necessary, in order to be able to increase the Euclidean distance between the RBF centres and thus to decrease the probability of mis-classification. It is apparent that the minimum distance amongst the constellation points of the subsets $V_{m,\tau,j}^{+}$ and $V_{m,\tau,j}^{-}$ of Figure 2.26 for a particular feedback state $\mathbf{s}_{f,j}$, is larger than amongst the points of the full subsets $V_{m,\tau}^{+}$ and $V_{m,\tau}^{-}$ of Figure 2.25. Thus, with the introduction of decision feedback, the Euclidean distance between the centres is already increased and hence a smaller m is sufficient for maintaining a given equaliser performance. Again, the increased Euclidean distance can be observed by comparing the noiseless channel outputs $\tilde{v}_k$, $\tilde{v}_{k-1}$ in Figure 2.25 and those in Figure 2.26. The distance between a constellation point or state corresponding to the transmitted symbol $I_k = +1$ and the nearest point or state corresponding to the transmitted symbol $I_k = -1$ is increased, when the DFE scheme is used. Another important advantage of the decision feedback method is that the noiseless channel states $\mathbf{r}_{j,i}$ corresponding to different transmitted symbols are linearly separable, provided that the parameters of the RBF DFE are chosen to be $\tau = L$, $m = \tau + 1 = L + 1$ and $n = L + m - \tau - 1 = L$, which was proven mathematically by Chen, Mulgrew, Chng and Gibson [70] for a PAM modulation scheme. This proof can be

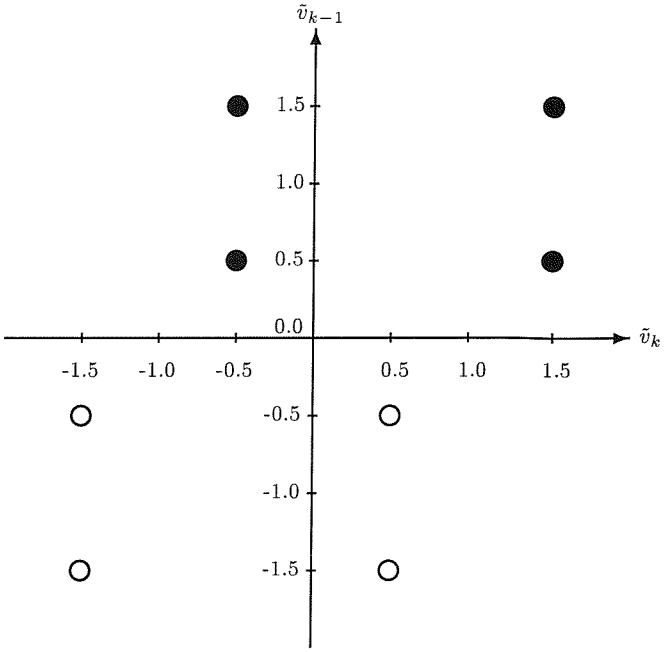


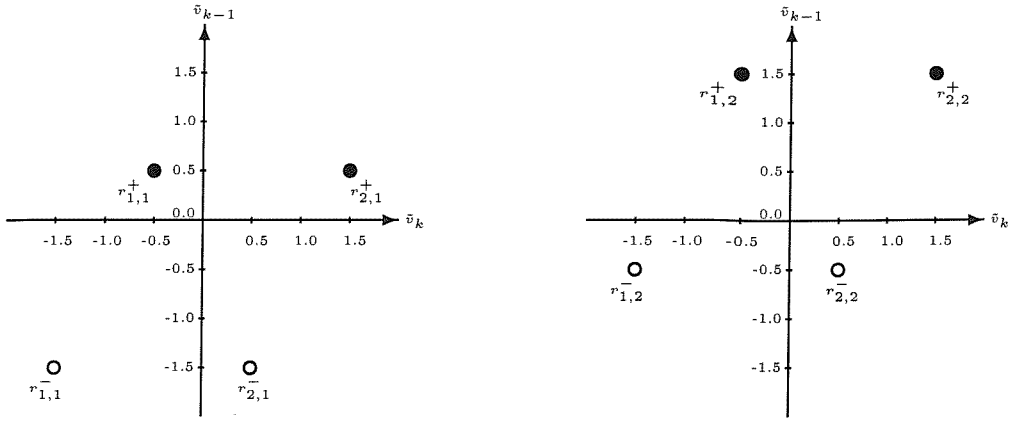
Figure 2.25: The noiseless channel states $\tilde{v}_k$, $\tilde{v}_{k-1}$ of a channel having a CIR of $F(z) = 1 + 0.5z^{-1}$ in a two-dimensional observation space. The filled circles represent the channel states in the set $V_{m,\tau}^{+1}$ corresponding to the transmitted symbol of $I_{k-\tau} = +1$ and the hollow circles represent the channel states in the set $V_{m,\tau}^{-1}$ corresponding to the transmitted symbol of $I_{k-\tau} = -1$, where the feedforward order is $m = 2$ and the decision delay of the equaliser is $\tau = 1$.

readily extended to a QAM scheme. It should be emphasized that even though the noiseless channel states are linearly separable for the conditions stated above for the equaliser's parameters, the optimal decision boundary will generally be nonlinear. However, the linear separability is a highly desired property to have, since the equalisation performance in this case is generally significantly better, than that of the nonlinearly separable case [71]. Note that the noiseless channel states $\mathbf{r}_i$ in the equaliser's observation space can be inseparable, as it will be demonstrated in Section 2.12.1.

2.11.2 Space Translation Properties of the Decision Feedback

In this section we provide a brief discourse on a technique, which can be used to reduce the number of states to be stored by the equaliser and also to eliminate the selection of the subset of states corresponding to the feedback symbol I_{k-2} in the example of Figure 2.26. In general, when $\tau > 1$, several feedback symbols influence the number of feedback states and hence the associated storage and complexity reduction may be significant.

For a particular feedback state $\mathbf{s}_{f,j}$ characterized by the specific symbols $\tilde{I}_k$ in the feedback register, the subsets $V_{m,\tau,j}^{+}$ and $V_{m,\tau,j}^{-}$ are related to the subsets $V_{m,\tau,l}^{+}$ and $V_{m,\tau,l}^{-}$ having



(a) The detected feedback symbol I_{k-2} is equivalent to **-1**

(b) The detected feedback symbol I_{k-2} is equivalent to **+1**

Figure 2.26: The noiseless channel states $\tilde{v}_k$, $\tilde{v}_{k-1}$ observed by the RBF DFE with feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol, assuming that the feedback symbols are correct. The channel has a CIR of $F(z) = 1 + 0.5z^{-1}$. The filled circles represent the channel states in the set $V^{+1}_{m,\tau,j}$ corresponding to the transmitted symbol of $I_{k-\tau} = +1$ and the hollow circles represent the channel states in the set $V^{-1}_{m,\tau,j}$ corresponding to the transmitted symbol of $I_{k-\tau} = -1$. Again, our final decision concerning the transmitted bit I_{k-2} is based on identifying the symbol $\mathcal{I}_i$, $i = 1, \ldots, \mathcal{M}$ associated with the highest probability.

consecutive feedback states of $\mathbf{s}_{f,j}$ and $\mathbf{s}_{f,l}$, respectively, by a linear transformation. This can be shown mathematically as follows. Upon rewriting Equation 2.11, in order to take into account the decision feedback state in the expression of the noisy channel output and assuming $n \leq L$ and $m = \tau + 1$, gives:

$$\mathbf{v}_k = \mathbf{F}\mathbf{I}_k + \boldsymbol{\eta}_k, \tag{2.108}$$

where $\boldsymbol{\eta}_k = \begin{bmatrix} \eta_k & \ldots & \eta_{k-m+1} \end{bmatrix}^T$. The transmitted symbols influencing $\mathbf{v}_k$ can be divided in three classes as follows: $\mathbf{I}_k = \begin{bmatrix} \mathbf{I}^T_{1,k} & \mathbf{I}^T_{2,k} & \mathbf{I}^T_{3,k} \end{bmatrix}^T$, where $\mathbf{I}_{1,k}$ indicates those symbols, which reside in the feedforward shift register, $\mathbf{I}_{2,k}$ denotes those in the feedback register and $\mathbf{I}_{3,k}$ consists of the rest of the symbols that influence $\mathbf{v}_k$ but are left out by the DFE. These symbols can be written as:

$$\left.\begin{aligned} \mathbf{I}_{1,k} &= \begin{bmatrix} I_k & \ldots & I_{k-\tau} \end{bmatrix}^T \\ \mathbf{I}_{2,k} &= \begin{bmatrix} I_{k-\tau-1} & \ldots & I_{k-\tau-n} \end{bmatrix}^T \\ \mathbf{I}_{3,k} &= \begin{bmatrix} I_{k-\tau-n-1} & \ldots & I_{k-m-L+1} \end{bmatrix}^T \end{aligned}\right\}. \tag{2.109}$$

Furthermore, the $m \times (m + L)$ CIR-related matrix $\mathbf{F}$ has the form

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 & \mathbf{F}_3 \end{bmatrix}, \tag{2.110}$$

with the $m \times (\tau + 1)$ matrix $\mathbf{F}_1$, $m \times n$ matrix $\mathbf{F}_2$ and $m \times (m + L - n - \tau - 1)$ matrix $\mathbf{F}_3$ defined by

$$\mathbf{F}_1 = \begin{bmatrix} f_0 & f_1 & \cdots & f_\tau \\ 0 & f_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & f_1 \\ 0 & \cdots & 0 & f_0 \end{bmatrix}, \tag{2.111}$$

$$\mathbf{F}_2 = \begin{bmatrix} f_{\tau+1} & f_{\tau+2} & \cdots & f_{\tau+n} \\ f_\tau & f_{\tau+1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & f_{n+1} \\ f_1 & \cdots & f_{n-1} & f_n \end{bmatrix}, \tag{2.112}$$

$$\mathbf{F}_3 = \begin{bmatrix} f_{\tau+n+1} & f_{\tau+n+2} & \cdots & f_{m+L-1} \\ f_{\tau+n} & f_{\tau+n+1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & f_{L+1} \\ f_{n+1} & \cdots & f_{L-1} & f_L \end{bmatrix}, \tag{2.113}$$

where $f_i = 0$ for $i < L$. Explicitly, $\mathbf{F}_1$ hosts those CIR taps, which affect the feedforward section, symbols contained by $\mathbf{I}_{1,k}$, $\mathbf{F}_2$ encompasses those, which weight the feedback symbols $\mathbf{I}_{2,k}$, while $\mathbf{F}_3$ contains the symbols not considered by the DFE. Under the assumption that the feedback symbol is correct, that is $\tilde{\mathbf{I}}_{feedback,k-\tau} = \mathbf{I}_{2,k}$ and based on Equation 2.108, the noise-free channel output vector of Equation 2.6 can be rewritten as

$$\begin{aligned} \tilde{\mathbf{v}}_k &= \mathbf{F}_1\mathbf{I}_{1,k} + \mathbf{F}_2\tilde{\mathbf{I}}_{feedback,k-\tau} + \mathbf{F}_3\mathbf{I}_{3,k} \\ &= \tilde{\mathbf{v}}'_k + \mathbf{F}_2\tilde{\mathbf{I}}_{feedback,k-\tau}, \end{aligned} \tag{2.114}$$

where we introduced

$$\tilde{\mathbf{v}}'_k = \mathbf{F}_1\mathbf{I}_{1,k} + \mathbf{F}_3\mathbf{I}_{3,k}. \tag{2.115}$$

Thus the linear transformation between the consecutive noise-free channel output vectors of $\tilde{\mathbf{v}}_k$ and $\tilde{\mathbf{v}}'_k$ is provided by the term $\mathbf{F}_2\tilde{\mathbf{I}}_{feedback,k-\tau}$ in Equation 2.114.

Using the CIR of Equation 2.13 we have $f_{\tau+1} = 0, f_1 = 0.5, f_0 = 1$ in Equations 2.111 – 2.113, yielding $\mathbf{F}_1 = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}$, $\mathbf{F}_2 = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}$ and $\mathbf{F}_3 = 0$. Assuming $I_k = +1$ and $I_{k-1} = +1$, we have $\mathbf{I}_{1,k} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and evaluating Equation 2.115 gives:

$$\tilde{\mathbf{v}}'_k = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 0 = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix}. \tag{2.116}$$

Hence from Equation 2.114, for the specific feedback state of $\mathbf{s}_{f,1} = \tilde{\mathbf{I}}_{feedback,k-\tau} = [-1]$ the noiseless channel state is given by

$$\mathbf{r}_{2,1}^{+} = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ -0.5 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix}, \tag{2.117}$$

while for the feedback state of $\mathbf{s}_{f,2} = \tilde{\mathbf{I}}_{feedback,k-\tau} = [+1]$ the noiseless channel state is given by

$$\mathbf{r}_{2,2}^{+} = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}, \tag{2.118}$$

as seen in Table 2.11.

We note that the linear transformation of Equation 2.114 between the consecutive noise-free channel outputs of $\tilde{\mathbf{v}}_k$ and $\tilde{\mathbf{v}}'_k$ depends on the feedback states $\mathbf{s}_{f,j}$ and the CIR. The geometric distance amongst the corresponding points of the set of $V^{+}_{m,\tau,j}$ and of the set $V^{-}_{m,\tau,j}$ for the same feedback state $\mathbf{s}_{f,j}$ is not altered by the transformation. Using the example in Figure 2.26, the geometric distance between the points $\mathbf{r}^{+}_{2,1}$ and $\mathbf{r}^{-}_{2,1}$ corresponding to the feedback symbol $I_{k-2} = -1$, is equivalent to the geometric distance between the points $\mathbf{r}^{+}_{2,2}$ and $\mathbf{r}^{-}_{2,2}$ corresponding to the feedback symbol $I_{k-2} = -1$. Thus it is sufficient to consider just one particular feedback state, when examining the Symbol Error Rate (SER) performance.

Previous research [70, 72] pointed out furthermore that the elements of $\tilde{\mathbf{v}}'_k$ can be computed recursively. The ith element of $\tilde{\mathbf{v}}'_k$, where, $i = m-1, \ldots, 2, 1$, can be represented by its unit delayed version as follows:

$$v_{k-i} = z^{-1} v_{k+1-i} \qquad i = m-1, \ldots, 2, 1, \tag{2.119}$$

where z^{-1} is the unit-delay operator. From Equation 2.114, the ith and $(i-1)$th elements of $\tilde{\mathbf{v}}'_k$ can be written as

$$v_{k-i} = v'_{k-i} + \sum_{j=1}^{n} f_{m+j-1-i} I_{k-\tau-j} \tag{2.120}$$

$$v_{k-i+1} = v'_{k-i+1} + \sum_{j=1}^{n} f_{m+j-1-i+1} I_{k-\tau-j}. \tag{2.121}$$

Using Equation 2.119, 2.120 and 2.121, we have:

$$\begin{aligned} v_{k-i} &= z^{-1} v_{k+1-i} \\ &= z^{-1} \left(v'_{k-i+1} + \sum_{j=1}^{n} f_{m+j-1-i+1} I_{k-\tau-j} \right) \\ &= z^{-1} v'_{k-i+1} + \sum_{j=1}^{n} f_{m+j-i} I_{k-1-\tau-j} \end{aligned}$$

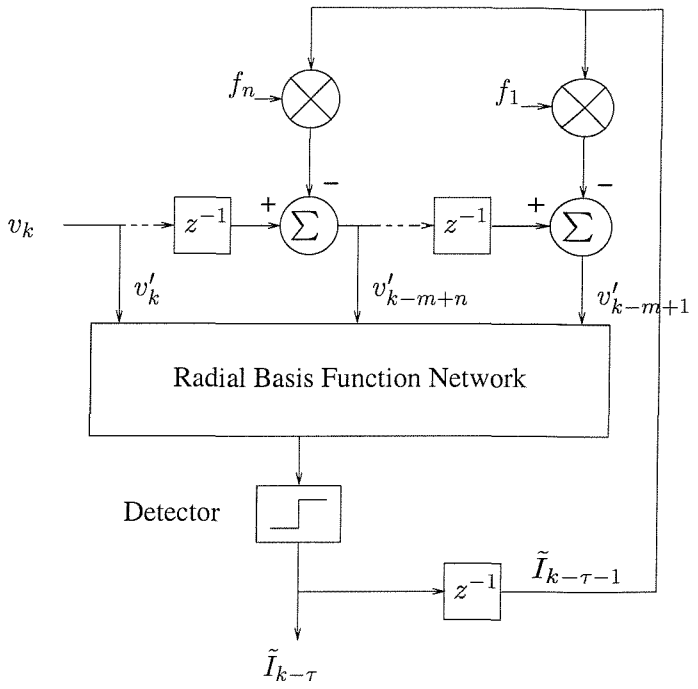


Figure 2.27: Space-translated RBF DFE

$$
\begin{aligned}
v'_{k-i} + \sum_{j=1}^{n} f_{m+j-i-1} I_{k-\tau-j} &= z^{-1} v'_{k-i+1} + \sum_{j=1}^{n} f_{m+j-i} I_{k-1-\tau-j} \\
&= z^{-1} v'_{k-i+1} + \sum_{j=2}^{n+1} f_{m+j-i-1} I_{k-\tau-j} \\
v'_{k-i} &= z^{-1} v'_{k+1-i} - f_{m-i} I_{k-\tau-1} + f_{m-i+n} I_{k-\tau-n-1} \\
& \qquad i = m-1, \ldots, 2, 1. \qquad (2.122)
\end{aligned}
$$

Upon substituting $n = L$ into Equation 2.122 we arrive at:

$$
\begin{aligned}
v'_{k-i} &= z^{-1} v'_{k+1-i} - f_{m-i} I_{k-\tau-1} \qquad (2.123) \\
v'_k &= v_k. \qquad (2.124)
\end{aligned}
$$

Based on this interpretation of decision feedback, an alternative DFE structure is depicted in Figure 2.27. This version of the space-translated RBF DFE realises the same optimal solution as the subset centre selection RBF DFE depicted in Figure 2.20. However, the space-translated RBF DFE of Figure 2.27 removes the requirement of different set of centres for different decision feedbacks and has hence a clear advantages in hardware implementational terms. The decision feedback 'merges' the channel states corresponding to different feedback states and hence the DFE of Figure 2.27 can be studied more conveniently in the translated $\mathbf{v}'$-space. [1]

[1] This property leads to the implementation of the so-called Minimum BER (MBER) DFE based on either

2.12 Simulation Results

2.12.1 Performance of RBF Assisted Equalisers over Dispersive Gaussian Channels

In all our results presented in this section the transmitted symbols I_k were equiprobable binary symbols assuming values from the set $\{\pm 1\}$. Therefore the weights of the RBF network were fixed to $w_i = +1$, if the RBF centroids $\mathbf{c}_i$ seen in Figure 2.12 correspond to a positive channel state $\mathbf{v}_i^+$ and to $w_i = -1$, if the RBF centroids $\mathbf{c}_i$ correspond to a negative channel state $\mathbf{v}_i^-$, as explained in Section 2.9.1. The noise variance σ_η^2 was fixed to unity, while the power of the transmitted symbol was varied according to the SNR per bit, namely E_b/N_0. The transmitted symbol was oversampled by a factor of 10 and it was pulse-shaped. Both the transmitter and receiver had a square root Nyquist filter [61] with a roll-off factor of 0.5. The combined transfer function of these two filters produced a raised cosine filter and this design satisfies the Nyquist criterion of zero ISI at sampling instants.

Initially the centres of the RBF network were positioned at the desired channel states seen for example in Figure 2.3. The width of the RBF network ρ was set to $2\sigma_\eta^2$. We assumed that the CIR was known and the number of hidden nodes was set to $n_s = 2^{L+m}$. In practice the CIR can be estimated using channel sounding [75, 76] and using the estimated CIR would result in some performance degradation. The impulse responses of the channels used for the simulations were characterized by Figure 2.21(a) for the two-path channel, Figure 2.21(b) for the three-path channel and Figure 2.21(c) for the five-path channel.

The BER performance of the RBF network was compared with that of the linear MSE equaliser [59] (pp. 607-612). The tap weights of the linear MSE equaliser were set to obtain the best possible performance and both schemes used the same number of taps given by m. Figure 2.28 and Figure 2.29 shows our BER performance comparison for the two-path channel and the three-path channel, respectively. The two-path results of Figure 2.28 show that for the same number of taps the RBF network equaliser provides superior performance in comparison to the linear MSE equaliser, before the residual BER is reached, above which the BER performance did not improve upon increasing E_b/N_0. Beyond this point the RBF network equaliser and the linear MSE equaliser have a similar BER performance.

This can be explained graphically by first observing the desired channel states in the channel observation space of Figure 2.30. For the two-path channel environment of Figure 2.21(a) and an equaliser having three taps, the desired channel states and a linear decision boundary surface provided by the linear MSE equaliser is shown in Figure 2.30. Note that the noiseless channel output $\tilde{\mathbf{v}} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ due to the transmitted data

a linear filter [70] or on the so-called support vector machine [73, 74] proposed by Chen et. al. that construct hyperplanes, which can separate the different signal classes.

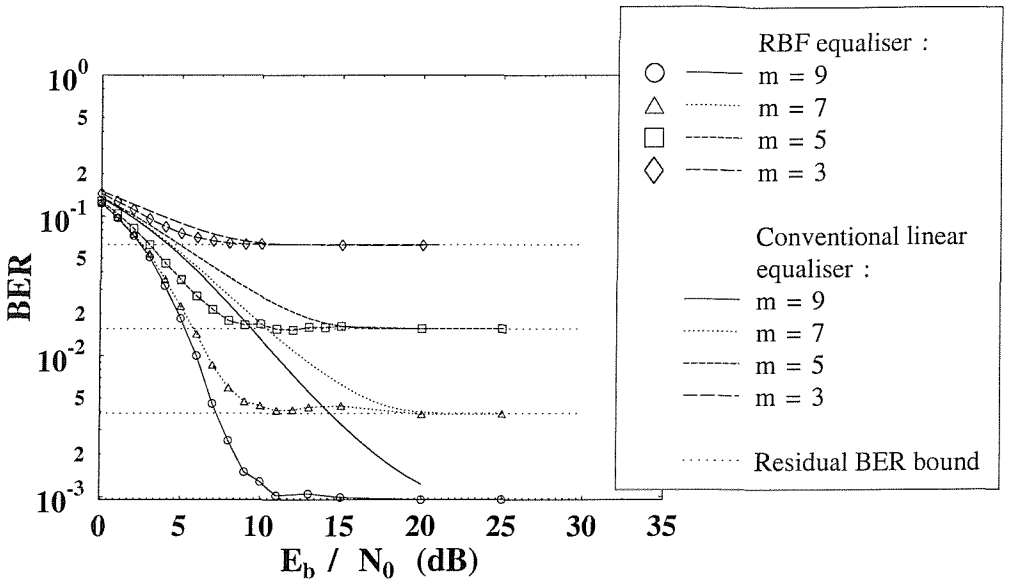


Figure 2.28: BER versus E_b/N_0 performance of the RBF equaliser using no decision feedback upon varying the number of equaliser taps m over the two-path Gaussian channel of Figure 2.21(a) using BPSK. The performace is compared to that of the linear MSE equaliser using m number of taps. The residual BER bound ($= \frac{1}{2^{m+L}}$, where $L+1$ is the CIR length) is shown for different values of m. The residual BER is due to the constellation points appearing on top of each other in Figure 2.30.

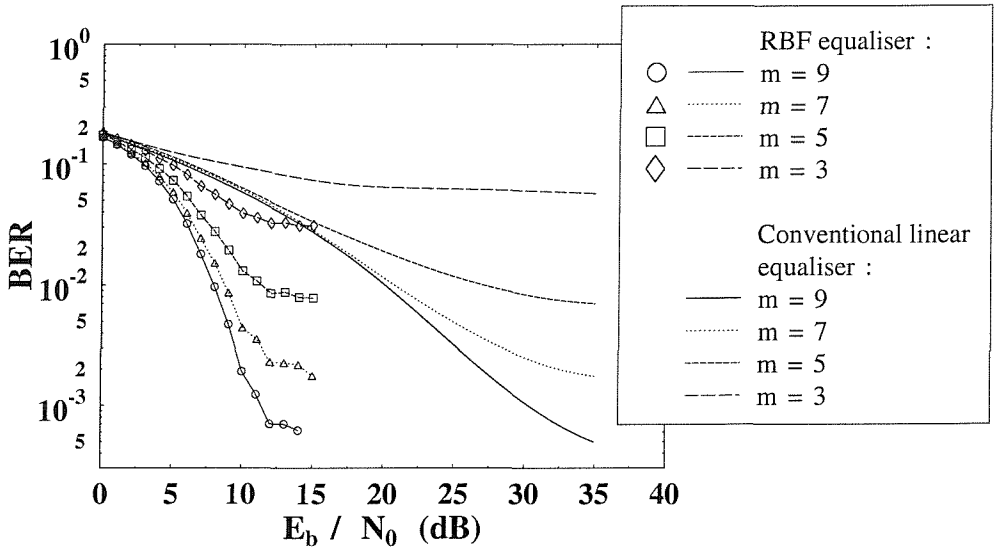


Figure 2.29: BER versus E_b/N_0 performance of the RBF equaliser using no decision feedback upon varying number of equaliser taps m over the three-path dispersive Gaussian channel of Figure 2.21(b) employing BPSK. The performace is compared to that of the linear MSE equaliser using m number of taps.

sequence of $\{-1+1-1+1\}$ and $\{+1-1+1-1\}$ corresponds to both $I_{k-\tau}=+1$ and $I_{k-\tau}=-1$. Thus the channel states $\mathbf{r}^+$ and $\mathbf{r}^-$ are inseparable both linearly and nonlinearly at that point, even when the input dimension is increased. This provides the performance limitation manifested in terms of the residual BER for both the linear MSE equaliser and the RBF network equaliser. The value of the residual BER is dependent on the relative frequency of encountering this inseparable channel state scenario. For example, in the case of the two-path channel environment mentioned above, where there are two channel states corresponding to the noiseless channel output vector $\tilde{\mathbf{v}}=\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ and both are classified as corresponding to $I_{k-\tau}=+1$ or $I_{k-\tau}=-1$, one channel state out of the total of n_s legitimate channel states will be classified wrongly, irrespective of E_b/N_0. Thus, the minimum achievable bit error rate will be $\frac{1}{n_s}$ for a particular equaliser order m. This explains the BER residual in Figure 2.28. The BER residual 'bound' of $\frac{1}{n_s}$ is also shown in Figure 2.28 using dashed line for the various m values employed. The three-path results of Figure 2.29 also show superior performance in comparison to the linear MSE equaliser, before the residual BER is reached. Again, the residual BER 'bound' can be explained by the inseparable channel states.

The BER performance generally improves upon increasing the number of equaliser taps m, as does the 'bound' $\frac{1}{n_s}$. The dimension of the channel observation space that increases with increasing m has the effect of increasing the Euclidean distance between the desired channel states and therefore improves the separability between $\tilde{\mathbf{v}}^+$ and $\tilde{\mathbf{v}}^-$, but does not irradicate the ambiguity associated with $\tilde{\mathbf{v}}=\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$.

Figure 2.31, Figure 2.32 and Figure 2.33 show the BER performance of the RBF network equaliser in conjunction with decision feedback for the two-path channel, three-path channel and five-path channels of Figure 2.21, respectively. The equaliser feedforward order m is fixed to $\tau+1$, while the feedback order was set to $n=L$, as described in Section 2.11. The results shows that the decision feedback structure not only decreases the computational complexity, since less taps and less hidden nodes are neccesary, it also substantially improves the BER performance. The residual BER is eliminated, since the desired states $\tilde{\mathbf{v}}^+$ and $\tilde{\mathbf{v}}^-$ that correspond to the same point in the channel observation space have now different feedback states and the set of noiseless channel states $V^+_{m,\tau,j}$ and $V^-_{m,\tau,j}$ are now separable. This confirms the findings by Chen, Mulgrew, Chng and Gibson [70] that the noiseless channel states corresponding to a different transmitted symbol are linearly separable, provided that the decision delay, feedforward section and feedback section length of the RBF DFE are chosen to be $\tau=L$, $m=\tau+1=L+1$ and $n=L+m-\tau-1=L$.

Note furthermore that the error propagation due to erroneous decision feedback has a moderate effect on the performance of the BPSK RBF network equaliser, amounting to

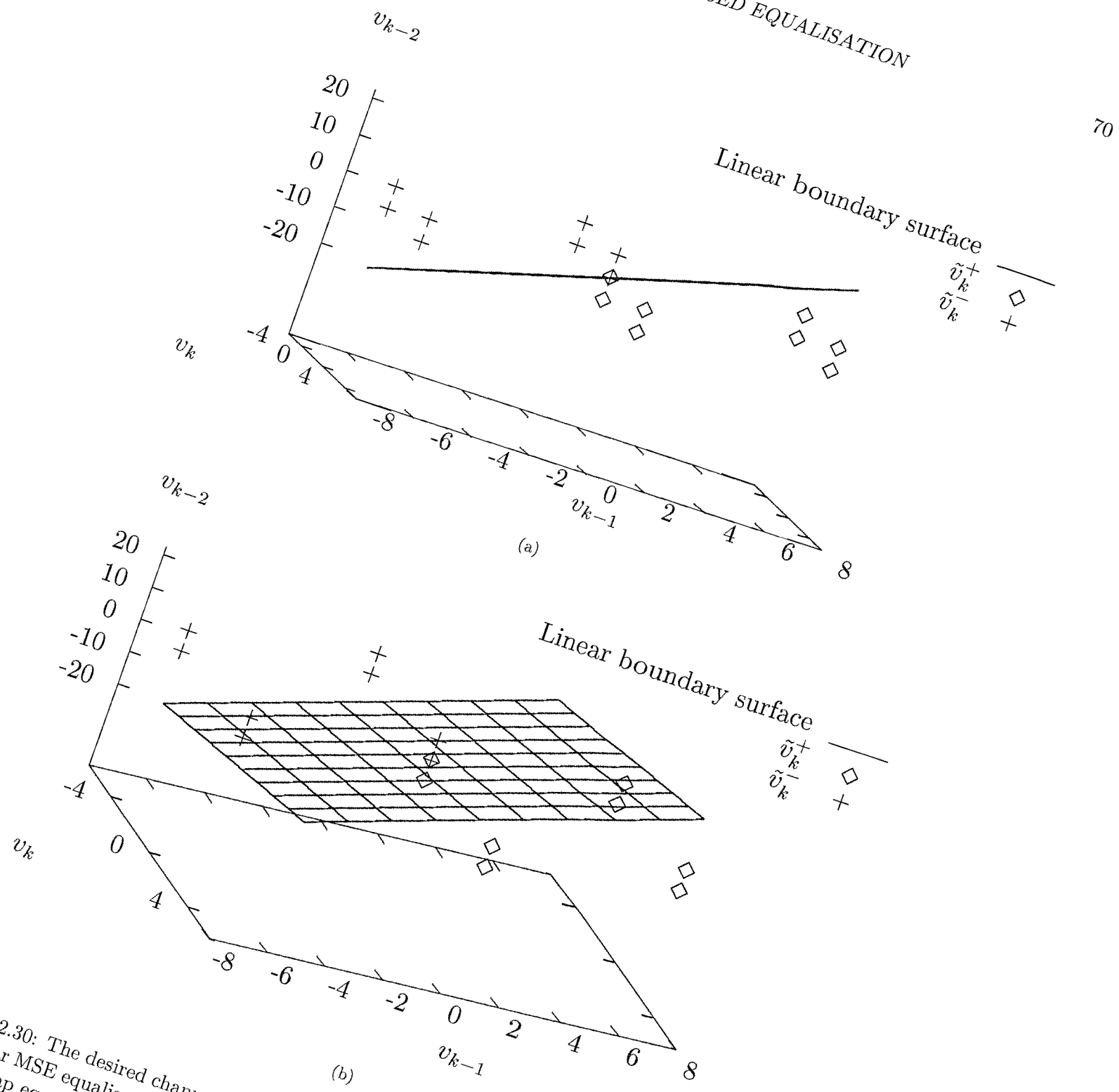


Figure 2.30: The desired channel states and a linear decision boundary surface provided by the linear MSE equaliser in a three-dimensional channel observation space corresponding to a three-tap equaliser and the CIR of $F(z) = 0.707 + 0.707z^{-1}$, viewed at different angles.

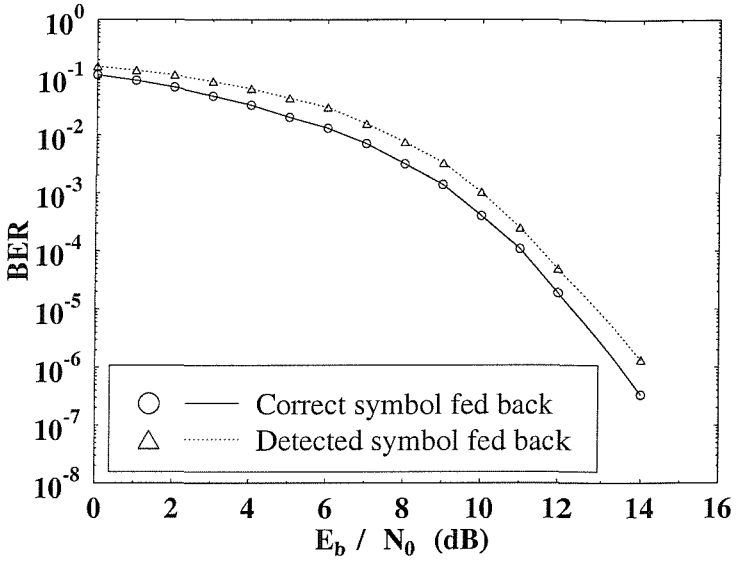


Figure 2.31: BER versus E_b/N_0 performance of the BPSK RBF equaliser with decision feedback over the dispersive **two-path** Gaussian channel of Figure 2.21(a). The equaliser has a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol.

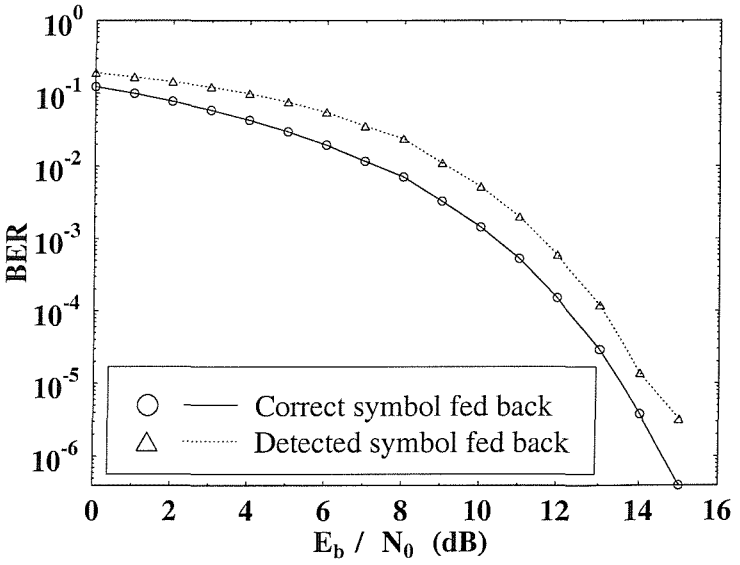


Figure 2.32: BER versus E_b/N_0 performance of the BPSK RBF equaliser with decision feedback over the dispersive **three-path** Gaussian channel of Figure 2.21(b). The equaliser has a feedforward order of $m = 3$, feedback order of $n = 2$ and decision delay of $\tau = 2$ symbols.

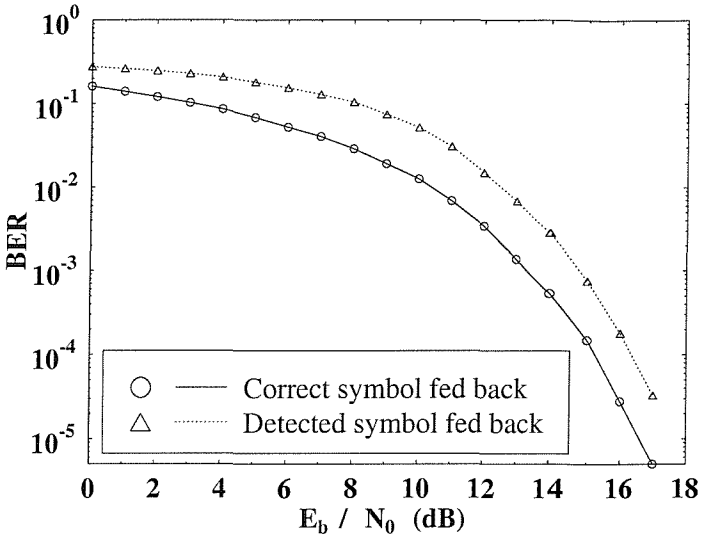


Figure 2.33: BER versus E_b/N_0 performance of the BPSK RBF equaliser with decision feedback over the dispersive **five-path** Gaussian channel of Figure 2.21(c). The equaliser has a feedforward order of $m = 5$, feedback order of $n = 4$ and decision delay of $\tau = 4$ symbols.

around 1dB performance degradation at BER $= 10^{-4}$ for all the three channels of Figure 2.21.

2.12.2 Performance of Adaptive RBF DFE

As our next endeavour, the adaptive performance of the RBF network equaliser employing the K-means clustering algorithm of Section 2.9.5 was investigated. Firstly, the average normalised MSE of the vector centres at signalling interval k was defined as:

$$\text{MSE}(\mathbf{c}, k) = \frac{1}{n_s \sigma_{\tilde{v}}^2} \sum_{i=1}^{n_s} \|\mathbf{c}_{i,k} - \mathbf{c}_{i,opt}\|^2, \tag{2.125}$$

where n_s is the number of RBF centres, $\sigma_{\tilde{v}}^2$ is the variance of the noise-free received signal, $\mathbf{c}_{i,k}, i = 1, \ldots, n_s, k = 0, \ldots, \infty$ represents the ith assumed RBF centre at signalling interval k and $\mathbf{c}_{i,opt}$ is the vector associated with the ith desired or assumed 'true' RBF center. The K-means clustering technique operates by iteratively adjusting the RBF centres upon every sampling instance according to Equation 2.94 during training mode, while Equation 2.96 is used during the decision-directed mode. The centres' MSE convergence performance is demonstrated over the two-path channel environment in Figure 2.34 using different number of training symbols, while in Figure 2.35 upon varying the learning rate μ_c of Equation 2.94 and 2.96. The results show a good convergence performance for our stationary two-path

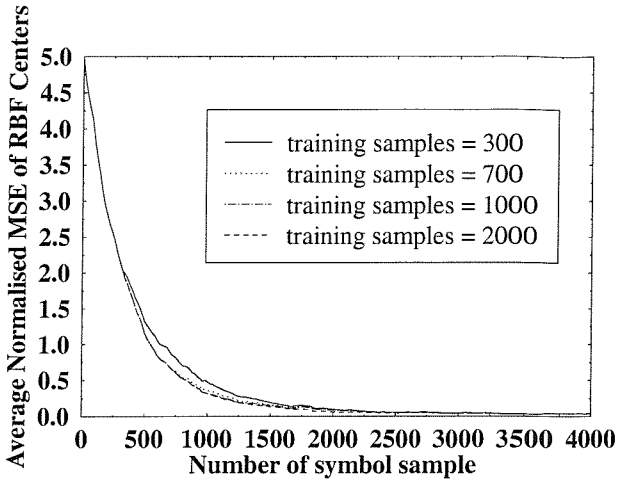
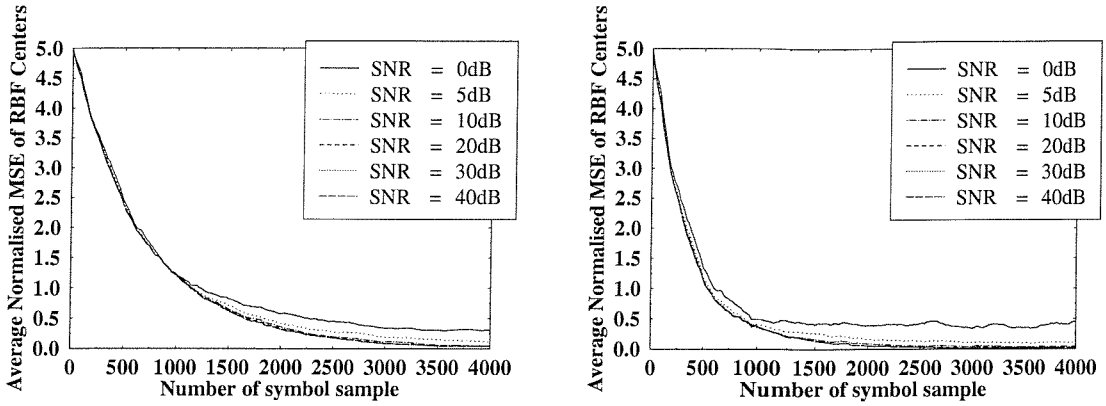


Figure 2.34: The MSE of the BPSK RBF equaliser centres versus transmitted symbol index for various numbers of training samples using the **vector centre clustering algorithm** of Section 2.9.5 over the two-path channel environment of Figure 2.21(a). The equaliser had $m = 5$ feedforward taps and a decision delay of $\tau = 2$ symbols. The centre learning rate μ_c of Equations 2.94 and 2.96 was set to 0.1 and the SNR was 10dB.

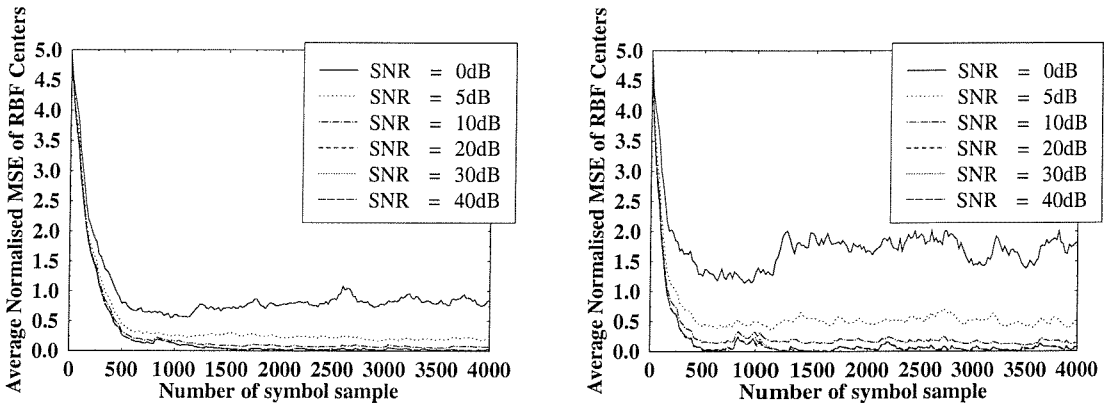
channel of Figure 2.21(a) upon invoking the decision-directed learning algorithm of Equation 2.96. However, further simulations have to be carried out, in order to investigate the effect of time-varying wideband mobile channels. As the learning rate μ_c is increased, the centres converge faster to their desired positions, but as expected, the MSE curves of the centres become more spurious, especially at low SNRs as we can see from Figure 2.35. Based on these results we recommend using a variable learning rate μ_c, where μ_c is set to a higher rate during the training mode so that the equaliser converges faster and is set to a lower rate during the decision-directed learning mode, in order to reduce the spuriousity of the centre MSE.

The performance of the scalar centre clustering algorithm described in Section 2.10 is demonstrated over the same two-path Gaussian channel environment in Figure 2.36 and Figure 2.37. Comparing Figure 2.35 and Figure 2.36 using various centre learning rates μ_r, shows that the scalar centre clustering algorithm provides a significantly faster convergence rate, since the number of scalar centres $n_{s,f} = \mathcal{M}^{L+1} = 4$ ($\mathcal{M} = 2, L = 1$) is only dependent on the number of symbol constellation points, $\mathcal{M}$, and on the CIR length $L + 1$. Hence the number of scalar centres is significantly less than the number of vector centres given by $n_s = \mathcal{M}^{L+m} = 64$ ($m = 5$), which is additionally dependent on the equaliser order m as well. However, the MSE learning curves of the centres are more spurious in conjunction with the scalar centre clustering algorithm, since the value of a scalar center affects the value of a few vector centres that contain that particular scalar centre and thus the estimation error



(a) (a) $\mu_c = 0.05$

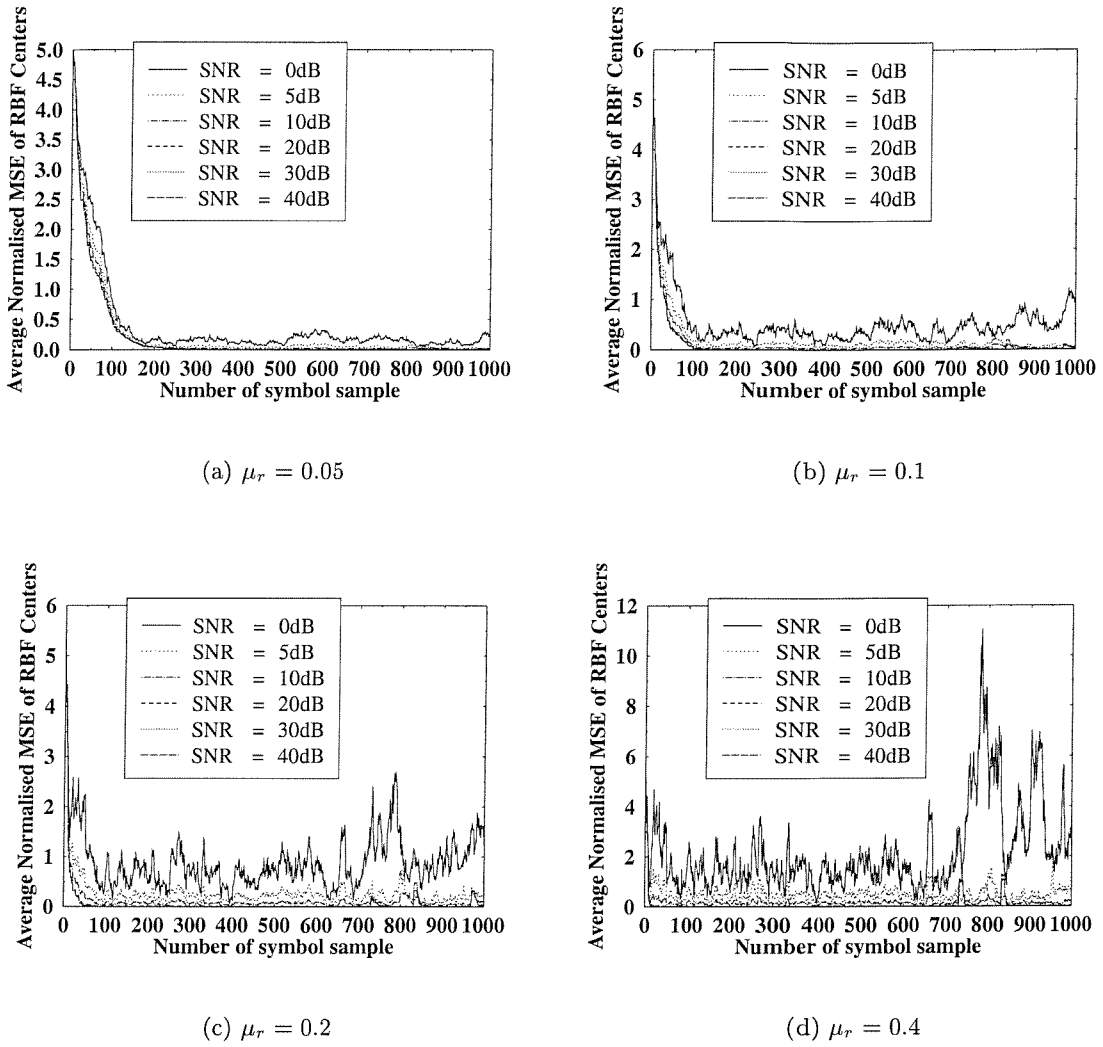
(b) (b) $\mu_c = 0.1$



(c) (c) $\mu_c = 0.2$

(d) (d) $\mu_c = 0.4$

Figure 2.35: The MSE of the BPSK RBF equaliser centres versus transmitted symbol index for **various learning rates** μ_c of the centres using the **vector centre clustering algorithm** of Section 2.9.5 over the two-path channel environment of Figure 2.21(a). The equaliser had $m = 5$ feedforward taps and a decision delay of $\tau = 2$ symbols, and the number of training symbols was 700.



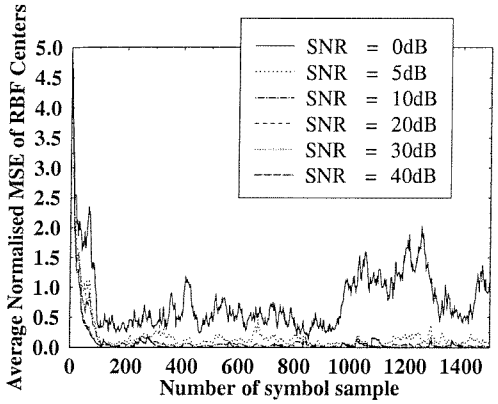
(a) $\mu_r = 0.05$

(b) $\mu_r = 0.1$

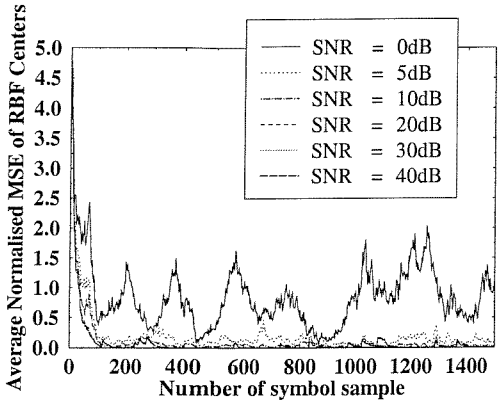
(c) $\mu_r = 0.2$

(d) $\mu_r = 0.4$

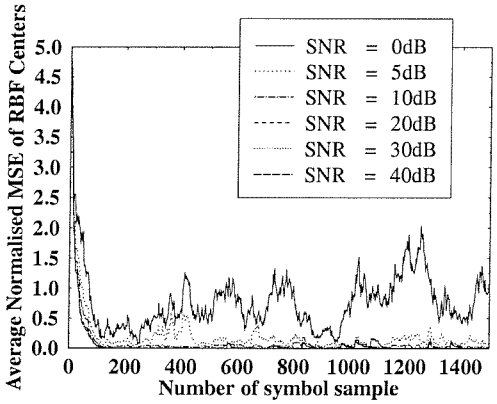
Figure 2.36: The MSE of the BPSK RBF equaliser centres versus transmitted symbol index for **various learning rates** μ_r of the centres using the **scalar centre clustering algorithm** of Section 2.10 with 700 training symbols over the two-path channel environment of Figure 2.21(a). The equaliser had $m = 5$ feedforward taps and a decision delay of $\tau = 2$ symbols.



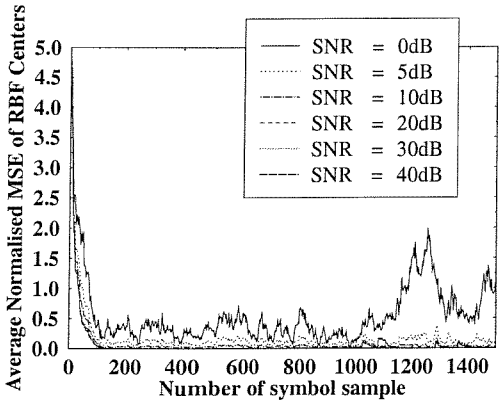
(a) 25 training symbols



(b) 50 training symbols



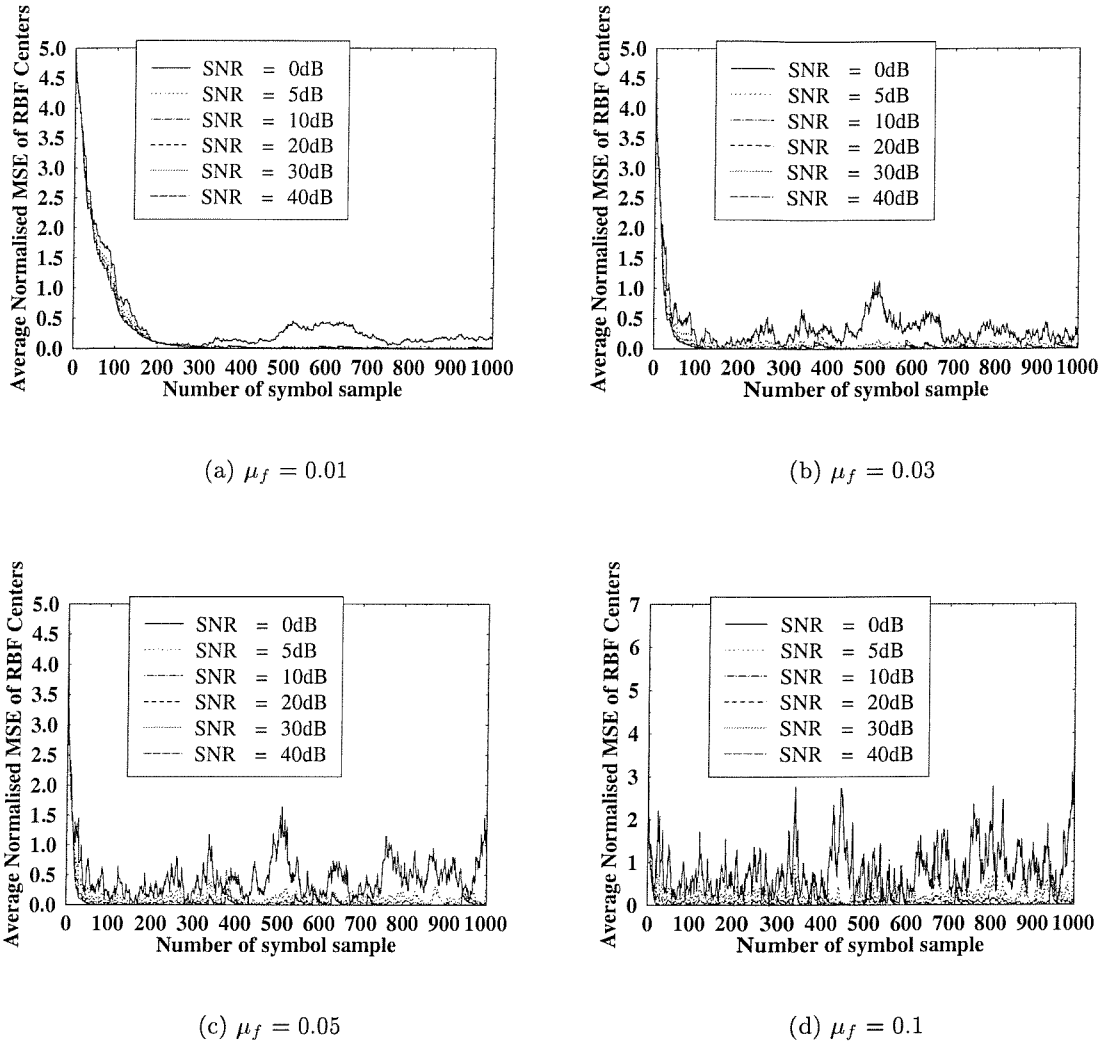
(c) 300 training symbols



(d) 1000 training symbols

Figure 2.37: The MSE of the BPSK RBF equaliser centres versus transmitted symbol index for **various number of training samples** using the **scalar centres clustering algorithm** of Section 2.10 over the two-path channel environment of Figure 2.21(a). The equaliser had $m = 5$ feedforward taps and a decision delay of $\tau = 2$ symbols. The centre learning rate μ_r was set to 0.1.

of a scalar center will be magnified, when we examine the average normalised MSE of the vector centres in Figure 2.36. Figure 2.37 shows the average normalised MSE of the RBF centres for a varying number of training symbols. Note that the algorithm still converges during the decision-directed mode, although the MSE curve behaves more spuriously during this mode compared to the learning phase, especially at low SNRs.



(a) $\mu_f = 0.01$

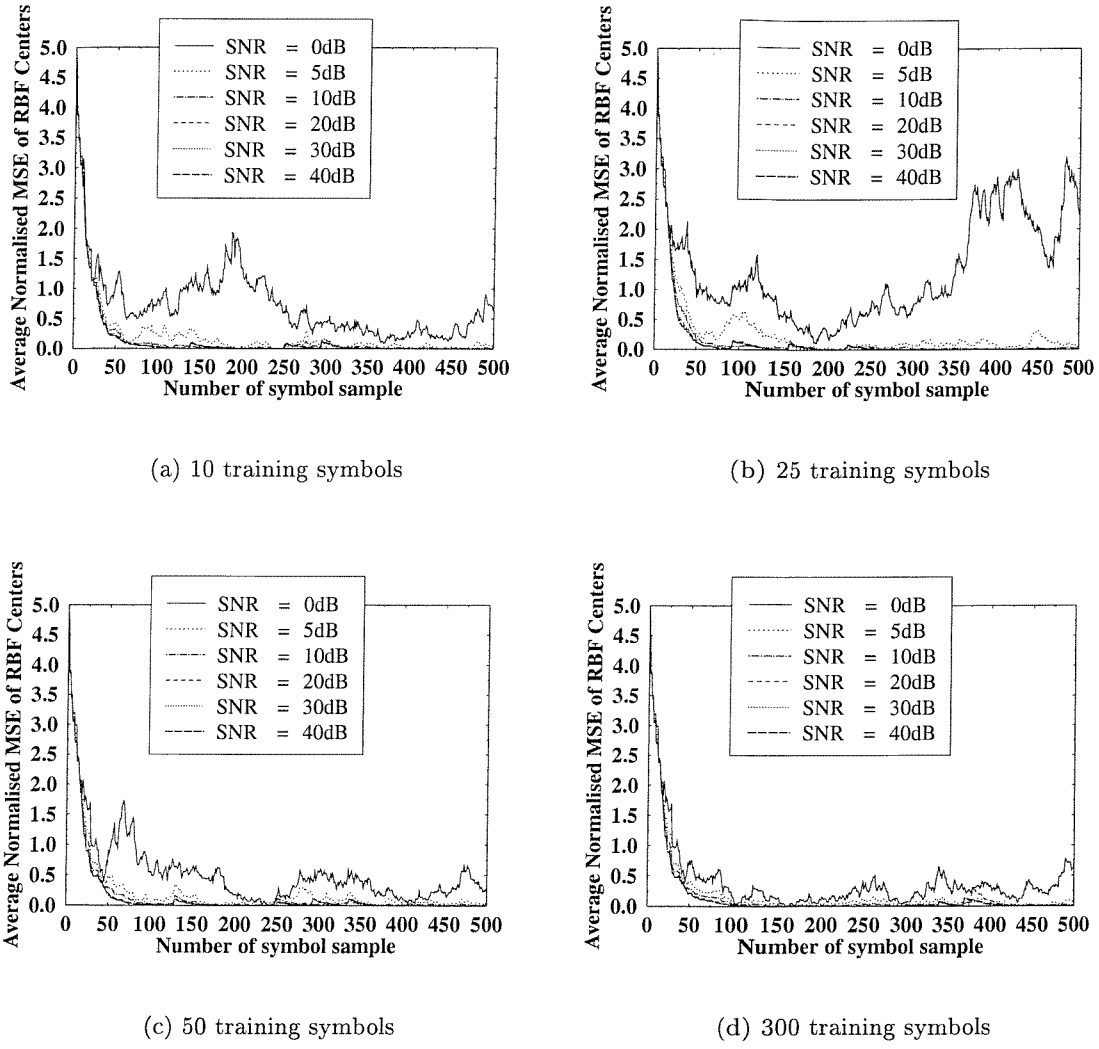
(b) $\mu_f = 0.03$

(c) $\mu_f = 0.05$

(d) $\mu_f = 0.1$

Figure 2.38: The MSE of the BPSK RBF equaliser centres versus transmitted symbol index for **various learning rates** μ_f using the **LMS channel estimator technique** of Section 2.9.4 with 300 training symbols over the two-path channel environment of Figure 2.21(a). The equaliser had $m = 5$ feedforward taps and a decision delay of $\tau = 2$ symbols.

The centres' MSE convergence performance for the channel estimation method using the LMS algorithm described in Section 2.9.4 is demonstrated in Figures 2.38 and 2.39. Comparing Figure 2.36 and 2.38 using varying learning rates shows that the LMS channel



(a) 10 training symbols

(b) 25 training symbols

(c) 50 training symbols

(d) 300 training symbols

Figure 2.39: The MSE of the BPSK RBF equaliser centres versus transmitted symbol index for **various number of training samples** using the **LMS channel estimator technique** of Section 2.9.4 over the two-path channel environment of Figure 2.21(a). The equaliser had $m = 5$ feedforward taps and a decision delay of $\tau = 2$ symbols. The channel estimator learning rate μ_f was set to 0.03.

estimator technique provides faster convergence rate at a given learning rate, since the number of CIR coefficients that were adapted according to Equation 2.90 by the LMS channel estimator is less than the number of scalar centres of the scalar clustering algorithm. However, if we compare Figure 2.36(a), (b), (c) and (d) with Figure 2.38(a), (b), (c) and (d), respectively, they show rather similar convergence rates during the training mode since the number of scalar centres ($2^{L+1} = 4$) is not too high compared to the number of channel coefficients ($L + 1 = 2$) to be learnt adaptively. Figure 2.39 shows the average normalised MSE of the RBF centres for varying number of training symbols. Again, the LMS channel estimator technique of Section 2.9.4 still converges during the decision-directed mode.

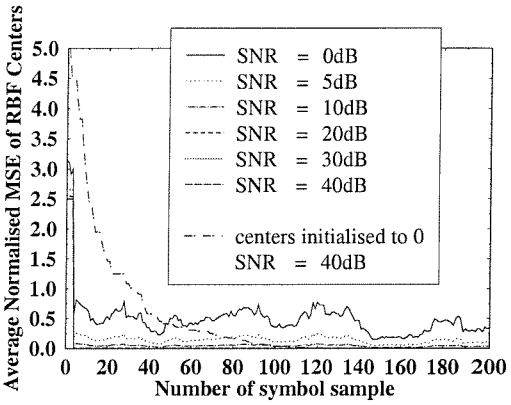


Figure 2.40: The MSE of the BPSK RBF equaliser centres versus transmitted symbol index using the **scalar centre clustering algorithm** of Section 2.10 with **centers initialised to the corresponding noisy channel states**, as described by Equation 2.97 over the two-path channel environment of Figure 2.21(a). The MSE of the BPSK RBF equaliser centres learnt using the scalar centre clustering algorithm with centres initially set to 0 for SNR = 40dB is shown for comparison. The equaliser had $m = 5$ feedforward taps and a decision delay of $\tau = 2$ symbols. The learning rate μ_r was set to 0.1 and the number of training symbols is 200.

Figure 2.40 shows the centres' MSE convergence performance, when the scalar centres clustering algorithm was initialised with the corresponding noisy channel states, as described in Section 2.10 over the same two-path Gaussian channel environment. Comparing Figure 2.36(b) and Figure 2.40 reveals that the initialisation to the corresponding noisy channel states significantly increases the convergence rate of the clustering algorithm at a low additional computational cost. The convergence rate is also seen to be faster than that of the LMS channel estimator technique when we compare Figure 2.38(d) with Figure 2.40, since the number of scalar centres ($2^{L+1} = 4$) is not significantly higher, than the number of channel coefficients ($L + 1 = 2$) to be learnt adaptively. Note that since the number of scalar centres ($\mathcal{M}^{L+1}$) increases exponentially with the length L of the CIR, the LMS

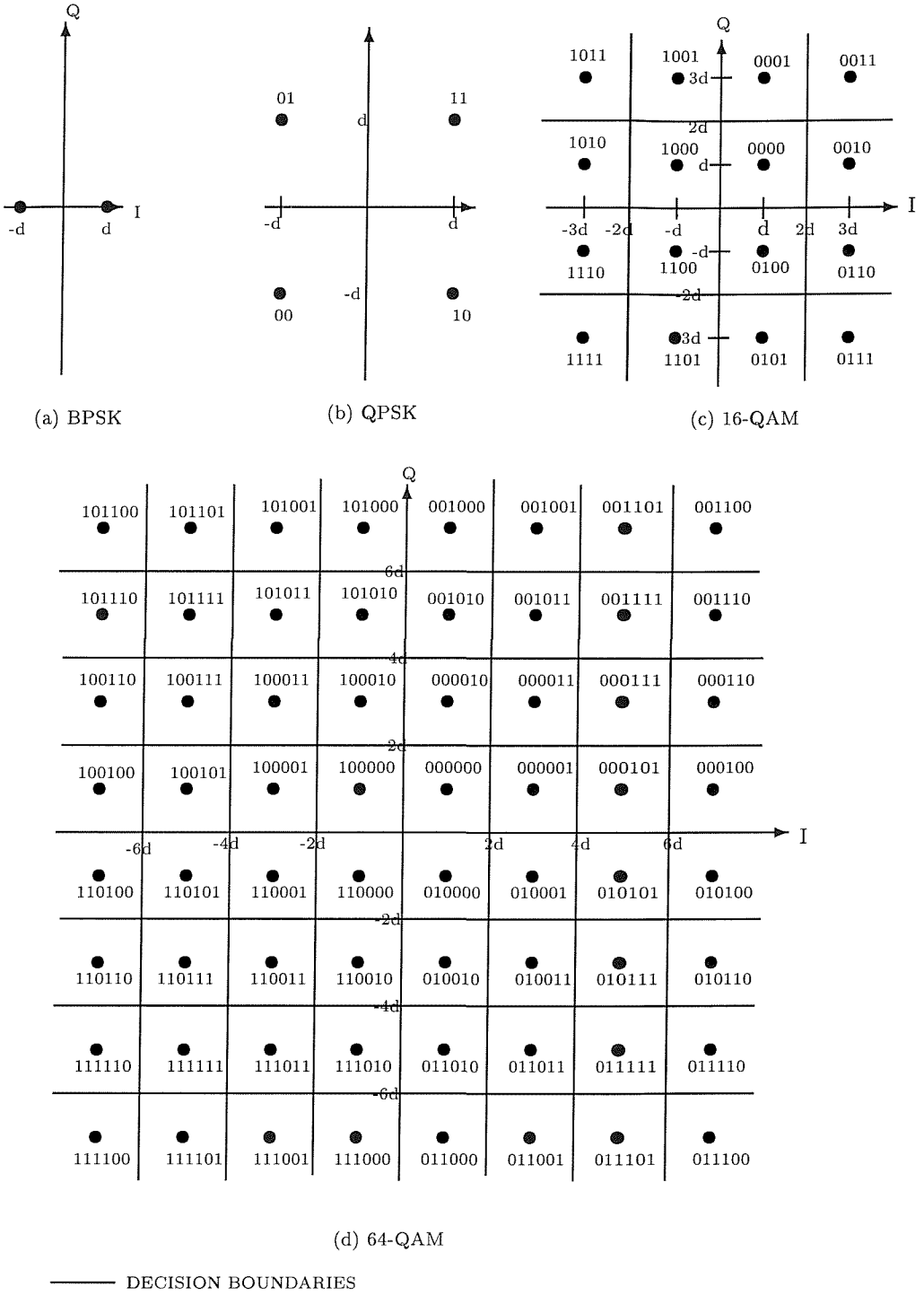
channel estimation technique will have a better convergence rate for high order modulation scheme and high CIR lengths, than the scalar centre clustering algorithm using the above-mentioned initialisation to the noisy centres.

2.12.3 Performance of the RBF Equaliser for Square-QAM over Gaussian Channels

In this section the performance of the RBF equaliser is analysed in conjunction with multilevel modulation schemes in a Gaussian environment. We used square-shaped Quadrature Amplitude Modulation (QAM) constellations [61]. Figure 2.41 portrays the location of each constellation point in terms of their in-phase (I) and quadrature-phase (Q) components for 2-, 4-, 16- and 64-QAM. Each constellation point is assigned a bit sequence. Gray coding is applied to assign the bit sequences to their respective constellation points, ensuring that the nearest-neighbour constellation points had a Hamming distance of one. Therefore the assignment of constellation points is optimised in terms of minimising the BER. For a more in-depth understanding of QAM techniques, the interested reader is referred to [61].

We use a RBF DFE for multilevel modems as discussed in Section 2.9.2 and Section 2.11. Figure 2.42 shows the bit error rate performance for the 2-, 4-, 16- and 64-QAM schemes in conjunction with correct and detected symbol feed-back. The performance degradation due to decision errors is approximately 0.5dB for 2- and 4-QAM, 1dB for 16dB, 1.5dB for 64dB at BER $= 10^{-4}$ and thus it has a moderate effect at low BERs. Note however the E_b/N_0 degradation increases, as the BER increases, which becomes more significant at higher order QAM.

Figure 2.43 shows the performance comparison between the conventional DFE and the RBF equaliser with decision feedback over the dispersive two-path Gaussian channel of Figure 2.21(a). The parameters of the conventional DFE were chosen such that it exhibited the best possible performance for our simulation scenario and hence a further increase of the feedforward order would not give a significant performance improvement. The conventional DFE used in our simulations had a feedforward order of $m = 7$, feedback order of $n = 1$ and decision delay of $\tau = 7$ symbols. The RBF equaliser using decision feedback was found to give a similar performance with a reduced feedforward order of 2, feedback order of 1 and decision delay of 1 symbol. The performance of the RBF assisted decision feedback equaliser can still be further improved quite significantly by increasing both the decision delay τ and the feedforward order m, as we discussed in Section 2.11 and this was demonstrated in Figure 2.23 for Binary Phase Shift Keying (BPSK).



(a) BPSK (b) QPSK (c) 16-QAM

(d) 64-QAM

——— DECISION BOUNDARIES

Figure 2.41: QAM Phasor Constellations

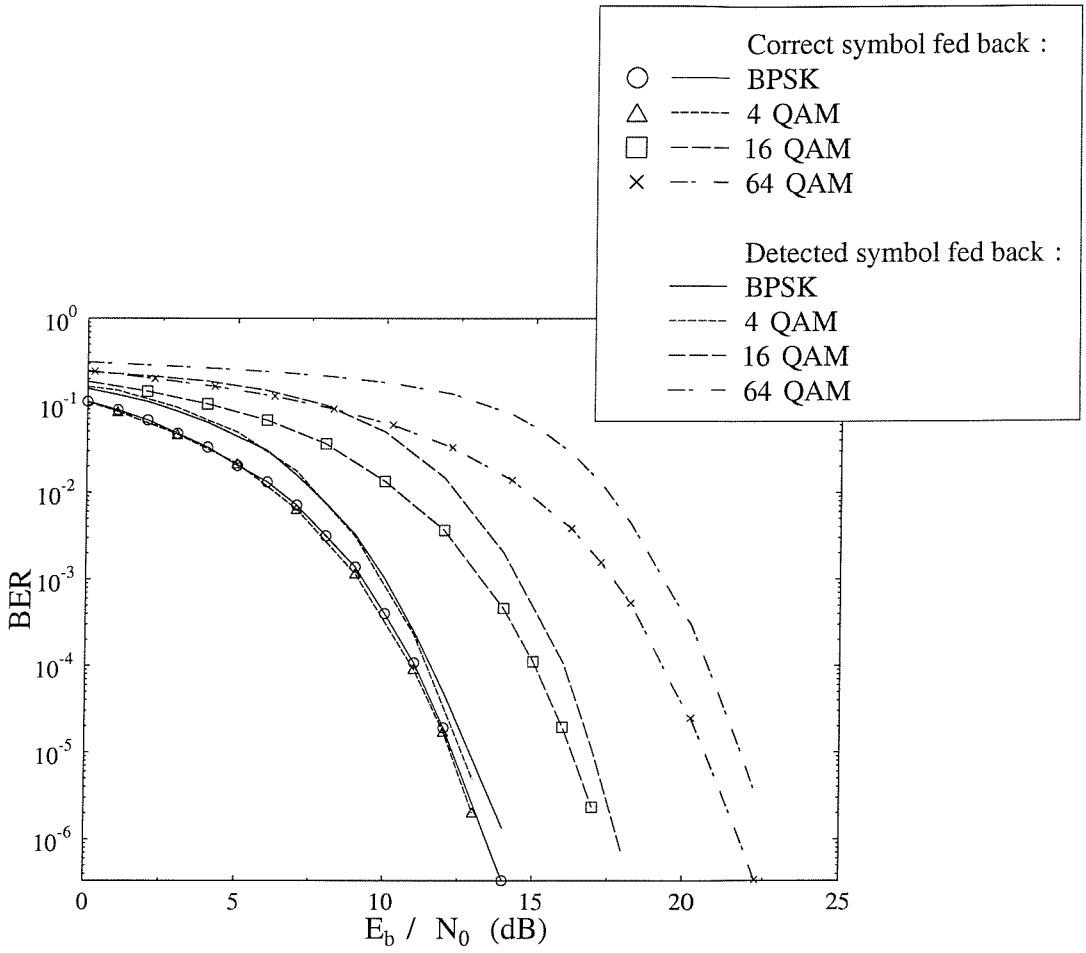


Figure 2.42: BER versus E_b/N_0 performance of the **RBF equaliser** using decision feedback over the dispersive two-path AWGN channel for different $\mathcal{M}$-QAM schemes. The impulse response of the two-path channel is described by Figure 2.21(a). The equaliser had a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol.

2.12.4 Performance of the RBF Equaliser over Wideband Rayleigh Fading Channels

In this section we used $\mathcal{M}$-QAM symbols. The combined transfer function of the transmitter and receiver filters yielded a raised cosine filter with a roll-off factor of 0.5. The transmitter and receiver filters were identical and were implemented as finite-impulse-response (FIR) filters. The filter tap weights were samples of the truncated square-root-raised-cosine impulse response. The transmitted symbol was oversampled by a factor of 8 and it was pulse-shaped. The baseband time-invariant multipath fading channel was represented as follows:

$$c(t) = \sum_{i=0}^{n_c} f_i(t)\delta_{t-\tau_i(t)}, \tag{2.126}$$

where n_c is the number of fading paths, $f_i(t)$ is the complex-valued ith CIR tap at time t, $\tau_i(t)$ is the excess delay at time t and δ_t is a delta function located at signalling instant t. The multipath components $f_i(t)$ have independent Rayleigh fading statistics, they are uncorrelated and are scaled by their designed weights. For a more in-depth charaterization of Rayleigh fading channels, the reader is referred to the tutorial by Sklar [77]. In our simulations, the fading parameters of the channel are given in Table 2.12 and we employ two symbol-spaced fading paths with the weights given by $0.707 + 0.707z^{-1}$. The structure of the transmitted burst is given in Figure 2.44, where the training symbol sequence is implemented as a preamble. In our simulations, the number of training symbols L_T was set to 27 and the number of data symbols L_D was set to 144.

Transmission Frequency	1800MHz
Transmission Rate	133kBds
Vehicular Speed	30 mph
Normalised Doppler Frequency	6×10^{-4}

Table 2.12: Simulation parameter of the Rayleigh fading channel

Figure 2.45 provides our BER performance comparison between the conventional DFE and the RBF DFE for different $\mathcal{M}$-QAM schemes. The conventional DFE assumed perfect channel estimation and its equaliser coefficients were optimised using the MSE criterion as described in [59] (pp. 607-612). The centres of the RBF DFE were positioned at the desired channel states. In these simulations the CIR taps were kept constant for the duration of the transmitted burst and were faded before the next burst, which we refer to as burst-invariant fading. From Figure 2.45 we note that for BPSK, the RBF DFE having a low feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol was found to give similar performance to the conventional DFE having a feedforward order

of $m = 7$, feedback order of $n = 1$ and decision delay of $\tau = 7$ symbols. For 4-QAM, 16-QAM and 64-QAM, the RBF DFE having the same parameters gives inferior performance compared to the conventional DFE in the two-path Rayleigh fading channel scenario. This is dissimilar to the performance of the two-path Gaussian channel shown in Figure 2.43. The performance degradations endured by the higher order modulation schemes are higher under fading channel conditions even in conjunction with perfect channel estimation, since these schemes are more sensitive to fades due to their reduced Euclidean distance between their neighbouring constellation points. Nevertheless, the performance of the RBF DFE can be improved by increasing both the decision delay τ and the feedforward order m, as we discussed in Section 2.11, at the expense of increased computational complexity. This is demonstrated in Figure 2.46, where the performance of the RBF DFE having an increased decision delay of $\tau = 2$ and corresponding feedforward order of $m = 3$ and $n = 1$ showed an improved performance, attaining similar BER performance curves to the previously described conventional DFE for BPSK, 4-QAM and 16-QAM. The performance of 64-QAM is not shown here due to the associated high computational complexity of the simulation.

The adaptive performance of the RBF DFE was investigated over the two-path Rayleigh fading channel at a normalised Doppler frequency of 6×10^{-4} for the BPSK modulation scheme. In our adaptive RBF DFE simulations, we used a variable centre learning rate μ_r, where we had $\mu_r = 0.3$ during the training mode and $\mu_r = 0.1$ during the decision-directed learning mode. We assigned a sequence of L_T pseudo-random binary symbols as the training symbol sequence as seen in Figure 2.44. We note, however that we will have to find the symbol sequence that can give the best training performance. Figure 2.47 provides our performance comparison for the RBF DFE using perfect channel estimation and when the adaptive RBF DFE is trained with the aid of the scalar centre clustering algorithm described in Section 2.10. Figure 2.47 shows that there is a high performance degradation due to the imperfect CIR knowledge and a residual BER is experienced in our simulations, when the wideband fading channel is symbol-invariant, as opposed to being burst-invariant, i.e. when it is kept invariant for only a symbol duration rather than for a burst duration. This phenomenon can be explained by comparing Figure 2.48 and 2.49 with Figure 2.50 and 2.51 that show the snapshots of the channel output vector $\tilde{\mathbf{v}}_k$, and that of the learnt and ideal channel states $\mathbf{r}_{j,i}$, when the feedback state is $\mathbf{s}_{f,j} = [-1]$ for the SNR of 30dB. The fades are symbol-invariant for Figure 2.48 and 2.49, and burst-invariant for Figure 2.50 and 2.51 throughout the transmission frame of $L_T + L_D = 177$ symbols. The ideal channel states were obtained from the taps of the impulse response of the channel at the start of the frame and the learnt channel states were obtained using the scalar centre clustering algorithm described in Section 2.10. We also observed from Figure 2.48 and 2.49 as well as from Figure 2.50 and 2.51 that the scalar clustering algorithm is capable of tracking the desired

channel states. At high SNRs the fades dominate, rather than the Gaussian noise, resulting in error statistics, which are not Gaussian. When the fading of the CIR is symbol-invariant, the effect of the fades is evident throughout the whole transmission burst, as we can observe from Figure 2.48 and 2.49 and this degrades the BER performance and gives an increased residual BER. But if the fade is burst-invariant, the channel effects due to fades will not be evident throughout the whole transmission frame, as shown in Figure 2.50 and 2.51, and thus this effect will not manifest itself in the results. The degradation of the BER performance due to fades within the transmission burst is evident in Figure 2.48(b). Theoretically, the channel output vectors are separable at any time instance due to the appropriate setting of the equaliser's parameters. However, the clustering algorithm tracking error and the small Euclidean distance between the channel states rendered the channel output vectors inseparable for the symbol-invariant fading scenario.

The channel output vectors are separable however for burst-invariant fading, as shown in Figure 2.50(b). During our simulations, the RBF DFE produced 77 symbol errors out of 144 data symbols in the frame of Figure 2.48(b) for the symbol-invariant fading scenario, but did not give any symbol errors in the frame of Figure 2.50(b) for the burst-invariant fading scenario. The inseparable channel output vectors explain the residual BER present during our symbol-invariant fading simulations, as shown in Figure 2.47. We note that even for relatively slow fading channels, the channel states can change significantly from symbol-to-symbol within a transmission burst duration. This phenomenon was noted in our simulations. Hence, when we assume perfect channel estimation and burst-invariant fading in our simulations, the results constitute best-case estimates.

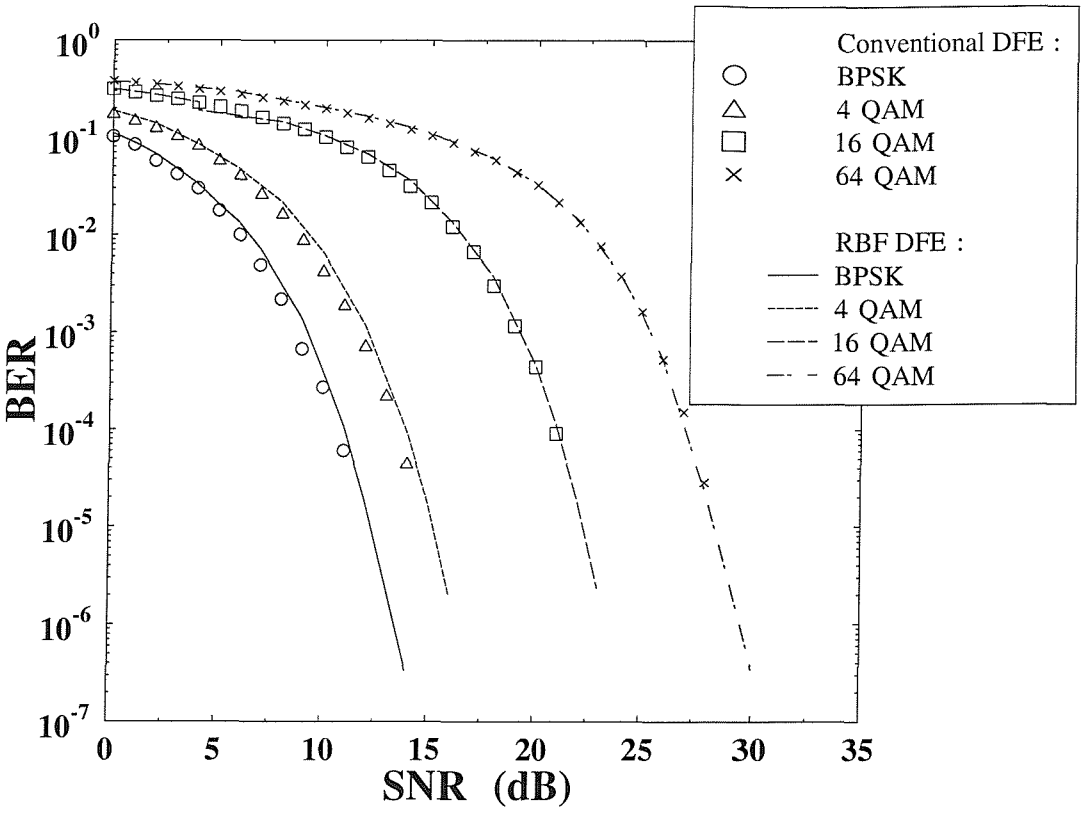


Figure 2.43: BER versus signal to noise ratio performance of the **RBF equaliser** using decision feedback and the **conventional DFE** over the dispersive two-path Gaussian channel for different $\mathcal{M}$-QAM schemes. The impulse response of the two-path channel is described by Figure 2.21(a). The RBF equaliser had a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol. The conventional DFE had a feedforward order of $m = 7$, feedback order of $n = 1$ and decision delay of $\tau = 7$ symbols. Correct symbols were fed back.

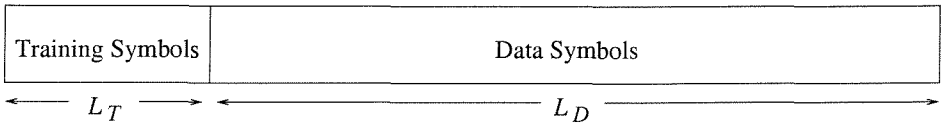


Figure 2.44: Transmitted frame structure depicting the position of the data and training symbols. For example in the context of the COST 207 CIR of Figure 2.52 the number of training symbols L_T were 49 and the number of data symbols L_D were 122.

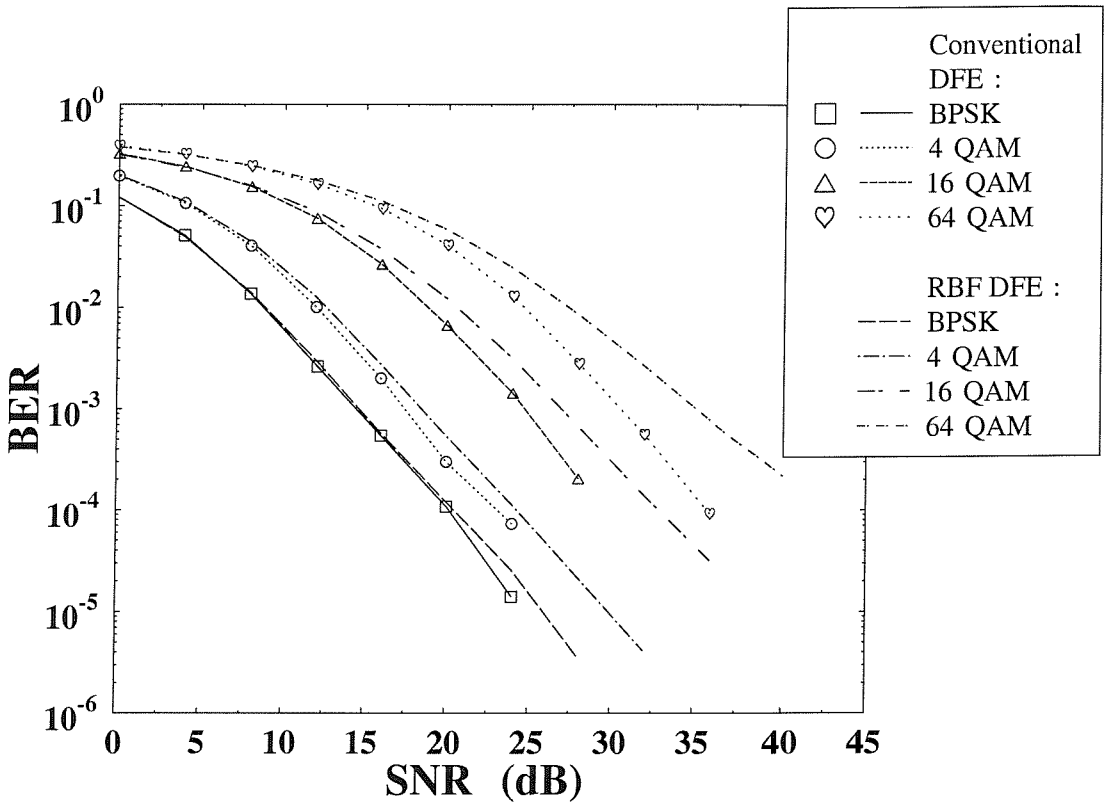


Figure 2.45: BER versus signal to noise ratio performance of the conventional DFE and the RBF equaliser with decision feedback over the two equal weight, symbol-spaced path Rayleigh fading channel of $F(z) = 0.707 + 0.707z^{-1}$ for different $\mathcal{M}$-QAM schemes. Both equalisers assume perfect CIR estimation. The conventional DFE had a feedforward order of $m = 7$, feedback order of $n = 1$ and decision delay of $\tau = 7$ symbols. The RBF DFE had a **feedforward order of** $m = 2$, feedback order of $n = 1$ and **decision delay of** $\tau = 1$ symbol. Correct symbols were fed back. The Rayleigh fading parameters are summarised in Table 2.12.

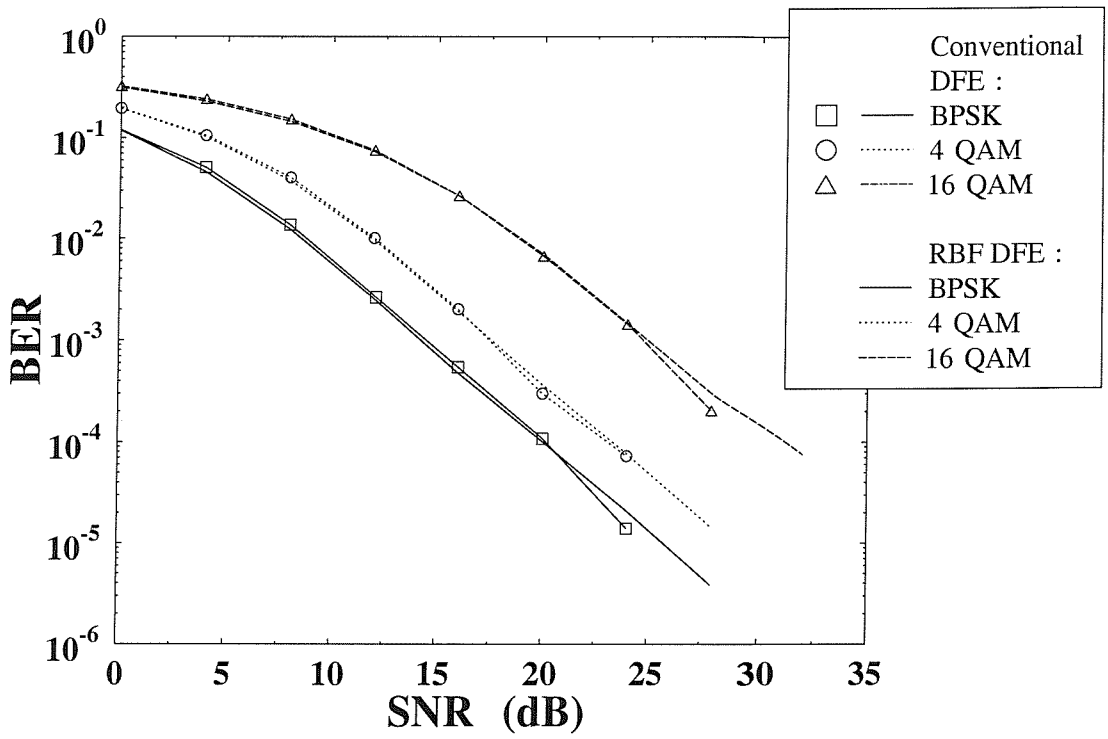


Figure 2.46: BER versus signal to noise ratio performance of the conventional DFE and the RBF equaliser with decision feedback over the two equal weight, symbol-spaced path Rayleigh fading channel of $F(z) = 0.707 + 0.707z^{-1}$ for different $\mathcal{M}$-QAM schemes. Both equalisers assume perfect CIR estimation. The conventional DFE had a feedforward order of $m = 7$, feedback order of $n = 1$ and decision delay of $\tau = 7$ symbols. The RBF DFE had a **feedforward order of** $m = 3$, feedback order of $n = 1$ and **decision delay of** $\tau = 2$ symbol. Correct symbols were fed back. The Rayleigh fading parameters are summarised in Table 2.12.

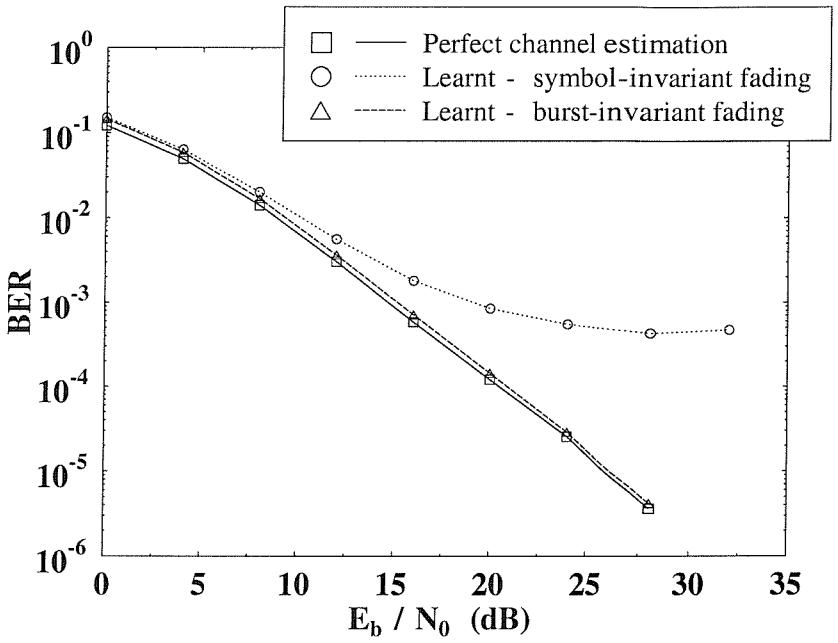
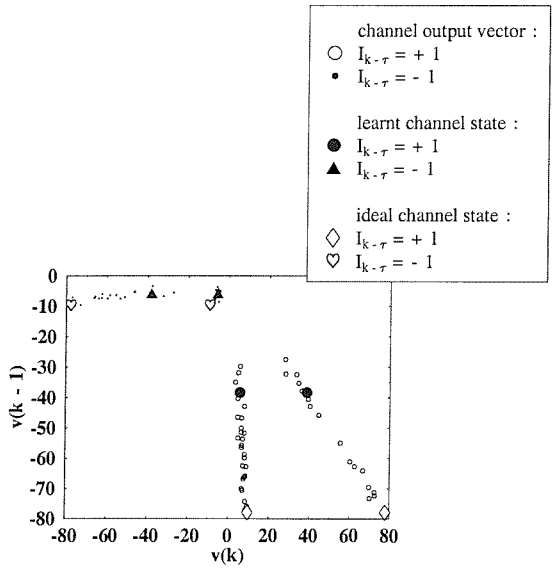
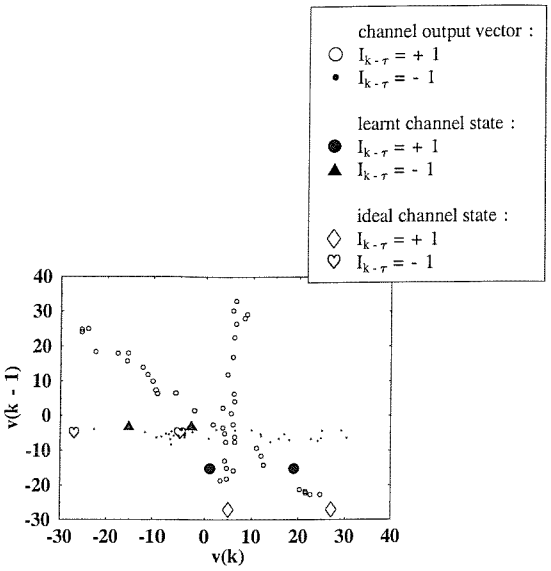


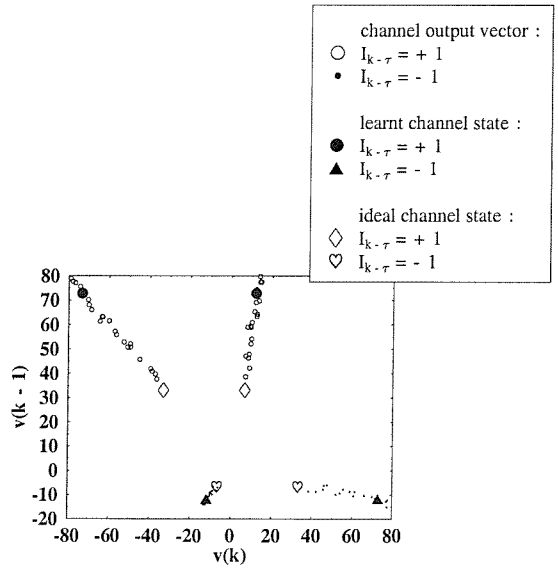
Figure 2.47: BER versus E_b/N_0 performance of the adaptive RBF DFE with correct decision fedback under burst-invariant fading and symbol-invariant fading. The RBF DFE is adapted using the scalar centre clustering algorithm described in Section 2.10. The performance of the RBF DFE using perfect CIR estimation is provided for comparison. The RBF DFE had a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol. The Rayleigh fading parameters are summarised in Table 2.12.



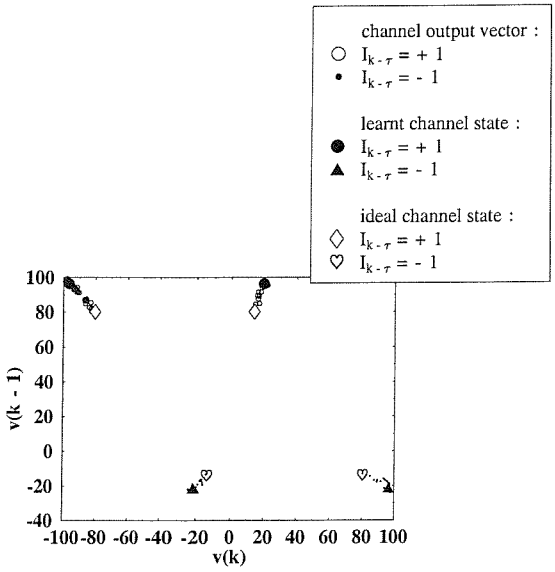
(a) **1st** frame burst (Number of errors = 0)



(b) **2nd** frame burst (Number of errors = 77)

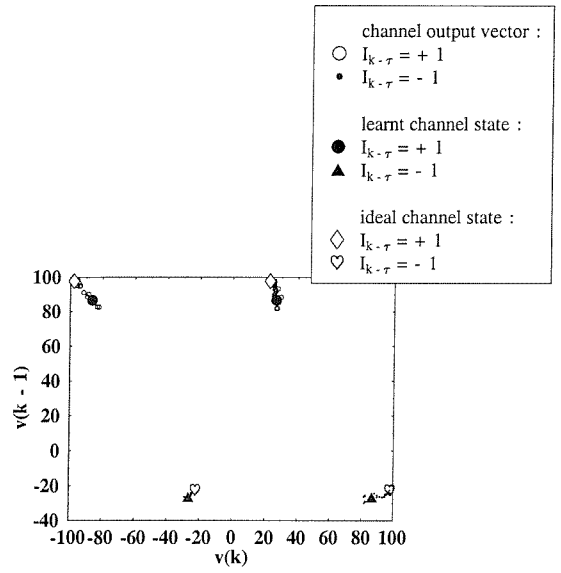


(c) **3rd** frame burst (Number of errors = 0)

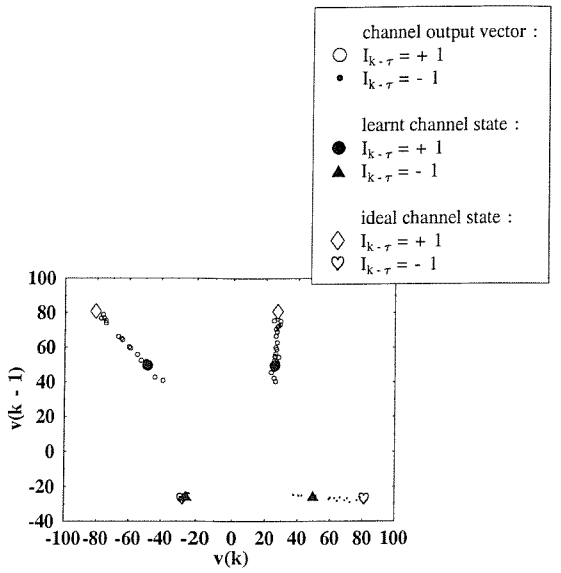


(d) **4th** frame burst (Number of errors = 0)

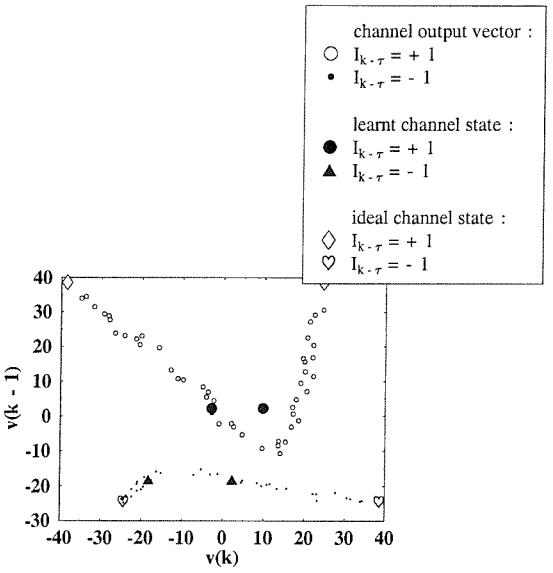
Figure 2.48: The channel output vectors, learnt channel output states and ideal channel output states of 1–4 transmission bursts in two-dimensional observation space, when the feedback symbol is -1 over the two-path symbol-spaced, equal-gain Rayleigh fading channel. **The fading is symbol-invariant.** The DFE had the parameters of $m = 2, n = 1$ and $\tau = 1$, while the SNR was 30dB.



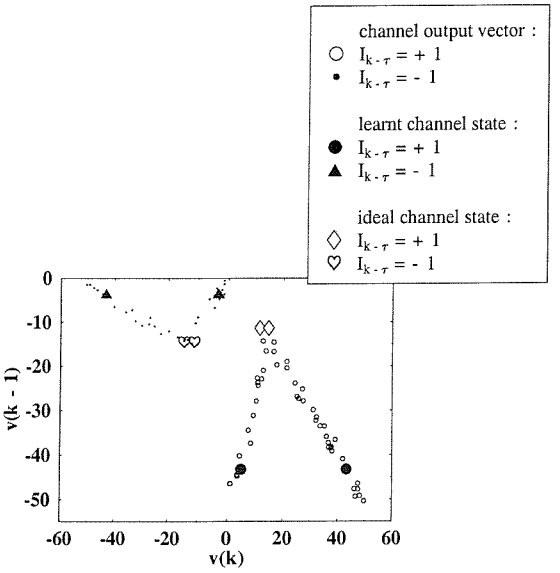
(a) **5th** frame burst (Number of errors = 0)



(b) **6th** frame burst (Number of errors = 0)

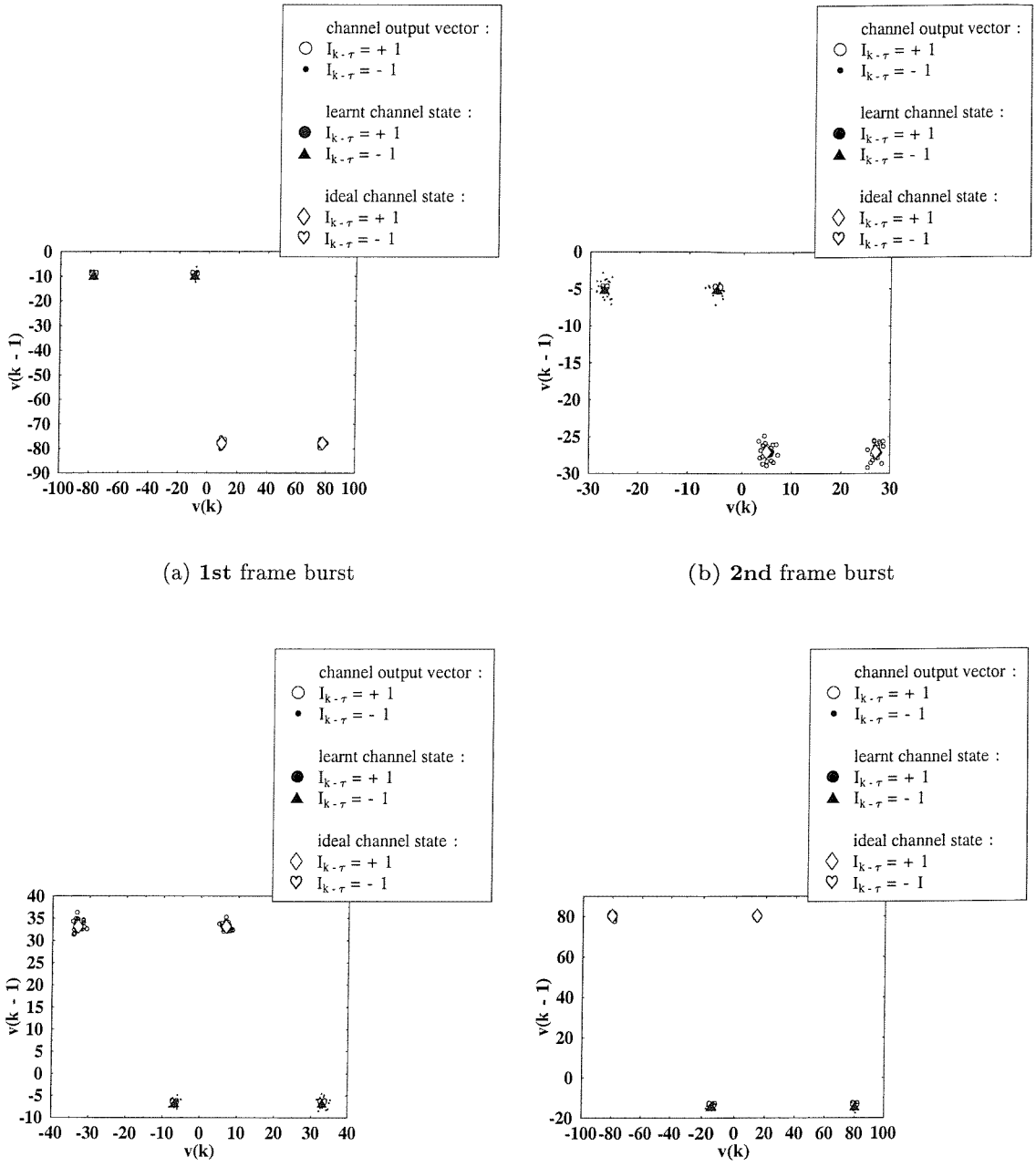


(c) **7th** frame burst (Number of errors = 4)



(d) **8th** frame burst (Number of errors = 0)

Figure 2.49: The channel output vectors, learnt channel output states and ideal channel output states of 5–8 transmission bursts in two-dimensional observation space, when the feedback symbol is -1 over the two-path symbol-spaced, equal-gain Rayleigh fading channel. **The fading is symbol-invariant**. The DFE had the parameters of $m = 2, n = 1$ and $\tau = 1$, while the SNR was 30dB.



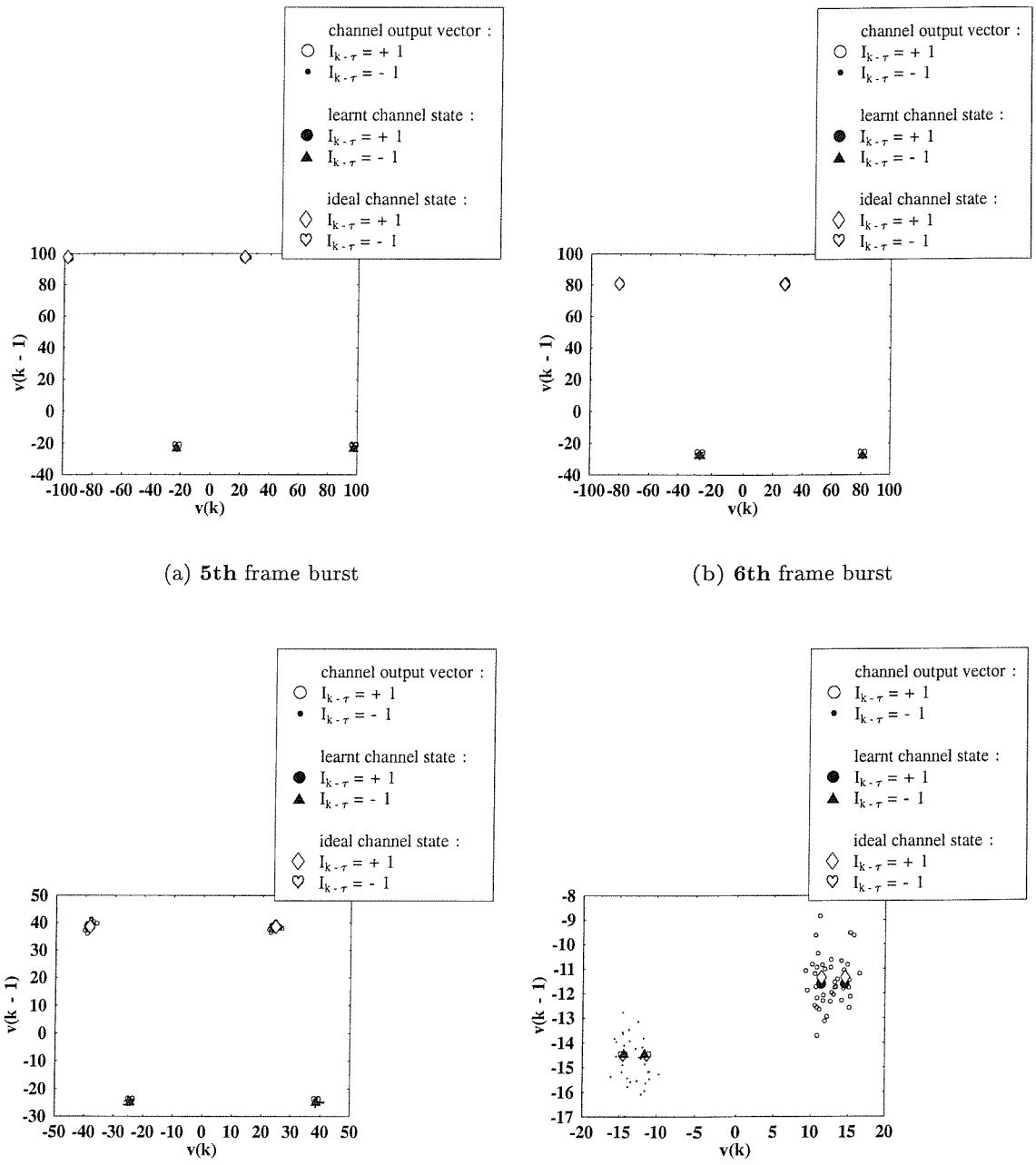
(a) **1st** frame burst

(b) **2nd** frame burst

(c) **3rd** frame burst

(d) **4th** frame burst

Figure 2.50: The channel output vectors, learnt channel output states and ideal channel output states of 1–4 transmission bursts in two-dimensional observation space when the feedback symbol is -1 over the two-path symbol-spaced, equal-gain Rayleigh fading channel. **The fading is burst-invariant**. The DFE had the parameters of $m = 2, n = 1$ and $\tau = 1$, while the SNR was 30dB.



(a) **5th** frame burst

(b) **6th** frame burst

(c) **7th** frame burst

(d) **8th** frame burst

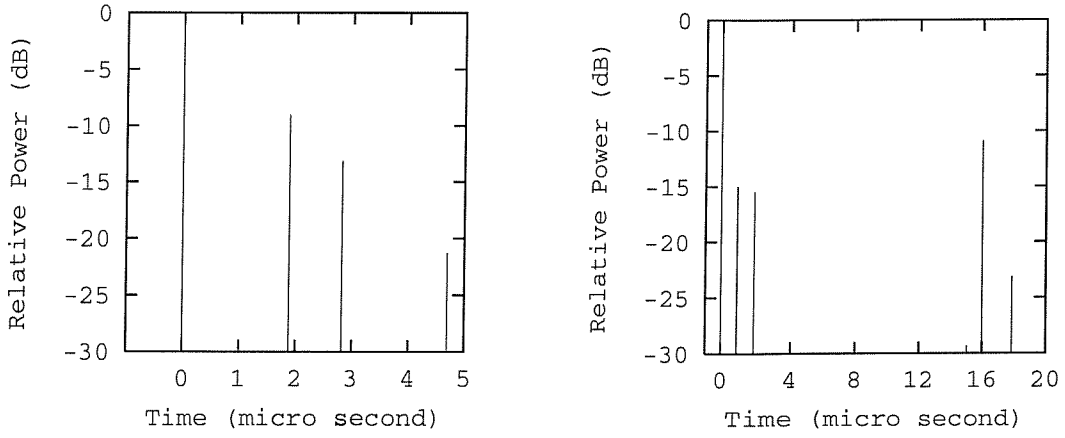
Figure 2.51: The channel output vectors, learnt channel output states and ideal channel output states in two-dimensional observation space when the feedback symbol is -1 over the two-path symbol-spaced, equal-gain Rayleigh fading channel. **The fade is burst-invariant**. The DFE had the parameters of $m = 2, n = 1$ and $\tau = 1$, and the SNR was 30dB.

2.12.5 Performance of the RBF DFE over COST 207 Channels

In this section the performance of the RBF DFE is investigated over the widely-used family of Rayleigh fading COST 207 test channels [78]. The magnitude of the impulse responses and their respective delays can be calculated by applying a set of rules, which is specified in the COST 207 report [78]. More specifically, the CIR taps may be positioned on an equispaced legitimate raster, provided the taps themselves are not equispaced. The impulse responses and the relative delays of the channels referred to as the Typical Urban (TU) and Hilly Terrain (HT) models are shown in Figure 2.52 and Table 2.13. Figure 2.53 shows the observed channel output and the learnt channel states in a two dimensional $\begin{bmatrix} v_k & v_{k-1} \end{bmatrix}$ space, when the decision delay is one symbol for the AWGN contaminated dispersive TU and HT channels without fading. Note in Figure 2.53(b) that the channel states are separable without fading. The fading parameters used in our simulations were given in Table 2.12. The structure of the transmitted burst was given in Figure 2.44. In our simulations, the number of training symbols L_T was set to 49 and the number of data symbols L_D was set to 122. The scalar centre clustering algorithm of Equation 2.100 was used in conjunction with a variable centre learning rate such that μ_r was 0.3 during the training mode and 0.1 during the decision-directed learning mode. Figure 2.54 shows the BER versus E_b/N_0 performance for the COST 207 TU and HT channels, where the RBF DFE parameters were set to be $m = 2, n = 1, \tau = 1$ for the TU channel and $m = 3, n = 2, \tau = 2$ for the HT channel, so that the decision delay covered the whole impulse response length. Thus, we assumed $L = 1$ for the TU channel and $L = 2$ for the HT channel. Figure 2.54 depicts the BER performance for both symbol-invariant and burst-invariant fading burst. From Figure 2.54 we observed again the residual BER of approximately 10^{-3} due to the desired channel states that are close together in terms of Euclidean distance, which is a consequence of the non-ideal learnt channel states and inseparable channel state clusters in the symbol-invariant scenario, as mentioned in Section 2.12.4. For the burst-invariant scenario, the residual BER was approximately 2×10^{-5} for TU channel and 5×10^{-6} for the HT channel, where again, the explanation of Section 2.12.4 applies. However, for the burst-invariant scenario, where the noiseless channel states remain the same throughout the burst duration, the performance degradation that produces the residual BER is due to the non-ideal learnt channel states, especially when the desired channel states are close together.

Hilly Terrain		Typical Urban	
Position (μs)	Relative Power (dB)	Position (μs)	Relative Power (dB)
0.00	-0.7	0.00	-0.87
0.94	-14.99	1.88	-9.03
1.88	-15.44	2.82	-13.12
15.04	-29.30	4.70	-21.28
15.98	-10.89		
17.86	-23.13		

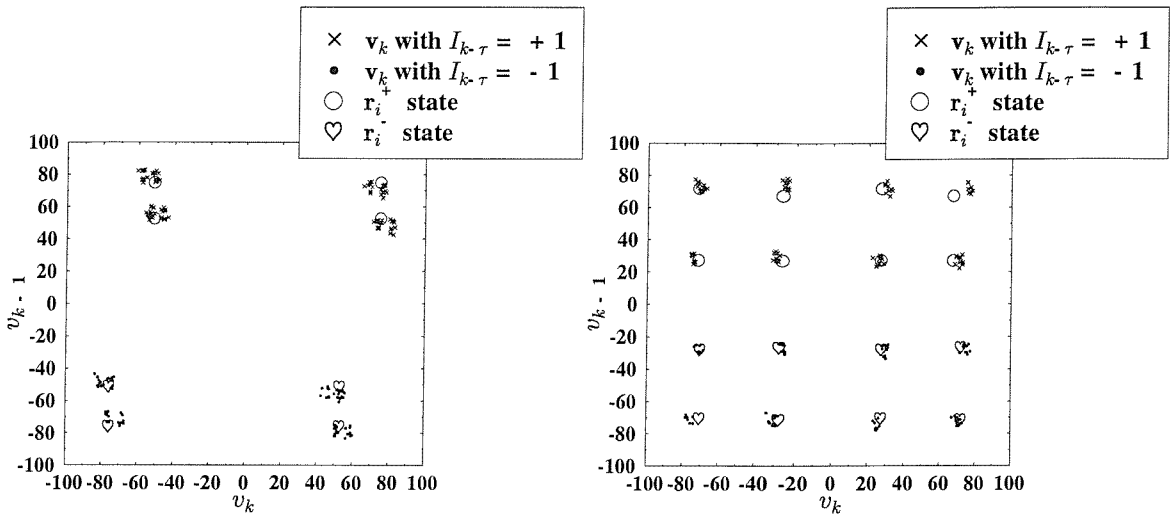
Table 2.13: The relative power and delay of each path in the COST 207 [78] Typical Urban and Hilly Terrain channels which are depicted in Figure 2.52.



(a) Typical Urban Channel

(b) Hilly Terrain Channel

Figure 2.52: The impulse response of the COST 207 Typical Urban and Hilly Terrain channels depicting the relative power of each impulse and their relative delays, as shown in Table 2.13



(a) COST 207 Typical Urban Channel

(b) COST 207 Hilly Terrain Channel

Figure 2.53: The noisy channel outputs $\mathbf{v}_k$ and the learnt channel states $\mathbf{r}_i$ of the COST 207 Typical Urban and Hilly Terrain channels with their transfer function depicted in Figure 2.52 for a BPSK modulation scheme. The SNR was 30dB, the number of samples was 171 and the decision delay was one symbol. The channel states $\mathbf{r}_i$ were learnt with the aid of the scalar clustering algorithm of Section 2.10, where the number of training symbols was 49 and the learning rate μ_r was set to 0.3 during training mode, while to 0.1 during decision-directed mode. By comparison, the corresponding quantities for the CIR of $F(z) = 0.707 + 0.707z^{-1}$ were plotted in Figures 2.50 and 2.51.

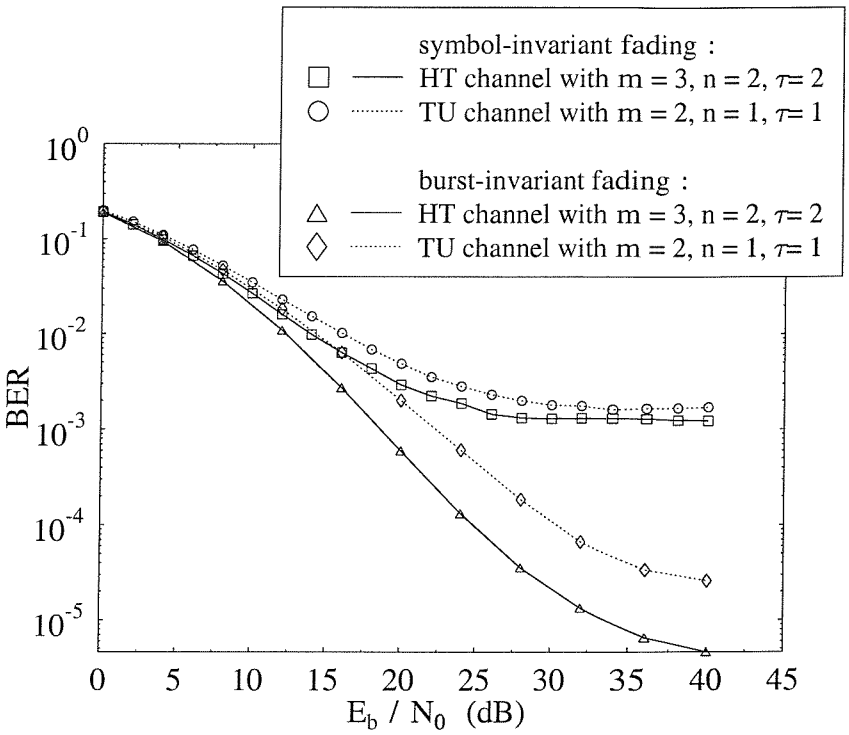


Figure 2.54: BER versus E_b/N_0 performance for the BPSK RBF DFE in conjunction with correct decision fedback over the COST 207 TU and HT channels with the impulse responses described by Table 2.13 and Figure 2.52.

2.13 Conclusions

In this chapter we provided a brief overview of neural networks and described, how equalisation can be viewed as a classification problem. The architecture of RBF networks was presented and we described the design of the RBF equaliser based on the Bayesian equaliser solution. Our performance comparisons between the linear MSE equaliser and RBF equaliser in Figure 2.28 and 2.29 demonstrated that the RBF equaliser is capable of providing superior performance with the aid of an equivalent equaliser order at the expense of an exponential complexity increment upon increasing the equaliser order. According to Figure 2.28 and 2.29, the RBF equaliser having a feedforward order of $m = 9$ provides a performance improvement of 10dB and 20dB over the linear MSE equaliser for the two-path and three-path Gaussian channel of Figure 2.21, respectively, at a BER of 10^{-3}. We note that both the linear MSE equaliser and the RBF equaliser exhibit a residual BER characteristic, if the channel states corresponding to different transmitted symbols are inseparable in the channel observation space, as shown in Figure 2.30.

The adaptive performance of the RBF equaliser employing the vector centre clustering algorithm of Section 2.9.5, scalar centre clustering algorithm of Section 2.10 and the LMS channel estimator of Section 2.9.4 were compared. The convergence rate of the clustering algorithm depends on the number of channel coefficients to be adapted and therefore depends on the modulation scheme used and on the CIR length. However, the convergence of the LMS channel estimation technique only depends on the CIR length and therefore this technique is preferred for high-order modulation schemes and high CIR lengths. This is particularly true for the scenario, where the modulation mode of the training sequence and the data sequence differs, e.g for the adaptive QAM system described in Chapter 3, where a more robust modulation mode is used for the training sequence. The LMS channel estimation technique could only be used to obtain the correponding RBF centres, since the desired channel output differs for the training- and data sequences. However, note that the LMS channel estimation technique incurs a higher computational complexity compared to both the vector and scalar clustering algorithms, as demonstrated in Table 2.7, 2.8 and 2.9.

Decision feedback was then introduced into the RBF equaliser, in order to reduce its computational complexity. As a result, its performance improved, since the Euclidean distance between the channel states corresponding to different transmitted symbols increased, when the DFE scheme was used. Recall that the parameters of the RBF DFE were chosen to be $m = \tau + 1$ and $n = L$, where m, n, τ and $L + 1$ are the feedforward order, feedback order, delay and CIR length, which provide the best solution for a fixed equaliser delay τ. As expected, the performance degradation due to decision error propagation increased, as the BER increased, which became more significant for higher order QAM, as it was shown in

Figure 2.42. For fading channel conditions, the performance degradation for higher order modulation schemes were higher, since they are more sensitive to fades due to the reduced Euclidean distance between the neighbouring channel states.

We investigated the performance of the adaptive RBF equaliser in symbol- and burst-invariant fading scenarios. We observed the effects of inseparable channel state clusters for symbol-invariant fading in Figure 2.48(b), which was due to the fast-fading effects present across the burst duration. This phenomena, together with the non-ideal learnt channel states, explain the residual BER present in our simulations. Therefore, we have to note that even for relatively slow fading channels, the channel states value can change significantly on a symbol-by-symbol basis in a transmission burst duration.

In the next chapter, we will proceed to investigate the implementation and performance of the RBF equaliser in the context of adaptive modulation schemes.

Chapter 3

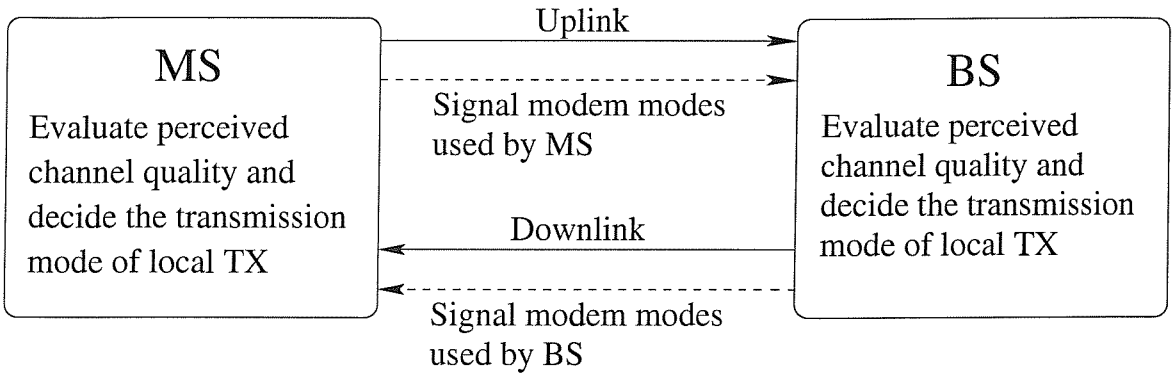
Adaptive Modulation

In this chapter, the concept of RBF equalisers is extended to Burst-by-Burst (BbB) Adaptive QAM (AQAM) schemes. BbB AQAM schemes employ a higher-order modulation mode in transmission bursts, when the channel quality is favourable, in order to increase the throughput and conversely, a more robust but lower-order modulation mode is utilized in those transmission bursts, where the instantaneous channel quality drops. The modem mode switching regime will be detailed in more depth during our further discourse. We will show that this RBF-AQAM scheme naturally lends itself to accurate channel quality estimation. We will provide an outline of our various assumptions and the description of the simulation model, leading to our RBF-AQAM performance studies. This scheme is shown to give a significant improvement in terms of the mean BER and bits per symbol (BPS) performance compared to that of the individual fixed modulation modes. Let us now commence with a brief background on adaptive modulation in both narrow- and wide-band fading channel environments.

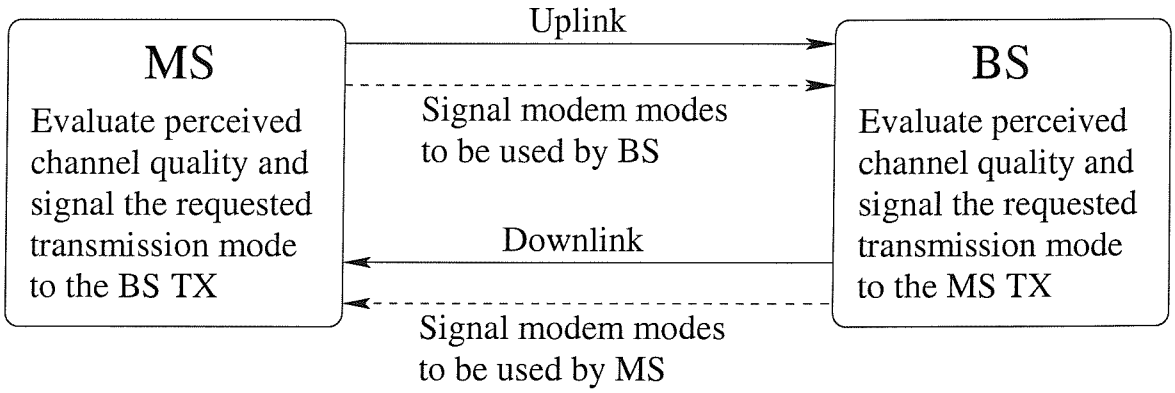
3.1 Background to Adaptive Modulation in a Narrowband Fading Channel

We summarise here the principles of *adaptive modulation* in a narrow-band Rayleigh fading channel environment. In a narrow-band channel, as a result of channel fading, the short-term SNR can be severely degraded. This typically degrades the short-term BER at the receiver. Again, the concept of adaptive modulation is to employ a higher modulation mode, when the channel quality is favourable, in order to increase the throughput and conversely, a more robust modulation mode is employed, in order to provide an acceptable BER, when the channel exhibits a deep fade. Thus, adaptive modulation is not only used to combat the fading effects of a narrow-band channel, but it also attempt to maximise the throughput.

This idea is somewhat reminiscent of invoking a coarse power control scheme although without the detrimental effects of inflicting increased interferences upon other system users due to powering up during the intervals of low channel quality. In our work we used a variable number of modulation levels and again, we refer to this scheme as AQAM, while maintaining a constant transmitted power.



(a) Open-loop based signalling



(b) Close-loop based signalling

Figure 3.1: Closed- and open-loop signalling regimes for AQAM, where BS represents the Base Station, MS denotes the Mobile Station and the transmitter is represented by TX.

Adaptive modulation can only be invoked in the context of duplex transmissions, since some method of informing the transmitter of the quality of the link as perceived by the receiver is required unless an explicit feedback control channel is provided by the system. More explicitly, in adapting the modulation mode, a signalling regime has to be implemented in order to harmonise the operation of the transmitter and receiver with regards to the adaptive modem mode parameters. The range of signalling options is summariszed

in Figure 3.1 for both so-called open-loop and closed-loop signalling. For example, adaptive modulation can be applied in a time division duplex (TDD) arrangement, where the uplink and downlink transmissions are time-multiplexed onto the same carrier as depicted in Figure 3.2. If the channel quality of the uplink and downlink can be considered similar, an open-loop signalling system can be implemented, where the modulation mode can be adapted at the transmitter based on the information about the channel quality acquired during its receiving mode. This open-loop system is portrayed in Figure 3.1(a). The specific modem mode invoked has to be explicitly signalled by the transmitter to the receiver along with the reverse-direction information and it must be strongly protected against transmission errors, in order to avoid catastrophic BER degradations in case of modem mode signalling errors. By contrast, if the above channel quality predicability is not applicable – for example due to the presence of co-channel interference, etc. – the closed-loop based signalling system shown in Figure 3.1(b) can be implemented. This would be typical in a frequency division duplex (FDD) based system, where the uplink and downlink transmission frequency bands are different. Explicitly, the receiver has to instruct the remote transmitter concerning the modem mode to be used for meeting the receiver's target integrity requirements. The modem mode side-information signalling requirement is the same for both of the above signalling scenarios. For example, two bits per transmission burst are required to signal four different modem modes. However, the channel quality information will be based on a more obsolete channel quality estimate in the dissimilar uplink/downlink scenario, when the receiver instructs the remote transceiver concerning the modem mode to be used for meeting the receiver's BER target. It was shown in the context of a Kalman-filtered DFE block turbo coded AQAM scheme that it is feasible to refrain from explicitly signalling the modem modes upon invoking blind mode detection and hence increase the associated throughput [79].

Having discussed briefly the principle of adaptive modulation and the associated scenarios, where it can be applied, we can now explore the methology used for choosing the appropriate number of modulation levels.

Torrance [80] used the instantaneous received power as the channel quality measure. The estimated instantaneous received power was used to select the suitable modulation mode by comparing the received power against a set of switching thresholds, $l_n, n = 1, \ldots, 4$, as depicted in Figure 3.3. These switching thresholds govern the tradeoff between the mean BER and the BPS performance of the system. If low switching thresholds are used, the probability of employing a high-order modulation mode increases, thus yielding a better BPS performance. Conversely, if high switching thresholds are used, a low-order modulation mode is employed more frequently, resulting in an improved mean BER performance. In his efforts to derive upper-bound performance bounds Torrance [80] assumed perfect channel

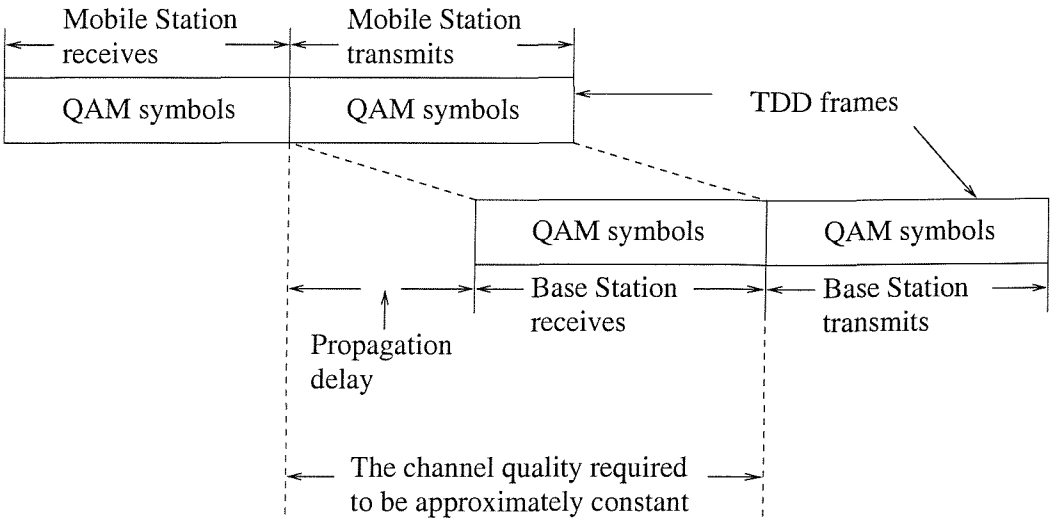


Figure 3.2: The TDD framing structure used in our AQAM system

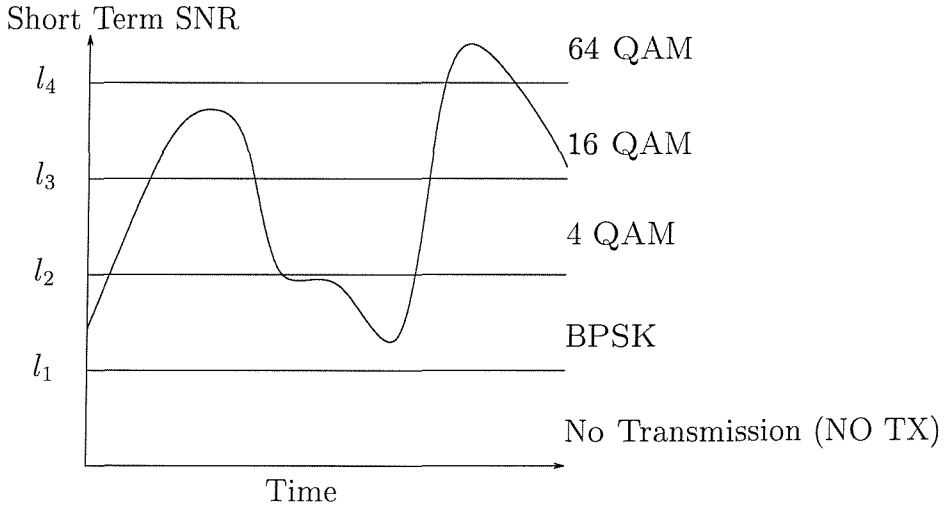


Figure 3.3: Stylised profile of the short-term received SNR, which is used to choose the next modulation mode of the transmitter in TDD mode.

quality estimation and compensation, perfect knowledge of the modulation mode at the receiver and perfect estimation of the expected received power prior to transmission.

Webb and Steele [81] used the received signal strength and the BER as channel quality measures in a flat Rayleigh-fading environment. The signal to co-channel interference ratio and the expected delay spread of the channel was used by Sampei, Komaki and Morinaga [82] as the criteria to switch amongst the modulation modes and the legitimate modulation rates. They used $\frac{1}{4}$-rate QPSK, $\frac{1}{2}$-rate QPSK, QPSK, 16-QAM, 64-QAM in a narrow-band channel environment. Sampei, Morinaga and Hamaguchi utilised the signal to noise ratio and the normalised delay spread as the channel quality measure.

For a review of other work that has been conducted using adaptive modulation, the reader is referred to Wong's thesis [83].

3.2 Background on Adaptive Modulation in a Wideband Fading Channel

In this section we will initially extend the AQAM concept to wideband fading channel environments by employing conventional channel equalisation. We will briefly summarise, how the performance of the equaliser and the AQAM scheme can be jointly optimized.

As expected, the AQAM switching criteria of the narrow-band scenario mentioned in Section 3.1 has to be modified for the wideband channel environment. In Torrance's paper [80] for example, the quality of the channel was determined on the basis of the short-term SNR, which was then used as a metric in order to choose the appropriate modulation mode for the transmitter. However, in a wideband environment, the SNR metric is not reliable in quantifying the quality of the channel, where the existence of the multipath components in the wideband channel produces not only power attenuation of the transmission burst, but also intersymbol interference, as discussed in Section 2.1. Even when the channel SNR is high, QAM transmissions over wideband Rayleigh fading channels are subjected to error bursts due to ISI. Consequently, the metric required to quantify the channel quality has to be redefined, in order to incorporate the effects of the wideband channel.

Wong and Hanzo [83, 84] approached this problem by formulating a two-step methology to mitigate the effects of the dispersive wideband channel. The first step employed a conventional Kalman-filtering based DFE, in order to eliminate most of the ISI. In the second step, the signal to noise plus residual interference ratio at the output of the equaliser was calculated based on the channel estimate. This ratio was referred to as the *pseudo SNR*, since it exhibited a Gaussian-like distribution and it was used as a metric to switch the modulation mode. Again, in [83, 84], Wong used the conventional Kalman-filtering based DFE depicted in Figure 3.4. If the ISI due to past detected symbols is eliminated by the

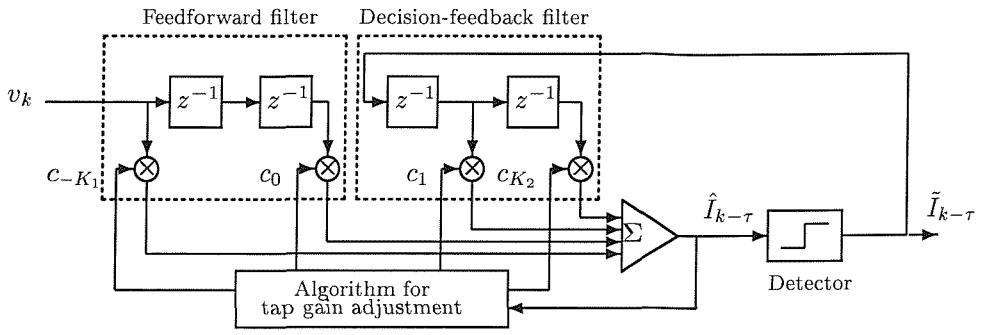


Figure 3.4: Decision-feedback equaliser schematic

feedback filter, then the wanted signal power, the residual ISI signal power and the effective noise power can be expressed as follows [85]:

$$\text{Wanted Signal Power} = E\left[|q_0 I_n|^2\right], \tag{3.1}$$

$$\text{Residual ISI Signal} = \sum_{k=-K_1}^{-1} E\left[|q_k I_{n-k}|^2\right], \tag{3.2}$$

$$\text{Effective Noise Power} = N_0 \sum_{j=-K_1}^{0} |c_j|^2, \tag{3.3}$$

$$n = -\infty, \ldots, \infty, \tag{3.4}$$

where $q_k = \sum_{j=-K_1}^{0} c_j f_{k-j}$, $c_j, j = -K_1, \ldots, 0$ are the feedforward tap coefficients, $c_j, j = 1, \ldots, K_2$ are the feedback tap coefficients, f_k is the kth impulse response tap of the channel and N_0 is the noise power. Therefore, the pseudo SNR output of the DFE, γ_{DFE}, can be calculated as follows:

$$\gamma_{\text{DFE}} = \frac{E\left[|q_0 I_n|^2\right]}{\sum_{k=-K_1}^{-1} E\left[|q_k I_{n-k}|^2\right] + N_0 \sum_{j=-K_1}^{0} |c_j|^2}. \tag{3.5}$$

The calculated pseudo SNR output of the DFE, γ_{DFE}, is then compared against a set of switching threshold levels, l_n, stored in a lookup table. The pseudo SNR output of the DFE, γ_{DFE}, is used for invoking the appropriate modem mode as follows [83]:

$$\text{Modulation Mode} = \begin{cases} \text{NO TX} & \text{if } \gamma_{DFE} < l_1 \\ \text{BPSK} & \text{if } l_1 < \gamma_{DFE} < l_2 \\ \text{4-QAM} & \text{if } l_2 < \gamma_{DFE} < l_3 \\ \text{16-QAM} & \text{if } l_3 < \gamma_{DFE} < l_4 \\ \text{64-QAM} & \text{if } \gamma_{DFE} > l_4, \end{cases} \tag{3.6}$$

where $l_n, n = 1, \ldots, 4$ are the pseudo-SNR thresholds levels, and Powell's Multi-dimensional Line Minimization technique [86] was used to optimize the switching levels l_n in [84].

In the forthcoming sections, instead of the conventional DFE, we will explore using the RBF network for the equalisation process, as described in Section 2.9. The joint adaptive

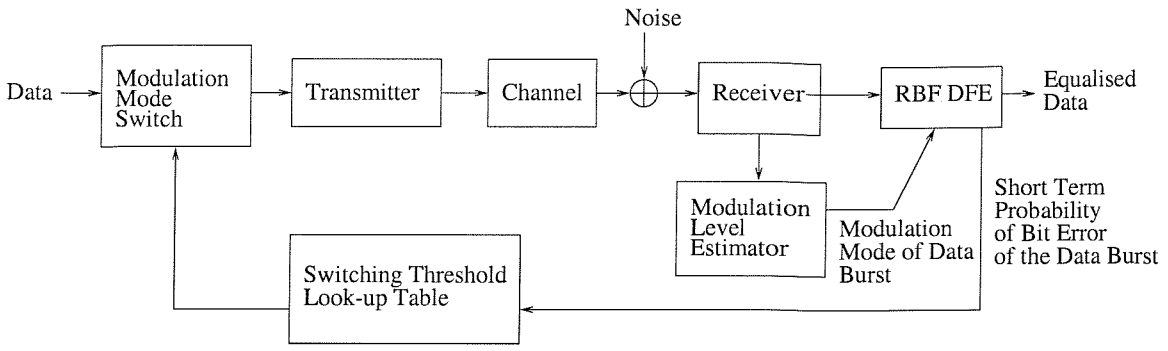


Figure 3.5: System schematic of the joint adaptive modulation and RBF equaliser scheme

modulation and RBF equalisation scheme will be described next, followed by our simulation results.

3.3 System Description of Joint Adaptive Modulation and RBF based Equalisers

In this section, we will describe the joint AQAM and RBF network based equalisation scheme and the switching metric employed. We commence by exploring the joint AQAM and RBF equaliser scheme's best-case performance. Finally, the performance of this scheme and that of the individual fixed modulation modes is compared in terms of their mean BER and BPS.

3.3.1 System Overview

The schematic of the joint AQAM and RBF network based equalisation scheme is depicted in Figure 3.5. We use the RBF DFE described in Section 2.11 in this scheme. At the receiver, the RBF DFE is trained using the method described in Section 2.9.3 and then the corrupted received signal is equalized. The short-term probability of bit error or short-term BER of the transmitted burst is calculated from the output of the sub-RBF network, and is used as the switching metric. Section 3.3.2 will highlight this issue in more detail. The short-term BER is compared to a set of switching BER values corresponding to the modulation mode of the received data burst. Consequently, a modulation mode is selected for the next transmission, assuming channel quality similarity for the uplink and downlink transmissions. This implies that the similarity of the short-term BER of consecutive uplink and downlink data bursts can be exploited, in order to set the next modulation mode. The modulation modes utilized in our system are BPSK, 4-QAM, 16-QAM, 64-QAM and no transmission (NO TX), similarly to Equation 3.6. Therefore, the modulation mode is

switched according to the estimated short-term BER, $P_{\text{bit, short-term}}$, as follows:

$$\text{Modulation Mode} = \begin{cases} \text{NO TX} & \text{if } P_{\text{bit, short-term}} \geq P_2^{\mathcal{M}} \\ \text{BPSK} & \text{if } P_2^{\mathcal{M}} > P_{\text{bit, short-term}} \geq P_4^{\mathcal{M}} \\ \text{4-QAM} & \text{if } P_4^{\mathcal{M}} > P_{\text{bit, short-term}} \geq P_{16}^{\mathcal{M}} \\ \text{16-QAM} & \text{if } P_{16}^{\mathcal{M}} > P_{\text{bit, short-term}} \geq P_{64}^{\mathcal{M}} \\ \text{64-QAM} & \text{if } P_{64}^{\mathcal{M}} > P_{\text{bit, short-term}}, \end{cases} \tag{3.7}$$

where $P_i^{\mathcal{M}}, i = 2, 4, 16, 64$ are the switching BER thresholds corresponding to the various $\mathcal{M}$-QAM modes.

3.3.2 Modem Mode Switching Metric

The RBF equaliser based on the optimal Bayesian decision function of Equation 2.18, as described in Chapter 2, is capable of providing the 'on-line' estimation of the BER in the receiver without the knowledge of the transmitted symbols. This is possible, since the equaliser is capable of estimating the *a posteriori* probability of the transmitted symbols, if the CIR is known and provided that the centres of the RBF network are assigned the values of the channel states, as it was originally suggested in Section 2.9.

Referring to Section 2.9.2 and Figure 2.19, the output of the RBF networks provides the conditional probability density function of each legitimate QAM symbol, $\mathcal{I}_i, i = 1, \ldots, \mathcal{M}$ which is described by Equation 2.86. The *a posteriori* probability $\varsigma_i(k)$ of the transmitted symbols, can be evaluated from the conditional density function, $\zeta_i(k)$ as follows:

$$\begin{aligned} \varsigma_i(k) &= P(I_{k-\tau} = \mathcal{I}_i | \mathbf{v}_k) \\ &= \frac{P(\mathbf{v}_k | I_{k-\tau} = \mathcal{I}_i) \cdot P(I_{k-\tau} = \mathcal{I}_i)}{P(\mathbf{v}_k)} \\ &= \frac{\zeta_i(k)}{P(\mathbf{v}_k)}, \qquad -\infty \leq k \leq \infty. \end{aligned} \tag{3.8}$$

The *a posteriori* probability $\tilde{\varsigma}(k)$ of the detected symbol can be obtained without the knowledge of the term $P(\mathbf{v}_k)$, if the *a posteriori* probability has unity support (i.e. the sum of the *a posteriori* probabilities of all symbols is unity):

$$\tilde{\varsigma}(k) = \frac{\zeta_i^*(k)}{\sum_{i=1}^{\mathcal{M}} \zeta_i(k)}, \qquad -\infty \leq k \leq \infty, \tag{3.9}$$

where $\zeta_i^*(k) = \max\{\zeta_i(k), 1 \leq i \leq \mathcal{M}\}$, as defined in Equation 2.85. Therefore, the probability of a *symbol* error associated with the decision $\tilde{I}_{k-\tau} = \mathcal{I}_i^*$ is given by:

$$P_s(k) = 1 - \tilde{\varsigma}(k), \qquad -\infty \leq k \leq \infty, \tag{3.10}$$

and the overall probability of symbol error of the detector is given by:

$$P_{\text{symbol}} = E\{P_s(k)\} \qquad -\infty \leq k \leq \infty. \tag{3.11}$$

Similarly, the probability of a *bit* error can be obtained from the *a posteriori* probability of the bits representing the QAM symbols. Below we provide an example for the 4-QAM scheme. The *a posteriori* probability of the 4 symbols, $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{I}_3$ and $\mathcal{I}_4$, is estimated by the RBF networks as ς_1, ς_2, ς_3 and ς_4, respectively. A 4-QAM symbol is denoted by the bits U_0U_1 and the symbols $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{I}_3$ and $\mathcal{I}_4$ correspond to 00, 01, 10 11, respectively. Thus, the *a posteriori* probability of the bits is given as follows:

$$\begin{aligned}
P(U_0 = 1) &= P(U_0U_1 = 11 \cup U_0U_1 = 10) = \varsigma_4 + \varsigma_3, \\
P(U_0 = 0) &= P(U_0U_1 = 01 \cup U_0U_1 = 00) = \varsigma_2 + \varsigma_1, \\
P(U_1 = 1) &= P(U_0U_1 = 11 \cup U_0U_1 = 01) = \varsigma_4 + \varsigma_2, \\
P(U_1 = 0) &= P(U_0U_1 = 10 \cup U_0U_1 = 00) = \varsigma_3 + \varsigma_1.
\end{aligned} \tag{3.12}$$

In general, the average probability of bit error for the detected symbol at signalling instant k is given by:

$$P_b(k) = \frac{\sum_{i=0}^{\text{BPS-1}} 1 - P(U_i(k) = b_i)}{\text{BPS}}, \tag{3.13}$$

where BPS denotes the number of bits per symbol and b_i is the value (either 0 or 1) of the ith bit of the symbol exhibiting the maximum *a posteriori* probability. The overall probability of bit error for the detector is given by:

$$P_{\text{bit}} = E\{P_b(k)\} \qquad -\infty \leq k \leq \infty. \tag{3.14}$$

For our joint RBF based equalisation and AQAM scheme, we are unable to obtain the true probability of bit error for the detector, namely P_{bit} averaged over all data bursts, since we need to collect a large number of received samples for an accurate estimation. We can only obtain the short-term probability of bit error, $P_{\text{bit, short-term}}$, which is the average bit error probability over a data burst that was received, i.e.,

$$P_{\text{bit, short-term}} = \frac{\sum_{n=1}^{L_D} P_b(n)}{L_D}, \tag{3.15}$$

where L_D is the number of data symbols per burst. Thus, we could estimate the channel quality on a BbB basis, relying on the estimated $P_{\text{bit, short-term}}$ value. The short-term probability of bit error or BER is only an estimate of the actual P_{bit} of the system for the duration of the data burst. The accuracy of the estimation is dependent on the number of data symbols L_D in the burst. This issue will not be discussed further for now.

Having described the switching metric used by the joint AQAM and RBF equaliser scheme, we will further investigate this scheme with the aim of producing a best-case performance estimate. Before proceeding, the next section will present the assumptions used, when we employ this scheme in a wideband channel environment.

3.3.3 Best-case Performance Assumptions

In deriving the best-case performance of this joint adaptive modulation and RBF based equalisation scheme, the following assumptions are made:

1. Perfect CIR estimation or channel state estimation is assumed at the receiver. The RBF's centres are assigned the values of the channel states. The associated CIR and channel state estimation techniques were presented in Section 2.9.3, 2.9.4 and 2.9.5. We note that incorrect estimation of the channel states will degrade the performance of the constituent fixed modulation modes, as it was demonstrated by our simulation results in Section 2.12. This degradation is neglected here with the aim of deriving a best-case performance estimate.

2. The CIR is time-invariant for the duration of the transmission burst, but varies from burst to burst, which corresponds to assuming that the channel is slowly varying. However, if the CIR changes during the transmission burst or if the estimation algorithm gives an inaccurate channel estimate, the effect of the channel variations can be considered by modifying the noise variance estimate, as discussed in [38, 87]. Let us briefly summarize this idea. We define the error between the noisy channel output v_k and the estimated noiseless channel state output $\hat{v}_k$ as follows:

$$\begin{aligned} e_k &= v_k - \hat{v}_k \\ &= v_k - \sum_{n=0}^{L} \hat{f}_n I_{k-n} \\ &= \triangle_f(I_k, \ldots, I_{k-L}) + \eta_k, \end{aligned} \tag{3.16}$$

   where $\triangle_f(\cdot)$ is an error function caused by an inaccurate estimate of the channel impulse response $\hat{f}_n, n = 0, \ldots, L$. Having determined this noise term, the RBF equaliser uses the noise variance in its width parameter seen in Equation 2.81 in order to compute the conditional probability densities of each ligitimate QAM symbols. Therefore, by computing the 'noise variance' as the average of e_k^2, and substituting these values in Equation 2.81 yields $\rho = 2E\left[e_k^2\right]$. Hence we translated the CIR estimation error to a noise-like term.

3. We assume furthermore that the receiver has perfect knowledge of the modulation mode used in its received transmission burst. For a practical system, control symbols

must be used to convey the modulation mode employed by the transmitter to the receiver [88, 89].

4. The RBF DFE used in the system neglects error propagation by feeding the correct symbol to be used for RBF subset centre selection or space translation, as described in Section 2.11. However, at low target BERs, we will expect low performance degradation due to decision feedback error propagation, as it was demonstrated in Figure 2.42.

5. The short-term probability of error estimate, namely $P_{\text{bit, short-term}}$, is known prior to transmission for all the modulation modes used in the system. This can be assumed in a TDD scenario, where the channel can be considered similar in the uplink and downlink transmission and when the channel is slowly varying. We also assume that given the estimated $P_{\text{bit, short-term}}$ for a particular modulation mode, the transmitter knows the corresponding short-term probability of bit error for the other modulation modes used in the system under the same channel conditions. Thus, the transmitter of a base station for example, can utilize its receiver's $P_{\text{bit, short-term}}$ estimation for its next transmission, provided that there is a high channel quality correlation between the transmitter and receiver slots. Note however that the latency between the transmitter and receiver slots can affect the quality of the estimation. This latency is mitigated, when employing slot-by-slot TDD - as in the third generation IMT-2000 and UTRA [90, 91, 92] proposals - where any TDD-slot can be configured as an uplink or downlink slot, hence reducing the latency of channel quality estimates.

During our further discourse we will gradually remove these idealistic assumptions.

Having described the assumptions stipulated, in order to derive the best-case performance of this joint adaptive modulation and RBF based equalisation scheme, we now describe our simulation model.

3.3.4 Simulation Model for Best-case Performance

In our experiments, pseudo-random symbols were transmitted in a fixed-length burst for all modulation modes over the burst-invariant wideband channel to fulfill assumptions 2 and 5. The receiver received each data burst having different modulation modes and equalised each one of them independently. The estimated short-term probability of bit error or BER was obtained for each modulation mode, as described in Section 3.3.2. The highest-order modulation mode, $\mathcal{M}^*$ that provided a short-term BER $P^{\mathcal{M}^*}_{\text{bit, short-term}}$, which was below

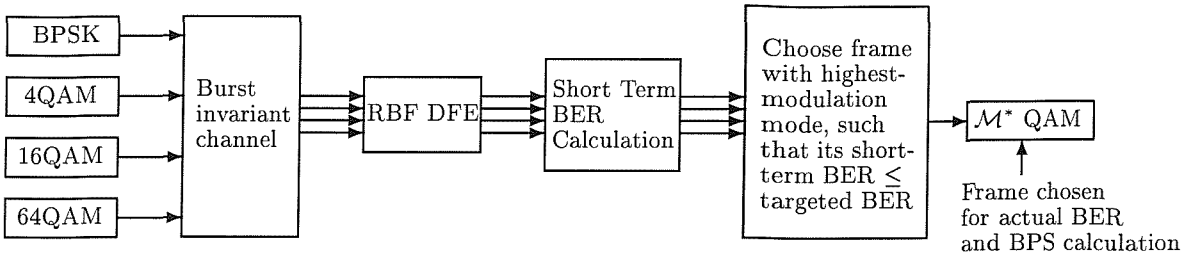


Figure 3.6: The simulation schematic of the joint AQAM and RBF DFE arrangement used for best-case BER performance estimation.

the target BER $P_{\text{bit, target}}$, when:

$$\mathcal{M}^* = \max\{\mathcal{M} = 2, 4, 16, 64, \text{ such that } P^{\mathcal{M}}_{\text{bit, short-term}} \leq P_{\text{bit, target}}\}, \tag{3.17}$$

was chosen to be the actual modulation mode that was used by the transmitter and the received equalised burst was used for the BER estimation of the system. The notation $P^{\mathcal{M}}_{\text{bit, short-term}}$ represents the short-term BER of $\mathcal{M}$-QAM. However, if all the modulation mode could not provide the targetted BER performance, i.e. $P^{2}_{\text{bit, short-term}} > P_{\text{bit, target}}$, NO TX mode is utilized. Figure 3.6 shows the simulation schematic of the joint AQAM and RBF DFE scheme used in our best-case BER performance evaluation. The next section will present our simulation results and analysis.

3.3.5 Simulation Results

The simulation parameters are listed in Table 3.1, noting that we analysed the joint AQAM and RBF equaliser scheme over a two-path Rayleigh fading channel. The wideband fading channel was burst-invariant. The RBF DFE used in our simulations had a feedforward order of $m = 2$, feedback order of $n = 1$ and delay of $\tau = 1$.

Number of data symbols per burst, L_D	144
Number of training symbols per burst, L_T	27
Transmission Frequency	1.9GHz
Transmission Rate	2.6MBd
Vehicular Speed	30 mph
Normalised Doppler Frequency	3.3×10^{-5}
Channel weights	$0.707 + 0.707z^{-1}$
RBF DFE feedforward order, m	2
RBF DFE feedback order, n	1
RBF DFE decision delay, τ	1

Table 3.1: Simulation parameters

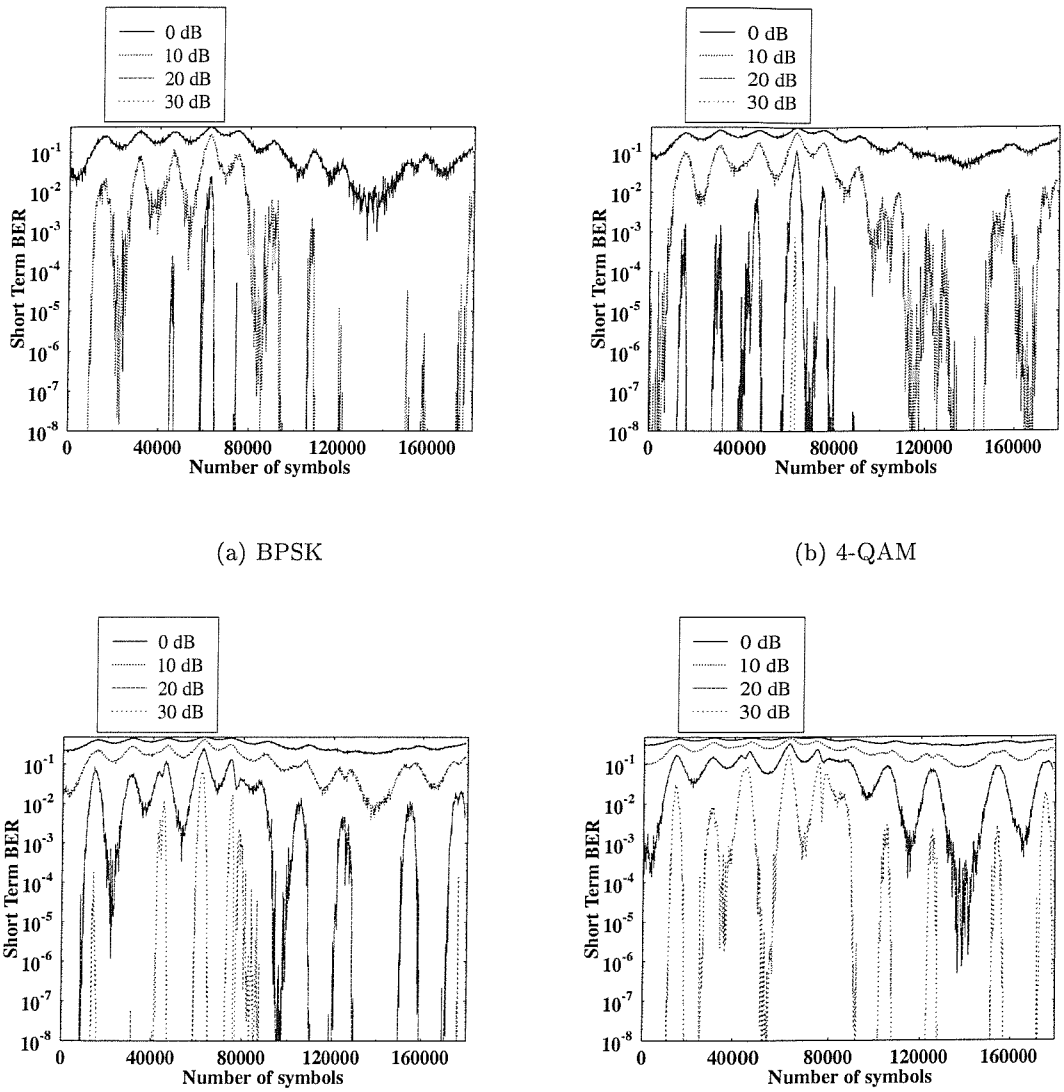
Figure 3.7 portrays the short-term BER of the burst-invariant channel versus symbol index, as estimated by the RBF DFE. For the simulated scenario, i.e., for a Doppler frequency of 3.3×10^{-5} the short-term BER is slowly varying and it is relatively predictable for a number of consecutive data bursts. Thus, assumption 2 of Section 3.3.3 is valid for this scenario.

The probability density function (PDF) of the BER estimation error of the RBF DFE for various channel SNRs is shown in Figure 3.8 for BPSK transmission bursts. The actual BER is the ratio of the number of bit errors encountered in a data burst to the total number of bits transmitted in that burst. Figure 3.8 suggests that the RBF DFE provides a good BER estimation, especially for high channel SNRs. We note, however that the accuracy of the actual BER evaluation is limited by the burst-length of 144 bits and its resolution is 1/144. Hence at high SNRs the actual number of errors registered is often 0, which portrays the BER estimation algorithm of Equation 3.15 in a less accurate light in the PDF of Figure 3.8, than it is in reality.

We will now analyse the best-case performance of the joint AQAM and RBF DFE scheme in more detail, using the simulation model described in Section 3.3.4 and the assumptions listed in Section 3.3.3. We designed two systems, a higher integrity scheme, having a target BER of 10^{-4}, which can be rendered error-free by error correction coding and hence we refer to this arrangement as a data transmission scheme; the lower integrity scheme was designed for maintaining a BER of 10^{-2}, which is adequate for speech transmission especially in conjunction with FEC. The target BPS values of these schemes were 3 and 4.5 bits per symbol, respectively, although these values can only be attained for sufficiently high SNRs.

Figure 3.9(a) and Figure 3.9(b) shows the simulated best-case performance of the joint AQAM and RBF DFE scheme for the target BER of 10^{-2} designed for speech transmission and for the target BER of 10^{-4} created for data transmission, respectively. The BER performance of the constituent fixed modulation modes is also depicted in both figures for comparison. The best-case performance was evaluated for two different adaptive modulation schemes. In the first scheme, the transmitter always transmitted data without transmission blocking, i.e. the NO TX mode of Equation 3.7 was not invoked. By contrast, in the second scheme, dummy data was transmitted, whenever the estimated short-term BER was higher than the target BER, a scenario, which we referred to as transmission blocking. The transmission of dummy data during blocking allowed us to keep monitoring the BER, in order to determine when to commence transmission and in which modem mode.

We will commence by analysing Figure 3.9(a), where the joint AQAM and RBF DFE scheme was designed for speech transmission, i.e. for a BER of 10^{-2}. For the adaptive scheme, which did not incorporate transmission blocking, the performance of adaptive modulation was better or equivalent to the performance of BPSK in terms of the mean



(a) BPSK

(b) 4-QAM

(c) 16-QAM

(d) 64-QAM

Figure 3.7: Short-term BER versus symbol index as estimated by the RBF DFE over the two-path equal-weight, symbol-spaced Rayleigh fading channel of Table 3.1. The RBF DFE had a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol. Perfect channel impulse response estimation is assumed and the error propagation due to decision feedback is ignored. The transmitted burst of Figure 2.44 consists of 171 symbols (144 data symbols and 27 training symbols).

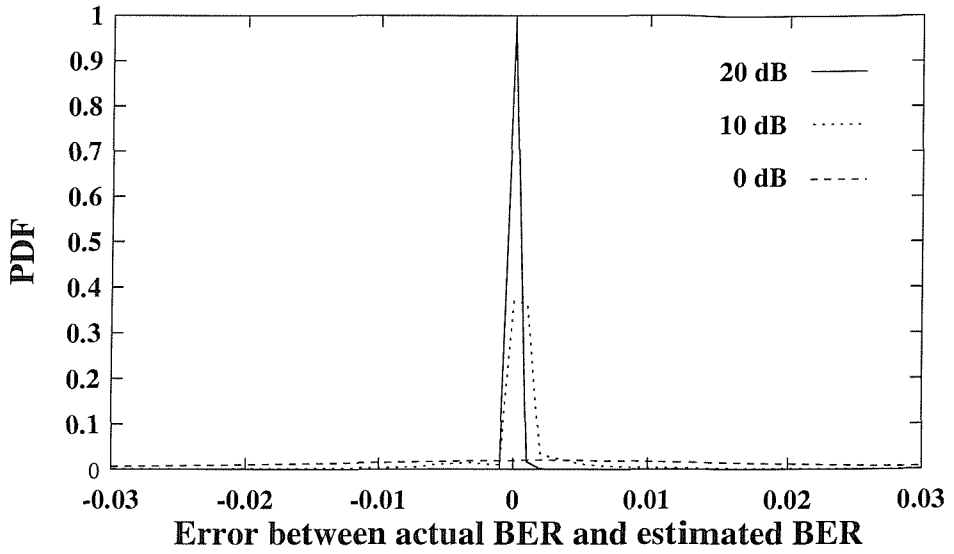


Figure 3.8: Discretised PDF of the error between the actual BER of the data burst and the BER estimated by the RBF DFE for the two-path Rayleigh fading channel of Table 3.1 using BPSK.

BER and mean BPS for the SNR range between 0dB and 9dB. At the channel SNR of 9dB, even though the mean BER performance was equivalent for the adaptive scheme and the BPSK scheme, the mean BPS for the adaptive scheme improved by a factor of 1.5, resulting in a mean BPS of 1.5. In the SNR range of 9dB to 16dB, the adaptive scheme outperformed the 4-QAM scheme in terms of the mean BER performance. At the channel SNR of 16dB, the mean BERs of both schemes are equivalent, although the mean BPS of the adaptive scheme is 2.7, resulting in a BPS improvement by a factor of 1.35, when compared to 4-QAM. At the channel SNR of 26dB, the mean BPS improvement of the adaptive scheme is by a factor of 1.3 for an equivalent mean BER. The adaptive scheme that utilized transmission blocking achieved a mean BER below 1%. At the channnel SNR of 12dB, even though the mean BER performance was equivalent for the BPSK scheme and the adaptive scheme with transmission blocking, the mean BPS for the adaptive scheme improved by a factor of 2. As the SNR improved, the performance of the adaptive schemes both with and without transmission blocking converged, since the probability of encountering high short-term BERs reduced. The mean BER and mean BPS performance of both adaptive schemes converged to that of 64-QAM for high SNRs, where 64-QAM becomes the dominant modulation mode.

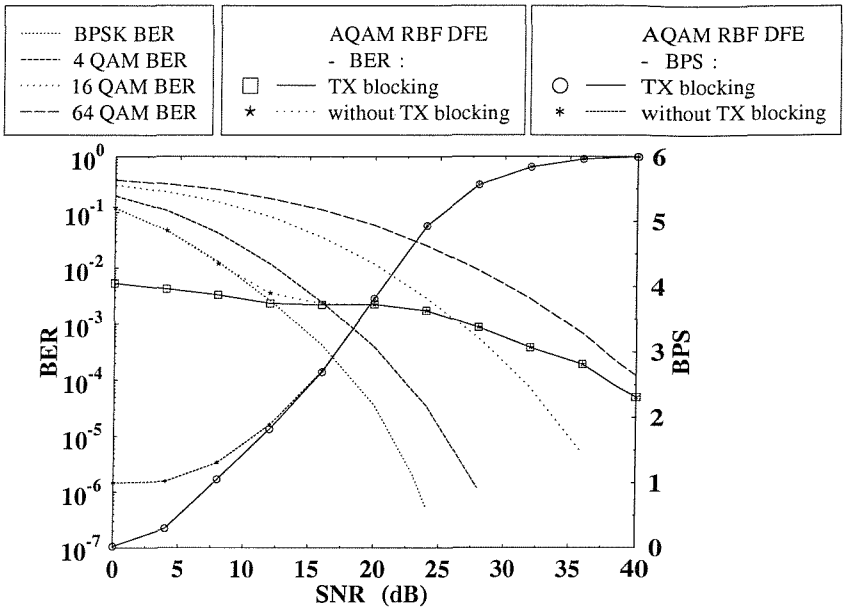
Similar trends were observed for data-quality transmission, i.e. for the 10^{-4} target BER scheme in Figure 3.9(b). However, we note that for the SNR range between 8dB to 20dB, the mean BER of the adaptive scheme without transmission blocking was better, than that of BPSK. This phenomenon was also observed in the narrowband adaptive modulation

scheme of [80] and in the wideband joint AQAM and DFE scheme of [83, 84], which can be explained as follows. The mean BER of the system is the ratio of the total number of bit errors to the total number of bits transmitted. The mean BER will decrease with decreasing number of bits error and with increasing number of total bits transmitted in the data burst. For a fixed number of symbols transmitted, the number of total bits transmitted in a data burst is constant for the BPSK scheme, while for the AQAM scheme the total number of bits transmitted in a data burst increased, when a higher-order AQAM mode was used. However, in this case the BER increased. If the relative bits per symbol increment upon using AQAM is higher than the relative bit error ratio increment, then the mean BER of the adaptive scheme will be improved. Consequently the adaptive mean BER can be lower than that of BPSK.

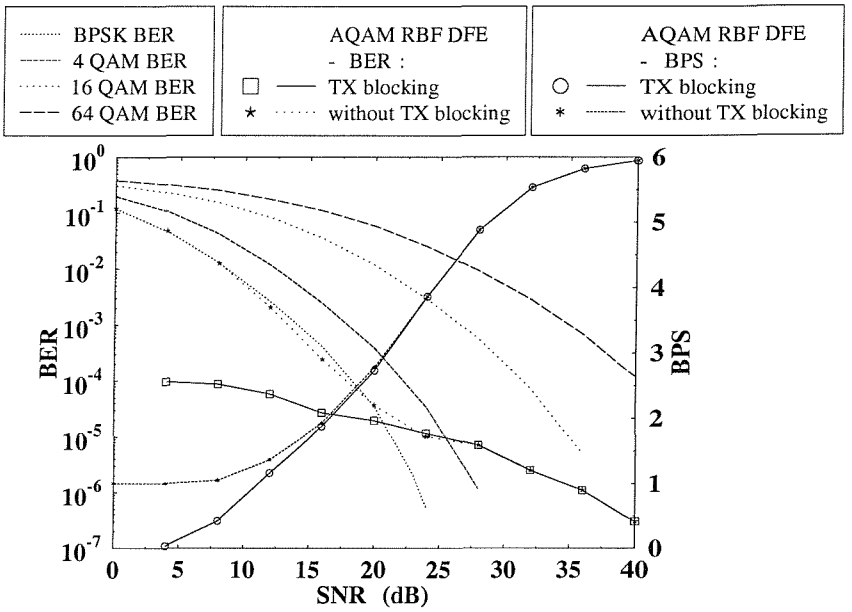
The probability of encountering each modulation mode employed in the adaptive scheme based on the estimated short-term BER switching mechanism is shown in Figure 3.10 and Figure 3.11 for the BER = 10^{-2} and BER = 10^{-4} schemes, respectively. As expected, the sum of the probabilities at each particular SNR is equal to one. At low SNRs, the lower order modulation modes (NO TX or BPSK) are dominant, producing a robust system. At higher SNRs, the higher order modulation modes become dominant, yielding a higher mean BPS and yet a reduced mean BER. From Figure 3.11(b), we observe that the transmission blocking mode was dominant in the SNR range of 0dB to 4dB and thus the mean BER performance was not recorded in that range of SNRs in Figure 3.9(b).

Comparing Figure 3.10(a) and Figure 3.11(a), the probability of transmission blocking was higher for data-quality transmission, in order to achieve a lower target BER due to the associated more stringent BER requirements of 10^{-4}. The probability of transmission blocking was close to zero, once the channel SNR increased to about 16dB and 20dB for the BER = 10^{-2} and BER = 10^{-4} schemes, respectively. These are the points, where the performance of the adaptive schemes with and without transmission blocking converged, as demonstrated in Figure 3.9. We observed that the probabilities of the 4-QAM, 16-QAM and 64-QAM modes being utilized for the adaptive scheme with and without transmission blocking was fairly similar. This is because introducing transmission blocking will predominantly affect the probability of BPSK, which will be utilized instead of no data transmission.

In summary, the AQAM RBF DFE scheme has its advantages, when compared to the individual fixed modulation modes in terms of the mean BER and mean BPS performance. Note however for the adaptive scheme without transmission blocking that the target performance of BER = 10^{-2} and BER = 10^{-4} can only be achieved, if the channel SNR is higher than 9dB and 18dB, respectively. The target mean BERs for speech transmission (BER = 10^{-2}) and data transmission (BER = 10^{-4}) were achieved for all channel SNRs, when

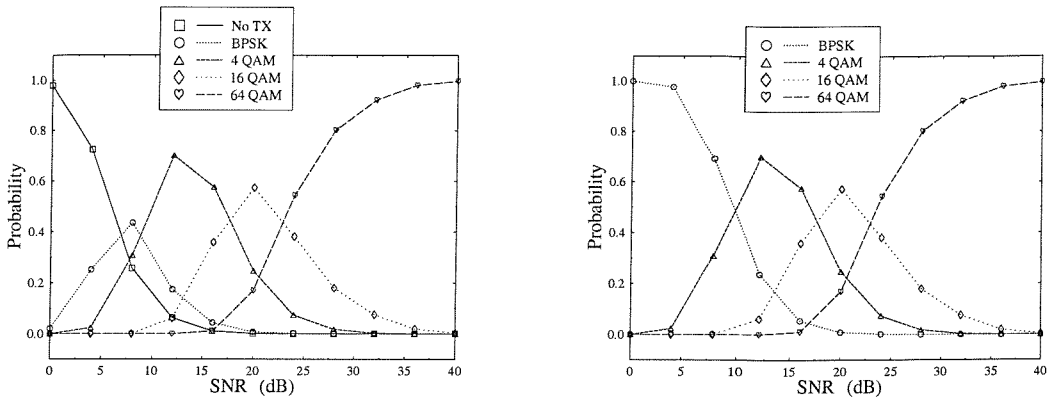


(a) Target BER is 10^{-2} (mean BER for speech transmission)



(b) Target BER is 10^{-4} (mean BER for data transmission)

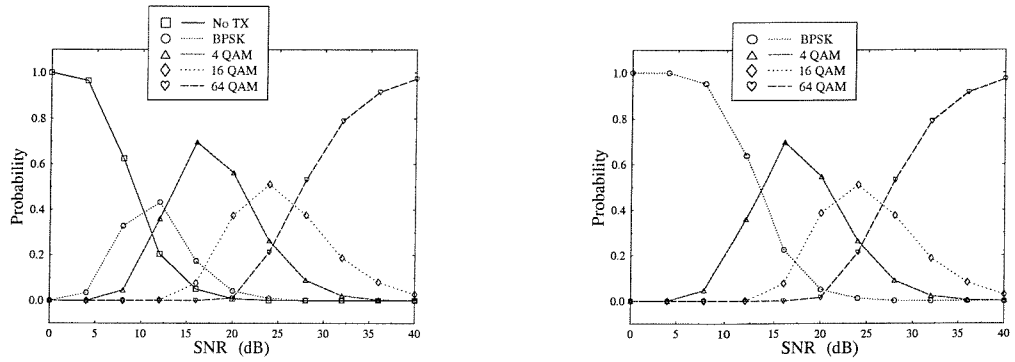
Figure 3.9: The simulated best-case performance of the AQAM RBF DFE showing also the BER performance of the constituent fixed modulation schemes, namely BPSK, 4-QAM, 16-QAM and 64-QAM, over the two-path Rayleigh-fading channel of Table 3.1 and using the assumptions of Section 3.3.3.



(a) With transmission blocking

(b) Without transmission blocking

Figure 3.10: The probability of encountering the various $\mathcal{M}$-QAM modulation modes in the joint AQAM and RBF DFE scheme for **best-case performance** during **speech-quality transmission (target BER of 0.01)** over the two-path equal-weight, symbol-spaced Rayleigh fading channel using the simulation parameters listed in Table 3.1 and the assumptions stated in Section 3.3.3.



(a) With transmission blocking

(b) Without transmission blocking

Figure 3.11: The probability of encountering the various $\mathcal{M}$-QAM modulation modes in the joint AQAM and RBF DFE scheme for **best-case performance** during **data-quality transmission (target BER of 10^{-4})** over the two-path equal-weight, symbol-spaced Rayleigh fading channel using the simulation parameters listed in Table 3.1 and the assumptions stated in Section 3.3.3.

we utilized transmission blocking. The target performance for speech (BER = 10^{-2}) and data (BER = 10^{-4}) transmission in terms of mean BPS (4.5 and 3, respectively) can only be achieved for the AQAM scheme with and without transmission blocking, if the channel SNR is in excess of about 22dB. Thus, the advantage of using an adaptive scheme with transmission blocking is that the performance of the joint AQAM and RBF DFE scheme can be 'tuned' to a certain required mean BER performance. However, the disadvantage is that the utilization of transmission blocking results in transmission latency, an issue, which was addressed for example in [93, 94]. Specifically, the interdependency of the required buffer size, doppler frequency and latency was analysed. Furthermore, frequency hopping was proposed for reducing the average duration of NO TX mode at low Doppler frequencies, where the latency and the buffer size may become excessive.

Let us now embark on a comparative analysis between the joint AQAM RBF DFE scheme and the Kalman-filtering based joint AQAM DFE scheme introduced by Wong *et. al.* [84] for wideband channels. The joint AQAM DFE scheme in [84] used the *pseudo-SNR* at the output of the DFE as the switching metric, an issue discussed briefly in Section 3.2. The pseudo-SNR at the output of the DFE was compared to a set of pseudo-SNR thresholds optimized using Powell's method [86]. Table 3.2 gives the results of the optimization process invoked, in order to achieve transmission integrities of 10^{-2} and 10^{-4} over the two-path Rayleigh-fading channel of Table 3.1 [84]. The conventional DFE used in the adaptive scheme had a feedforward order of $m = 15$, feedback order of $n = 2$ and decision delay of $\tau = 15$ symbols. The parameters m, n and τ of the conventional DFE were chosen such that it exhibited the best possible performance for our simulation scenario and hence further increase of the feedforward order would not give a significant performance improvement. We note again that for our best-case performance comparisons, the switching metric used for both schemes – namely the short-term BER for the AQAM RBF DFE scheme and the pseudo SNR for the AQAM DFE scheme – was estimated perfectly prior to transmission and the appropriate AQAM mode was chosen for the data burst to be transmitted, which satisfied the target BER requirement.

	l_1(dB)	l_2(dB)	l_3(dB)	l_4(dB)
Speech	3.68026	6.3488	11.7181	17.8342
Data	8.30459	10.4541	16.8846	23.051

Table 3.2: The optimized switching levels l_n of the joint adaptive modulation and DFE scheme for speech and data transmission in the two-path Rayleigh fading channel [84]. The target mean BER and BPS performance for speech was 10^{-2} and 4.5, respectively, while for computer data, 10^{-4} and 3, respectively.

Figure 3.12 provides the BER performance comparison of the conventional DFE and the RBF DFE over the two-path Rayleigh fading channel of Table 3.1 for the constituent fixed modulation modes. The BER performance of the RBF DFE for BPSK and 4-QAM was better than that of the conventional DFE, as the SNR increased. By contrast, the BER performance of the RBF DFE was inferior compared to that of the conventional DFE for 16- and 64-QAM. The performance of the RBF DFE can be, however, improved by increasing both the decision delay τ and the feedforward order m, as argued in Section 2.11, at the expense of increased computational complexity. However, the present parameter values for the conventional DFE and RBF DFE are convenient, since they yield similar BER performances.

The performance comparison of the adaptive schemes, i.e. that of the AQAM DFE and AQAM RBF DFE, is given in Figure 3.13. For the 10^{-2} target BER system, the AQAM RBF DFE provides a better BER performance, than the Kalman-filtering based AQAM DFE in the SNR range from 0dB to 28dB at the expense of a lower BPS performance, especially for higher SNRs. As the SNR exceeds 28dB, the BER performance of the AQAM DFE scheme becomes superior to that of the AQAM RBF DFE. This is because at higher SNRs the 64-QAM modulation mode prevails and since the 64-QAM BER performance of the conventional DFE was better, than that of the RBF DFE in Figure 3.12, hence the mean BER improvement of the AQAM DFE is expected, when compared to that of the AQAM RBF DFE.

For the 10^{-4} target BER system, the BER performance of the AQAM DFE and AQAM RBF DFE is fairly similar in the SNR range from 5dB to 12dB, but the BPS performance of the AQAM RBF DFE is better, than that of the AQAM DFE in that range. In this SNR range the lower-order modulation modes dominate. Since the RBF DFE can provide a better BER performance, than that of the conventional DFE for the lower-order modulation modes, the BPS performance of the AQAM RBF DFE can be improved, while maintaining a similar BER performance to that of the AQAM DFE. As the SNR exceeds 12dB, the BER performance of the AQAM RBF DFE remains better at the expense of a lower BPS performance.

The overall results of our simulations show that the AQAM RBF DFE is capable of performing similarly to the AQAM DFE at a lower decision delay and lower feedforward and feedback order. However, the computational complexity of the RBF DFE is dependent on the modulation mode, since the number of RBF centres increases with the number of modulation levels, as discussed in Section 2.7. This is not so in the context of the conventional DFE, where the computational complexity is only dependent on the feedforward and feedback order. Table 3.3 compares the computational complexity of the RBF DFE ($m = 2$, $n = 1$, $\tau = 1$) and the conventional DFE ($m = 15$, $n = 2$) used in our simulations. The

complexity analysis of the RBF DFE is based on Table 2.10. The high computational cost incurred by the RBF DFE in the high-order $\mathcal{M}$-ary modulation modes presents a drawback for the AQAM RBF DFE scheme.

Operation	RBF DFE				Conventional DFE
	BPSK	4-QAM	16-QAM	64-QAM	
subtration and addition	15	60	1008	16320	16
multiplication	12	48	768	12288	17
division	4	16	256	4096	0
exp()	4	16	256	4096	0

Table 3.3: Computational complexity of RBF DFE and conventional DFE per equalised output sample. The RBF DFE has a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol. The number of RBF hidden units $n_{s,j}$ is dependent on the order of the $\mathcal{M}$-QAM modes and the channel memory L where $n_{s,j} = \mathcal{M}^{m+L-n}$. The channel memory is assumed to be $L = 1$. The complexity analysis of the RBF DFE is based on Table 2.10. The conventional DFE has a feedforward order of $m = 15$, feedback order of $n = 2$ and decision delay of $\tau = 15$ symbols.

Nevertheless, we note that unlike the conventional DFE, the AQAM RBF DFE is capable of performing well over channels, which result in non-linearly separable received phasor constellations.

3.3.6 Discussion

In the above sections, BbB adaptive modulation was applied in conjunction with the RBF DFE of Section 2.11 in a wideband channel environment. The short-term BER of Equation 3.15 estimated by the RBF DFE was used as the modem mode switching metric in order to switch between different modulation modes. The validity of using this metric was tested in Section 3.3.5 and in Figure 3.8 it was shown that the RBF DFE gives a good BER estimate for the adaptive scheme to maintain the target mean BER performance. The simulation results also showed that there was a performance improvement in terms of the mean BER and mean BPS, when compared to the constituent fixed modulation modes. The performance of the joint AQAM RBF DFE scheme was then compared to that of the joint AQAM conventional DFE scheme investigated by Wong [84]. The AQAM RBF DFE having a lower feedforward and feedback order and a smaller decision delay, showed comparable performance to the AQAM DFE in our simulations.

In our future work, the performance of the AQAM RBF DFE will be investigated in practical situations, where the effect of discarding the assumptions made in Section 3.3.3 is to be quantified.

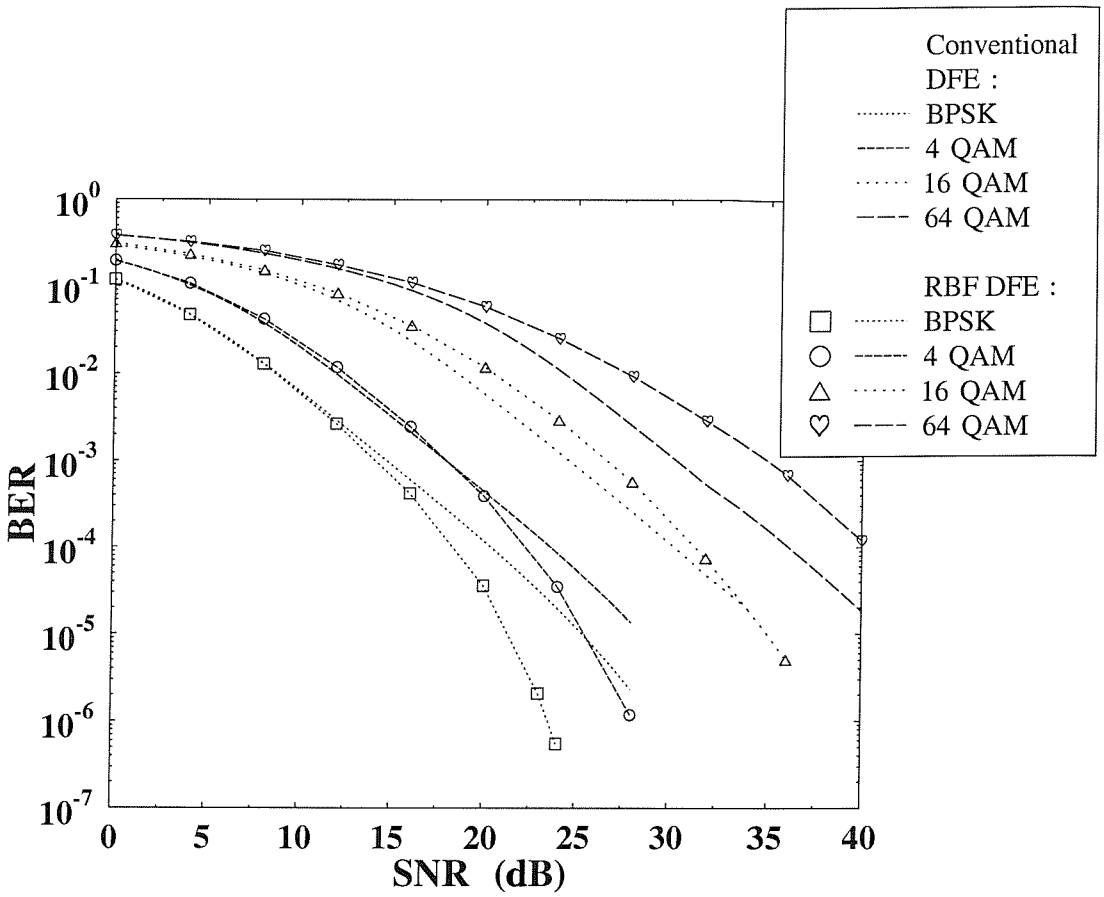


Figure 3.12: BER versus SNR performance of the conventional DFE and the RBF DFE over the two-path equal-weight symbol-spaced Rayleigh-fading channel of Table 3.1 for different $\mathcal{M}$-QAM schemes. The conventional DFE has a feedforward order of $m = 15$, feedback order of $n = 2$ and decision delay of $\tau = 15$ symbols. The RBF DFE has a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbols.

3.4 Performance of the AQAM RBF DFE Scheme: Switching Metric Based on the Previous Short-term BER Estimate

In this section, we analyse the performance of the AQAM RBF DFE scheme by discarding assumption 5 of Section 3.3.3. Therefore, the estimated short-term BER of the *current* transmitted burst is used to select the modulation mode for the *next* transmission burst, as described in Equation 3.7.

The BER switching thresholds corresponding to $\mathcal{M}$-QAM, $P_i^{\mathcal{M}}, i = 2, 4, 16, 64$, can be obtained by estimating the BER degradation/improvement, when the modulation mode is

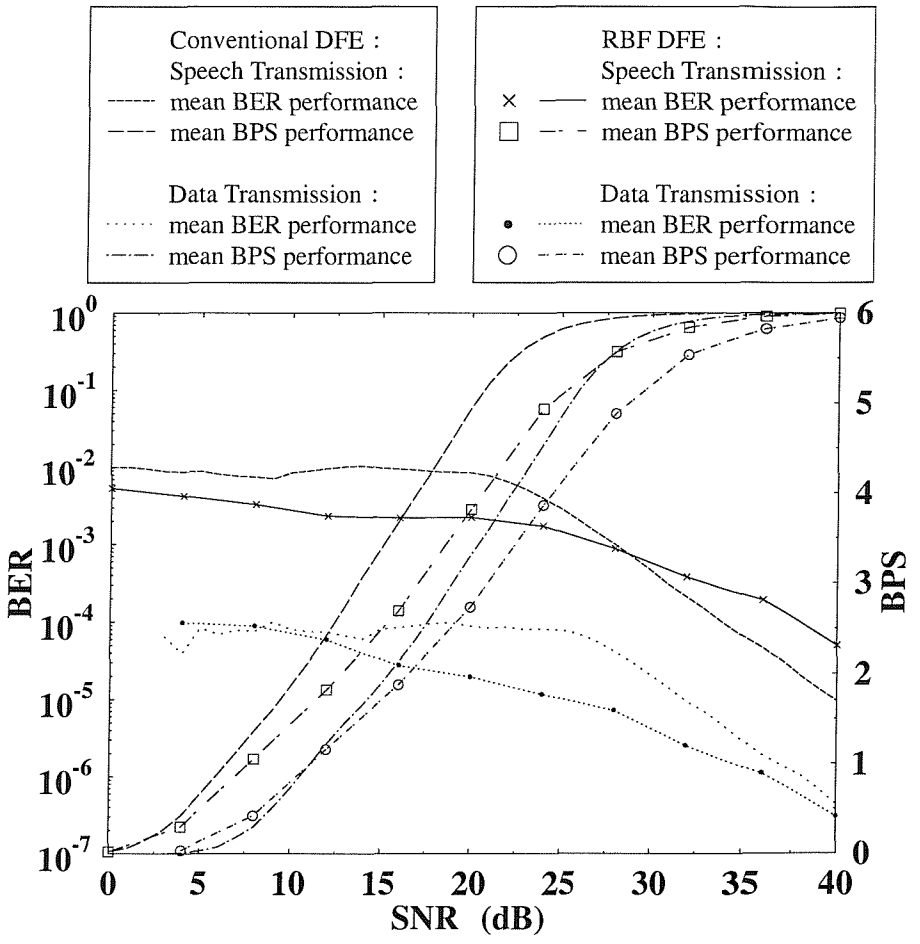


Figure 3.13: Simulated best-case performance of the AQAM RBF DFE scheme and the numerical best-case performance of the joint AQAM conventional DFE scheme for speech- and data-transmission [84], using the parameters listed in Table 3.1 and the assumptions stated in Section 3.3.3. The modem mode switching levels used for the joint AQAM conventional DFE scheme are listed in Table 3.2. The RBF DFE had a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol and the conventional DFE had a feedforward order of $m = 15$, feedback order of $n = 2$ and decision delay of $\tau = 15$ symbols.

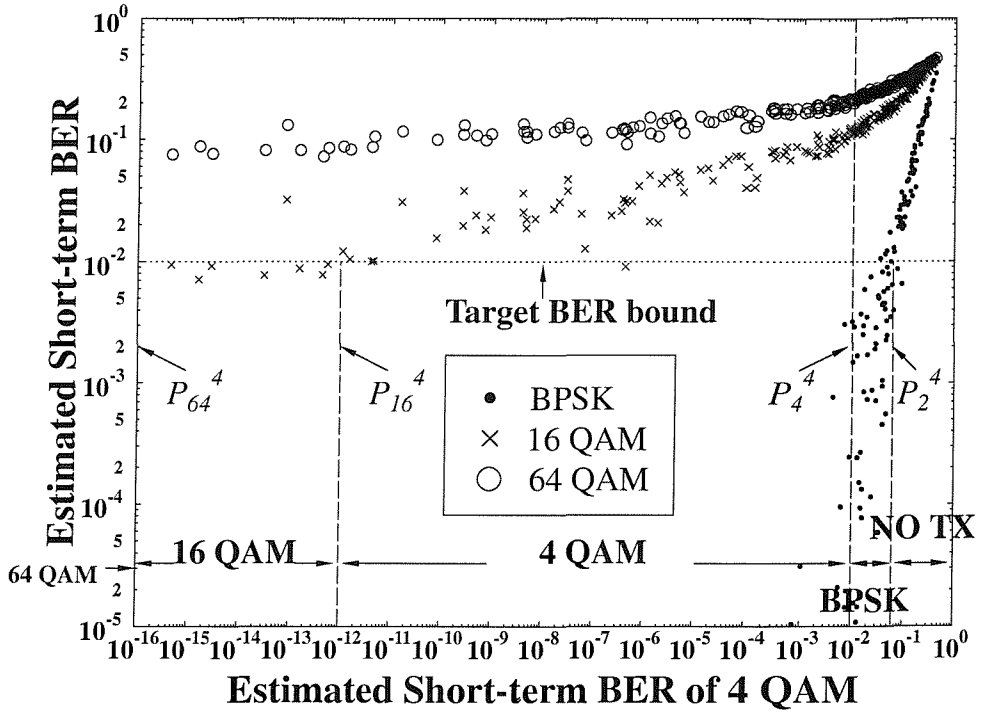


Figure 3.14: The estimated short-term BER for all the possible modulation modes that can be invoked, assuming that the current mode is 4-QAM – versus the estimated short-term BER of 4-QAM for the two-path Rayleigh fading channel of Table 3.1.

switched from $\mathcal{M}$-QAM to a higher/lower number of modulation levels. In this experiment, we obtain this BER degradation/improvement measure from the estimated short-term BER of every modulation mode used, under the same instantaneous channel conditions. Figure 3.14 shows the estimated short-term BER of all the possible modulation modes that can be invoked, assuming that the current mode is 4-QAM, versus the estimated short-term BER of 4-QAM under the same instantaneous channel conditions. The short-term BERs of the modulation modes are obtained on a burst-by-burst basis from the RBF DFE according to Equation 3.15. Each point in Figure 3.14 represents the RBF DFE's estimated short-term BER for a specific received data burst using the corresponding modulation mode. In order to maintain the target BER of 10^{-2}, Figure 3.14 demonstrates, how each switching BER threshold P_i^4 is obtained. The short-term BER of the 4-QAM transmission burst, when the corresponding BPSK, 16-QAM and 64-QAM transmission burst under the same instantaneous channel conditions has an estimated BER of 10^{-2} is approximately $6 \times 10^{-2}, 10^{-12}$ and 0, respectively. For example, if the estimated short-term BER of the received 4-QAM transmission burst is below $P_{16}^4 = 10^{-12}$, the modulation mode can be 'safely' switched to 16-QAM for the next transmission burst, since the short-term BER of this 16-QAM transmission burst is expected to be below the target BER of 10^{-2}. The 4-QAM error probability of $P_{16}^4 = 10^{-12}$ used in this example for switching to 16-QAM appears extremely conservative, but it is justified by the large uncertainty associated with the estimation of the BER due to the Rayleigh-faded impulse response taps. This manifests itself also in the rather spread nature of the BER estimates in Figure 3.14. A feasible technique for mitigating this phenomenon is employing the fade-tracking scheme of Figure 11.2 in Reference [61]. Using this method the switching BER thresholds were obtained for the target BER of 10^{-2} and 10^{-4}, as listed in Table 3.4 and Table 3.5, respectively in the context of all possible combinations of the mode transitions. Note that the extremely low values for $P_{16}^4 = 1 \times 10^{-45}$ and $P_{64}^{16} = 1 \times 10^{-50}$ in Table 3.5 were obtained by extrapolating the curves similar to Figure 3.14 but for 16-QAM and 64-QAM, respectively, in order to achieve the target BER of 10^{-4}.

Figure 3.15 shows the BER and BPS performance of the joint AQAM RBF DFE scheme designed for BER = 10^{-2} with the switching thresholds given in Table 3.4 – when using the current transmission burst's BER estimate, in order to determine the modem mode of the next transmission burst – in contrast to its best-case performance. The performance comparison shows that there is little performance degradation, when the *current* short-term BER estimate is used to control the modulation mode of the *next* transmission burst based on the switching parameters of Table 3.4 for the AQAM scheme designed for BER = 10^{-2}. Since the channel of Table 3.1 is slowly varying, the performance of the joint AQAM RBF DFE scheme based on the switching parameters of Table 3.4 is comparable to its best-case

	$P_2^{\mathcal{M}}$	$P_4^{\mathcal{M}}$	$P_{16}^{\mathcal{M}}$	$P_{64}^{\mathcal{M}}$
NO TX	9×10^{-3}	5×10^{-5}	0.0	0.0
BPSK	1×10^{-2}	5×10^{-5}	0.0	0.0
4-QAM	6×10^{-2}	1×10^{-2}	1×10^{-12}	0.0
16-QAM	2×10^{-1}	1×10^{-1}	1×10^{-2}	1×10^{-8}
64-QAM	3×10^{-1}	2×10^{-1}	9×10^{-2}	1×10^{-2}

Table 3.4: The switching BER thresholds $P_i^{\mathcal{M}}$ of the joint adaptive modulation and RBF DFE scheme for the target BER of 10^{-2} over the two-path Rayleigh fading channel of Table 3.1.

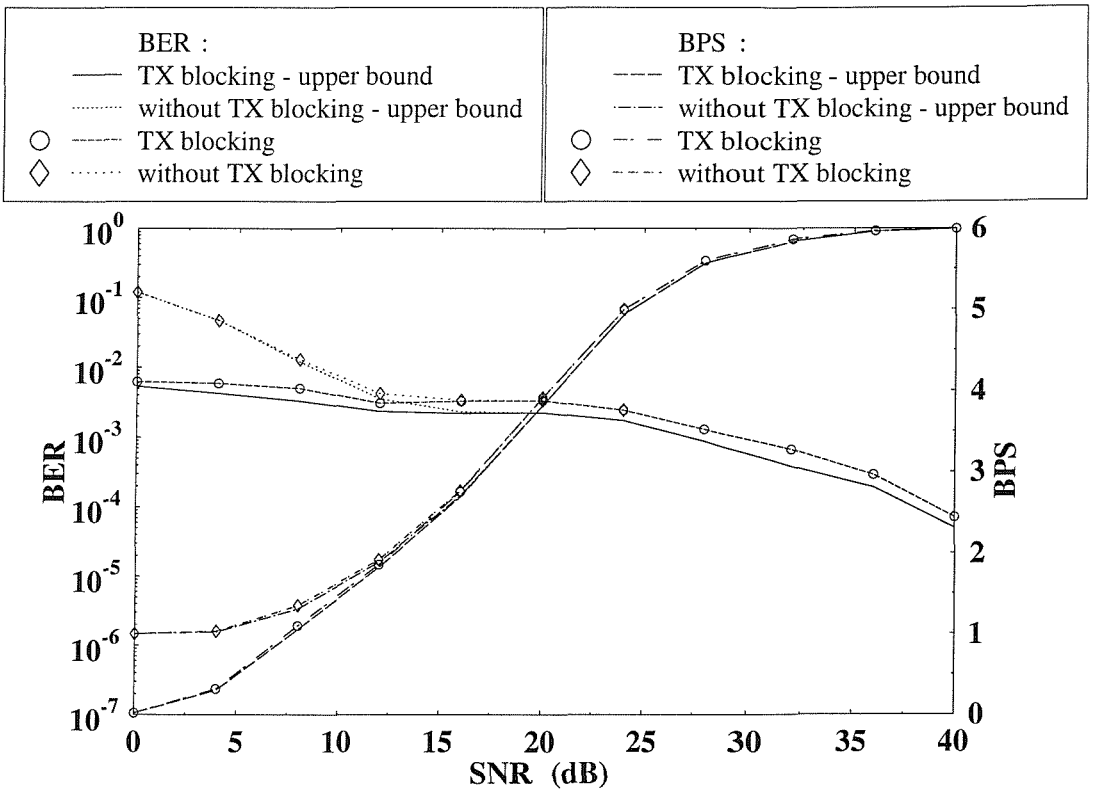


Figure 3.15: The BER and BPS performance of the joint AQAM RBF DFE scheme **using the current BER estimate in order to estimate the next burst's transmission mode**, and its best-case performance for the 10^{-2} target BER system, using the parameters listed in Table 3.1. The modem mode switching levels used for the joint AQAM RBF DFE scheme are listed in Table 3.4. The RBF DFE had a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol.

performance.

	$P_2^{\mathcal{M}}$	$P_4^{\mathcal{M}}$	$P_{16}^{\mathcal{M}}$	$P_{64}^{\mathcal{M}}$
NO TX	9×10^{-5}	1×10^{-15}	0.0	0.0
BPSK	1×10^{-4}	1×10^{-15}	0.0	0.0
4-QAM	1.5×10^{-2}	1×10^{-4}	1×10^{-45}	0.0
16-QAM	1.2×10^{-1}	5×10^{-2}	1×10^{-4}	1×10^{-50}
64-QAM	2.2×10^{-1}	1.5×10^{-1}	3×10^{-2}	1×10^{-4}

Table 3.5: The switching BER thresholds $P_i^{\mathcal{M}}$ of the joint adaptive modulation and RBF DFE scheme for the target BER of 10^{-4} over the two-path Rayleigh fading channel of Table 3.1.

Figure 3.16 shows the BER and BPS performance of the AQAM RBF DFE scheme designed for data-transmission using the switching threshold given in Table 3.5 in comparison to its best-case performance. The degradation with respect to the best-case performance of the AQAM RBF DFE scheme designed for data transmission at BER = 10^{-4} based on the switching threshold given in Table 3.5 is more significant compared to the adaptive scheme designed for BER = 10^{-2}, as seen in Figure 3.15 and Figure 3.16. Note that for the low BER switching thresholds of $P_{16}^4 (= 1 \times 10^{-45})$ and $P_{64}^{16} (= 1 \times 10^{-50})$ in Table 3.5 – which was required by the adaptive scheme for achieving the target BER of 10^{-4} – the RBF DFE is unable to provide BER estimates of such high accuracy. As the SNR improves, the relative frequency of encountering the switching thresholds P_{16}^4 and P_{64}^{16} increases and thus the performance degradation compared to the best-case increases. The performance degradation with respect to the best-case was also contributed by the spread nature of the BER estimates due to the Rayleigh-faded CIR taps. The BER estimation spread was more evident, when the BER estimate decreased, as shown in Figure 3.14. Therefore, there is a substantial BER estimation inaccuracy associated with the switching thresholds P_{16}^4 and P_{64}^{16}.

Figure 3.17 and 3.18 compare the probability of encountering each modulation mode employed in the adaptive scheme and those employed in the best-case performance scenario for speech-quality (BER = 10^{-2}) transmission and data-quality (BER = 10^{-4}) transmission, respectively. Figure 3.17 shows that the switching BER thresholds of Table 3.4, determined with our suggested method and the previous short-term BER estimate is capable of providing similar modulation mode utilization for the adaptive scheme designed for speech-quality transmission compared with its best-case performance. However, for data-quality transmission, we note from Figure 3.18 that the utilization of the 64-QAM mode of the adaptive scheme is more frequent, than that of the best-case performance for high SNRs. This also explains the substantial BER degradation from its best-case performance, as the

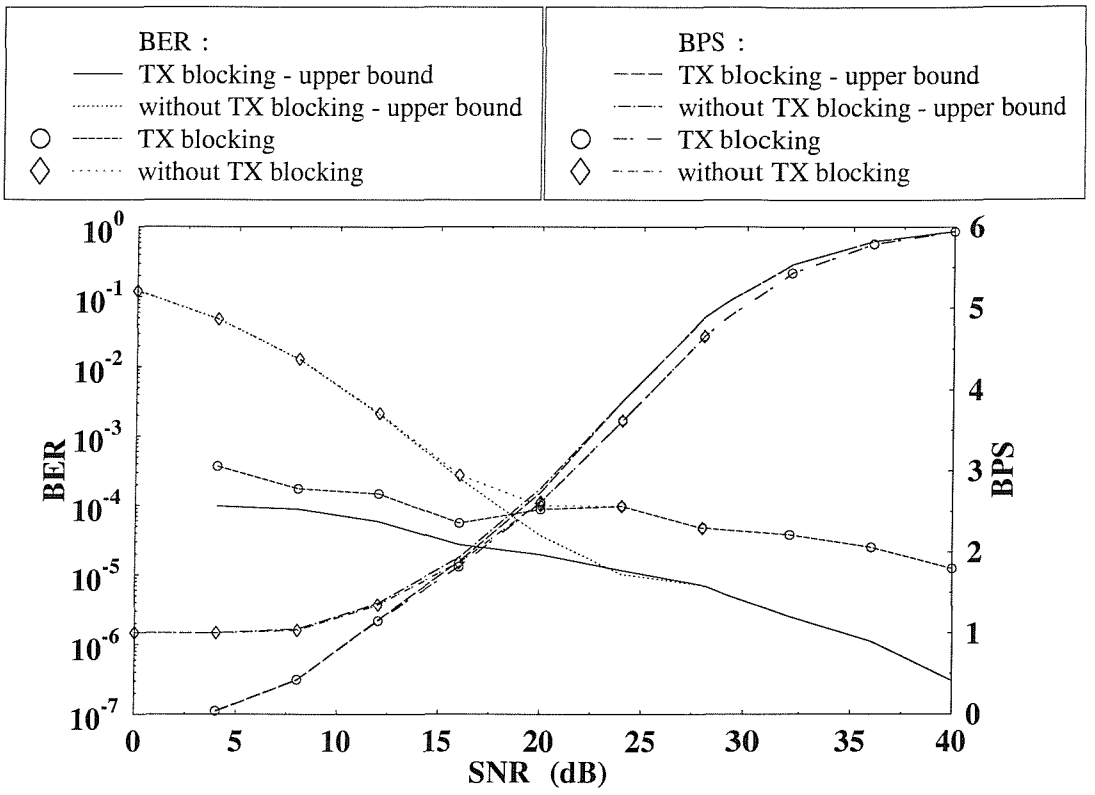
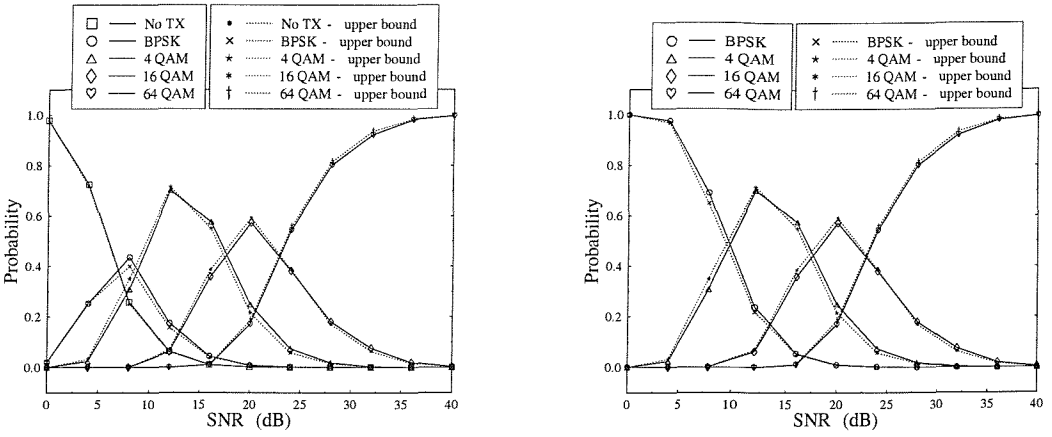


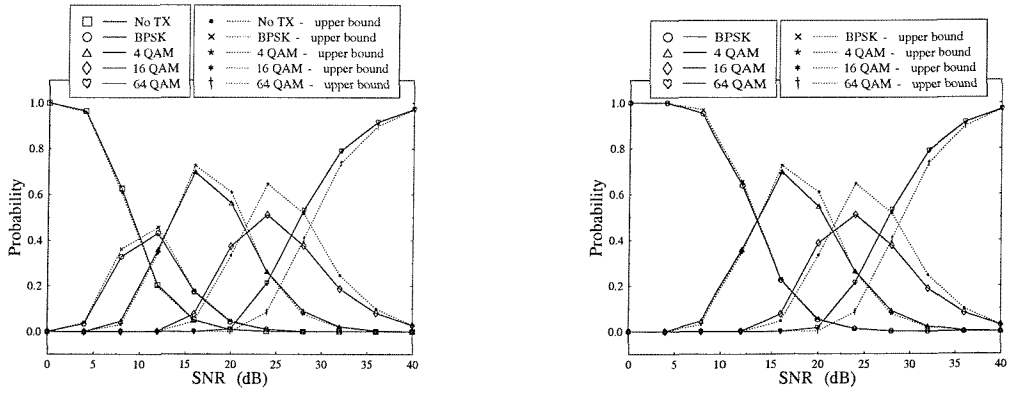
Figure 3.16: The BER and BPS performance of the joint AQAM RBF DFE scheme **using the current BER estimate in order to estimate the next burst's transmission mode**, and its best-case performance for **data-transmission**, using the parameters listed in Table 3.1. The modem mode switching levels used for the AQAM RBF DFE scheme are listed in Table 3.5. The RBF DFE had a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol.



(a) With transmission blocking

(b) Without transmission blocking

Figure 3.17: The probability of encountering the various $\mathcal{M}$-QAM modulation modes in the joint AQAM and RBF DFE scheme during **speech-quality transmission (target BER of 0.01)** over the two-path equal-weight, symbol-spaced Rayleigh fading channel using the simulation parameters listed in Table 3.1. The probability of modulation mode utilization for best-case performance, as given in Figure 3.10, is provided for comparison.



(a) With transmission blocking

(b) Without transmission blocking

Figure 3.18: The probability of encountering the various $\mathcal{M}$-QAM modulation modes in the joint AQAM and RBF DFE scheme during **data-quality transmission (target BER of 0.0001)** over the two-path equal-weight, symbol-spaced Rayleigh fading channel using the simulation parameters listed in Table 3.1. The probability of modulation mode utilization for the best-case performance, as given in Figure 3.11, is provided for comparison.

SNR improves, as demonstrate in Figure 3.16.

3.5 Conclusions

The RBF DFE was shown to provide a good 'on-line' BER estimation of the received data burst, which was used as the AQAM mode switching metric. Our simulation results showed that the proposed RBF DFE-assisted burst-by-burst adaptive modem outperformed the individual constituent fixed modulation modes in terms of the mean BER and BPS. Transmission blocking was utilised to maintain the target BER performance. Without transmission blocking, the target BER of 10^{-2} and 10^{-4} can only be achieved, when the channel SNR is higher than 9dB and 18dB, respectively. However, the disadvantage is that the utilization of transmission blocking results in transmission latency.

The AQAM scheme employing RBF DFE was compared to the AQAM scheme using conventional DFE in order to mitigate the effects of the dispersive wideband channel. Our results showed that the AQAM RBF DFE scheme was capable of performing as well as the conventional AQAM DFE at a lower decision delay and lower feedforward as well as feedback order. The performance of the AQAM RBF DFE can be improved by increasing both the decision delay τ and the feedforward order m, at the expense of increased computational complexity, while the performance of the conventional AQAM DFE cannot be improved significantly by increasing its equaliser order. However, the computational complexity of the RBF DFE is dependent on the AQAM mode and increases significantly for higher-order modulation modes. This is not so in the context of the conventional DFE, where the computational complexity is only dependent on the feedforward and feedback order.

A method to obtain the switching BER thresholds of the joint AQAM RBF DFE scheme was proposed in Section 3.4 and was shown to suffer only minor performance degradation in comparison to the achievable best-case performance generated by assuming that the corresponding BER of all modulation modes was known given the estimated BER of the received burst.

Overall, we have shown that our proposed AQAM scheme improved the throughput performance compared to the constituent fixed modulation modes. The RBF DFE provided a reliable channel quality measure, which quantified all channel impairments, irrespective of their source for the AQAM scheme and at the same time it improved the BER performance. In the following chapter, we will enhance the performance of the AQAM RBF DFE by invoking turbo coding.

Chapter 4

RBF Equalisation Using Turbo Codes

In this chapter, the wideband AQAM scheme explored in the previous chapter is extended to incorporate the benefits of channel coding. Channel coding, with its error correction and detection capability, is capable of improving the BER and throughput performance of the wideband AQAM scheme. Since the wideband AQAM scheme always attempts to invoke the appropriate modulation mode in order to combat the wideband channel effects, the probability of encountering a received transmitted burst with a high instantaneous BER is low, when compared to the constituent fixed modulation modes. This characteristic is advantageous, since due to the less bursty error distribution, a coded wideband AQAM scheme can be implemented successfully without the utilization of long-delay channel interleavers. Therefore we can exploit the error detection capability of the channel codes near-instantaneously at the receiver for every received transmission burst.

Turbo coding [7] is invoked in conjunction with the RBF assisted AQAM scheme in a wideband channel scenario in this chapter. We will first introduce the novel concept of Jacobian RBF equaliser, which is a reduced-complexity logarithmic version of the RBF equaliser. The Jacobian logarithmic RBF equaliser generates its output in the logarithmic domain and hence it can be used to provide soft outputs for the turbo decoder. We will investigate different channel quality measures – namely the short-term BER and average burst log-likelihood ratio magnitude of the bits in the received burst before and after channel decoding – for controlling the mode-switching regime of our adaptive scheme. We will now briefly review the concept of turbo coding.

4.1 Introduction to Turbo Codes

Turbo codes were introduced in 1993 by Berrou, Glavieux and Thitimajshima [7]. These codes achieve a near-Shannon-limit error correction performance with relatively simple component codes and invoking large interleavers. The component codes that are usually used are either recursive systematic convolutional (RSC) codes or block codes. The general structure of the turbo encoder is shown in Figure 4.1. The information sequence is encoded twice, using an interleaver or scrambler between the two encoders, rendering the two encoded data sequences approximately statistically independent of each other. The encoders produce a so-called systematically encoded output, which is equivalent to the original information sequence, as well as a stream of parity information bits. The parity outputs of the two component codes are then often punctured in order to maintain as high a coding rate as possible, without substantially reducing the codec's performance. Finally, the bits are multiplexed before being transmitted.

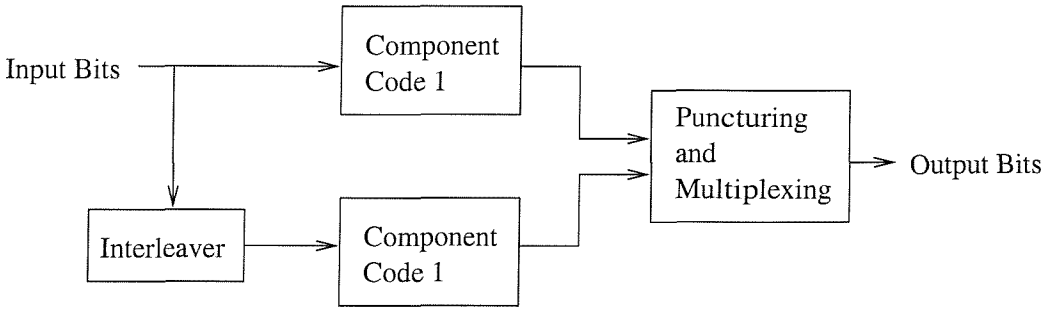


Figure 4.1: Turbo encoder schematic

The turbo decoder consists of two decoders, linked by interleavers in a structure obeying the constraints imposed by the encoder, as seen in Figure 4.1. The turbo decoder accepts soft inputs and provides soft outputs as the decoded sequence. The soft inputs and outputs provide not only an indication of whether a particular bit was a binary 0 or a 1, but also deliver the so-called log-likelihood ratio (LLR) of the bit which constituted by the logarithm of the quotient of the probability of the bit concerned being a logical one and zero, respectively. Two often-used decoders are the Soft Output Viterbi Algorithm (SOVA) [95] and the Maximum A Posteriori (MAP) [96] algorithm.

As seen in Figure 4.2, each decoder takes three types of inputs - the systematically encoded channel output bits, the parity bits transmitted from the associated component encoder and the information estimate from the other component decoder, referred to as the *a priori* information of the decoded bits. The decoder operates iteratively. In the first iteration, the first component decoder provides a soft output and the so-called extrinsic output based on the soft channnel outputs alone. The terminology 'extrinsic' implies that this information is

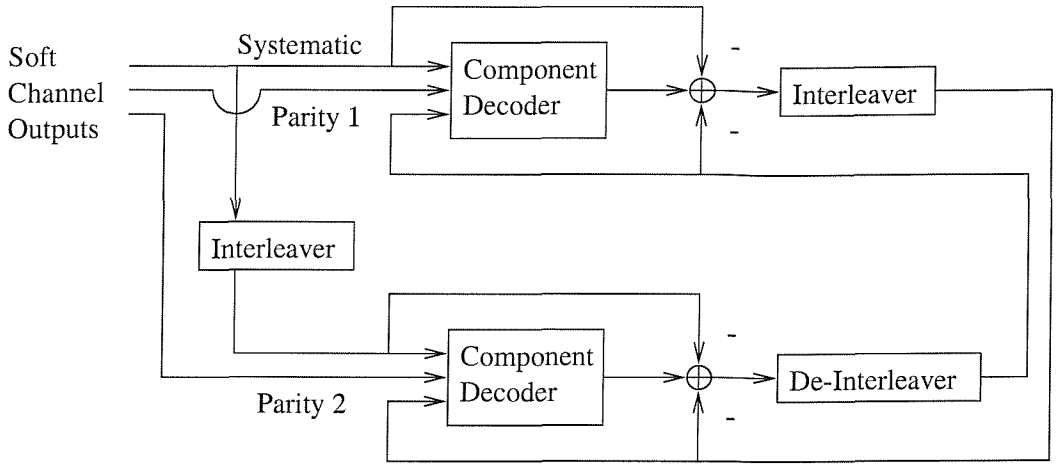


Figure 4.2: Turbo decoder schematic

not based on the received information directly related to the bit concerned, it is rather based on information, which is indirectly related to the bit due to the code-constraints introduced by the encoder. This extrinsic output generated by the first decoder – which constitutes the first decoder's 'opinion' as to the bit concerned – is used by the second component decoder as *a priori* information, and this information together with the channel outputs is used by the second component decoder, in order to generate its soft output and extrinsic information. Symmetrically, in the second iteration, the extrinsic information generated by the second decoder in the first iteration is used as the *a priori* information for the first decoder. Using this *a priori* information, the decoder is likely to decode more bits correctly than it did in the first iteration. This cycle continues and at each iteration the BER in the decoded sequence drops. However, the extra BER improvement obtained with each iteration diminishes, as the number of iterations increases. In order to limit the computational complexity, the number of iterations is usually fixed according to the prevalent design criteria expressed in terms of performance and complexity. When the series of iterations is curtailed, after either a fixed number of iterations or when a termination criterion is satisfied, the output of the turbo decoder is given by the de-interleaved *a posteriori* LLRs of the second component decoder. The sign of these *a posteriori* LLRs gives the hard decision output and in some applications the magnitude of these LLRs provides the confidence measure of the decoder's decision. Because of the iterative nature of the decoder, it is important not to re-use the same information more than once at each decoding step, since this would destroy the independence of the two encoded sequences which was originally imposed by the interleaver of Figure 4.2. For this reason the concept of the so-called extrinsic and intrinsic information was used in the original paper on turbo coding by Berrou *et al.* [7] to describe the iterative decoding of turbo codes.

For a more detailed exposition of the concept and algorithm used in the iterative decoding of turbo codes, the reader is referred to [7, 69]. Other, non-iterative decoders have also been proposed [97, 98] which give optimal decoding of turbo codes, but they are rather complex and providing disproportionately low improvement in performance over iterative decoders. Therefore, the iterative scheme shown in Figure 4.2 is usually used. Continuing from our previous work, where we used an RBF equaliser to mitigate the effects of the wideband channel, we will introduce turbo coding in order to improve the BER and/or BPS performance.

In the next section, before we discuss the joint RBF equalisation and turbo coding system, we will introduce the *Jacobian logarithmic RBF equaliser*, which computes the output of the RBF network in logarithmic form based on the Log-MAP algorithm [13] used in turbo codes to reduce their computational complexity.

4.2 Jacobian Logarithmic RBF Equaliser

The Bayesian-based RBF equaliser has a high computational complexity due to the evaluation of the nonlinear exponential functions in Equation 2.81 and due to the high number of additions/subtractions and multiplications/divisions required for the estimation of each symbol, as it was expounded in Section 2.9.

In this section – based on the approach often used in turbo codes – we propose generating the output of the RBF network in logarithmic form by invoking the so-called Jacobian logarithm [13, 14] , in order to avoid the computation of exponentials and to reduce the number of multiplications performed. We will refer to the RBF equaliser using the Jacobian logarithm as the *Jacobian logarithmic RBF equaliser*. Below we will present this idea in more detail.

We will first introduce the Jacobian logarithm, which is defined by the relationship [13]:

$$\begin{aligned} J(\lambda_1, \lambda_2) &= \ln(e^{\lambda_1} + e^{\lambda_2}) \\ &= \max(\lambda_1, \lambda_2) + \ln(1 + e^{-|\lambda_1 - \lambda_2|}) \\ &\approx \max(\lambda_1, \lambda_2) + f_c(|\lambda_1 - \lambda_2|), \end{aligned} \tag{4.1}$$

where the first line of Equation 4.1 is expressed in a computationally less demanding form as $\max(\lambda_1, \lambda_2)$ plus the correction function $f_c(\cdot)$. The correction function $f_c(x) = \ln(1 + e^{-x})$ has a dynamic range of $\ln(2) \geq f_c(x) > 0$, and it is significant only for small values of x [13]. Thus, $f_c(x)$ can be tabulated in a look-up table, in order to reduce the computational complexity [13]. The correction function $f_c(\cdot)$ only depends on $|\lambda_1 - \lambda_2|$, therefore the look-up table is one dimensional and experience shows that only few values have to be stored [99]. The Jacobian logarithmic relationship in Equation 4.1 can be extended also to cope with a

higher number of exponential summations, as in $\ln\left(\sum_{k=1}^{n} e^{\lambda_k}\right)$. Reference [13] showed that this can be achieved by nesting the $J(\lambda_1, \lambda_2)$ operation as follows:

$$\ln\left(\sum_{k=1}^{n} e^{\lambda_k}\right) = J(\lambda_n, J(\lambda_{n-1}, \ldots J(\lambda_3, J(\lambda_2, \lambda_1))\ldots)). \tag{4.2}$$

Having presented the Jacobian logarithmic relationship, we will now decribe, how this operation can be used to reduce the computational complexity of the RBF equaliser.

The overall response of the RBF network, given in Equation 2.81, is repeated here for convenience:

$$f_{RBF}(\mathbf{v}_k) = \sum_{i=1}^{M} w_i \exp(-\|\mathbf{v}_k - \mathbf{c}_i\|^2/\rho). \tag{4.3}$$

Expressing Equation 4.3 in a logarithmic form and substituting in the Jacobian logarithm, we obtain:

$$\begin{aligned}
\ln(f_{RBF}(\mathbf{v}_k)) &= \ln(\sum_{i=1}^{M} w_i \exp(-\|\mathbf{v}_k - \mathbf{c}_i\|^2/\rho)) \\
&= \ln(\sum_{i=1}^{M} \exp(\ln(w_i)) \exp(-\|\mathbf{v}_k - \mathbf{c}_i\|^2/\rho)) \\
&= \ln(\sum_{i=1}^{M} \exp(w_i' + \nu_{ik})) \\
&= \ln(\sum_{i=1}^{M} \exp(\lambda_{ik})) \\
&= J(\lambda_{Mk}, J(\lambda_{(M-1)k}, \ldots J(\lambda_{2k}, \lambda_{1k})\ldots)),
\end{aligned} \tag{4.4}$$

where $w_i' = \ln(w_i)$, which can be considered as a transformed weight. Furthermore, we used the shorthand $\nu_{ik} = -\|\mathbf{v}_k - \mathbf{c}_i\|^2/\rho$ and $\lambda_{ik} = \nu_{ik} + w_i'$. By introducing the Jacobian logarithm, every weighted summation of two exponential operations in Equation 4.3 is substituted with an addition, a subtraction, a table look-up and a max operation according to Equation 4.1, thus reducing the computational complexity. The term $\ln(\sum_{i=1}^{M} \exp(w_i' + \nu_{ik}))$ requires $3M - 1$ additions/subtractions, $M - 1$ table look-up and $M - 1$ $\max(\cdot)$ operations. Most of the computational load arises from computing the Euclidean norm term $\|\mathbf{v}_k - \mathbf{c}_i\|^2$, and the associated total complexity will depend on the number of RBF centres and on the dimension m of both the RBF centre vector $\mathbf{c}_i$ and the channel output vector $\mathbf{v}_k$. The evaluation of the term $\nu_{ik} = -\|\mathbf{v}_k - \mathbf{c}_i\|^2/\rho$ requires $2m - 1$ additions/subtractions, m multiplications and one division operation. Therefore, the computational complexity of a RBF DFE having m inputs and $n_{s,j}$ hidden RBF nodes per equalised output sample, which was previously given in Table 2.10, is now reduced to the values seen in Table 4.1 due to employing the Jacobian algorithm.

Determine the feedback state	
$n_{s,j}(2m+2)-2\mathcal{M}$	subtraction and addition
$n_{s,j}m$	multiplication
$n_{s,j}$	division
$n_{s,j}-\mathcal{M}+1$	max
$n_{s,j}-\mathcal{M}$	table look-up

Table 4.1: Computational complexity of a $\mathcal{M}$-ary Jacobian logarithmic decision feedback RBF network equaliser with m inputs and $n_{s,j}$ hidden units per equalised output sample based on Equations 2.104 and 4.4.

Exploiting the fact that the elements of the vector of noiseless channel outputs constituting the channel states $\mathbf{r}_i, i = 1, \ldots, n_s$ correspond to the convolution of a sequence of $(L+1)$ transmitted symbols and $(L+1)$ CIR taps – where these vector elements are referred to as the scalar channel states $r_l, l = 1, \ldots, n_{s,f} (= \mathcal{M}^{L+1})$ – we could use Patra's and Mulgrew's method [68] to reduce the computational load arising from evaluating the Euclidean norm ν_{ik} in Equation 4.4. Expanding the term ν_{ik} gives

$$\begin{aligned} \nu_{ik} &= \frac{-\|\mathbf{v}_k - \mathbf{c}_i\|^2}{\rho} \\ &= -\frac{(v_k - c_{i0})^2}{\rho} - \frac{(v_{k-1} - c_{i1})^2}{\rho} - \ldots \\ &\quad -\frac{(v_{k-j} - c_{ij})^2}{\rho} - \ldots - \frac{(v_{k-m+1} - c_{i(m-1)})^2}{\rho}, \\ &\qquad i = 1, \ldots, M, \qquad k = -\infty, \ldots, \infty, \end{aligned} \tag{4.5}$$

where v_{k-j} is the delayed received signal and c_{ij} is the jth component of the RBF centre vector $\mathbf{c}_i$, which takes the values of the scalar channel outputs $r_l, l = 1, \ldots, n_{s,f}$ as described in Section 2.10. Note from Equation 4.5 that ν_{ik} is a summation of the delayed components, $-\frac{(v_k - c_{ij})^2}{\rho}$ and the scalar centres c_{ij} take the values of the scalar channel outputs $r_l, l = 1, \ldots, n_{s,f}$. Thus, we could reduce the computational complexity of evaluating Equation 4.5 by pre-calculating $d_l = -\frac{(v_k - r_l)^2}{\rho}, l = 1, \ldots, n_{s,f}$ for all the $n_{s,f}$ possible values of the scalar channel outputs $r_l, l = 1, \ldots, n_{s,f}$ and storing the values. From Equation 4.5 the value of ν_{ik} can be obtained by summing the corresponding delayed values of d_l, which we will define as

$$d_{lj} = -\frac{(v_{k-j} - r_l)^2}{\rho}, \qquad l = 1, \ldots, n_{s,f}, \qquad j = 0, \ldots, m-1. \tag{4.6}$$

Substituting Equation 4.6 into Equation 4.5 yields:

$$\nu_{ik} = \sum_{\substack{j=0 \\ c_{ij} = r_l}}^{m-1} d_{lj}, \qquad i = 1, \ldots, M, \qquad k = -\infty, \ldots, \infty. \tag{4.7}$$

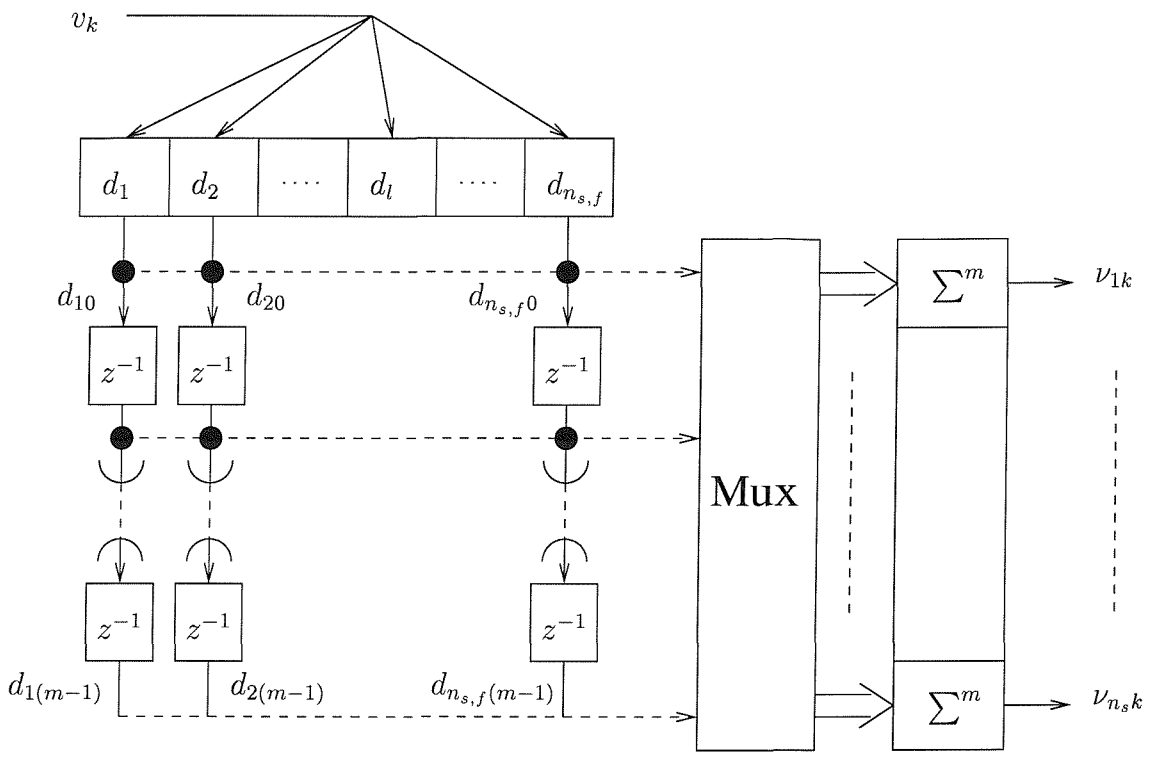


Figure 4.3: Reduced complexity computation of ν_{ik} in Equation 4.5 for substitution in Equation 4.4 based on scalar channel output.

The reduced complexity computation of ν_{ik} in Equation 4.7 for substitution in Equation 4.4 based on the scalar channel outputs r_l, can be represented as in Figure 4.3. The multiplexer (Mux) of Figure 4.3 maps d_{lj} of Equation 4.6 corresponding to the scalar centre r_l to the contribution of the vector centre's component c_{ij}.

The computation of $d_l = -\frac{(v_k - r_l)^2}{\rho}, l = 1, \ldots, n_{s,f}$ requires $n_{s,f}$ multiplication, division and subtraction operations. For every RBF centre vector $\mathbf{c}_i$, computing its corresponding ν_{ik} value according to Equation 4.7 needs $m - 1$ additions. The reduced computational complexity per equalised output sample of an $\mathcal{M}$-ary Jacobian DFE with m inputs, $n_{s,j} = \mathcal{M}^{m+L-n}$ hidden RBF nodes derived from $n_{s,f} = \mathcal{M}^{L+1}$ scalar centres is given in Table 4.2. Comparing Table 4.1 and 4.2, we observe a substantial computational complexity reduction, especially for a high feedforward order m, since $n_{s,f} < n_{s,j}$, if $m - n < 1$. For example, for the 16-QAM mode we have $n_{s,f} = 256$ and $n_{s,j} = 256$ for the RBF DFE equaliser parameters of $m = 2$, $n = 1$ and $\tau = 1$. The total complexity reduction is by a factor of about 1.3. If we increase the RBF DFE feedforward order and use the equaliser parameters of $m = 3$, $n = 1$ and $\tau = 2$ – which gives a better BER performance – then we have $n_{s,f} = 256$ and $n_{s,j} = 4096$ – and the total complexity reduction is by a factor of about 2.1. The computational complexity can be further reduced by neglecting the RBF scalar centres

situated far from the received signal v_k, since the contribution of RBF scalar centres r_l to the decision function is inversely related to their distance from the received signal v_k, as recognised by Patra [68].

Determine the feedback state	
$n_{s,j}(m+2) - 2\mathcal{M} + n_{s,f}$	subtraction and addition
$n_{s,f}$	multiplication
$n_{s,f}$	division
$n_{s,j} - \mathcal{M} + 1$	max
$n_{s,j} - \mathcal{M}$	table look-up

Table 4.2: Reduced computational complexity per equalised output sample of an $\mathcal{M}$-ary Jacobian logarithmic RBF DFE based on scalar centres. The Jacobian RBF DFE based on Equation 2.104 and 4.4 has m inputs and $n_{s,j}$ hidden RBF nodes, which are derived from the $n_{s,f}$ number of scalar centres.

Figures 4.4 and 4.5 show the BER versus SNR performance comparison of the RBF DFE and the Jacobian logarithmic RBF DFE over the two-path Gaussian channel and two-path Rayleigh fading channel of Table 3.1, respectively. For the simulation of the Jacobian logarithmic RBF DFE the correction function $f_c(\cdot)$ in Equation 4.1 was approximated by a pre-computed table having eight stored values ranging from 0 to ln(2). From these results we concluded that the Jacobian logarithmic RBF equaliser's performance was equivalent to that of the RBF equaliser, whilst having a lower computational complexity.

Having presented the proposed reduced complexity Jacobian logarithmic RBF equaliser, we will now proceed to introduce the joint RBF equalisation and turbo coding system and investigate its performance in both fixed QAM and burst-by-burst (BbB) AQAM schemes.

4.3 System Overview

The structure of the joint RBF DFE and turbo decoder is portrayed in Figure 4.6. The output of the RBF DFE provides the *a posteriori* LLRs of the transmitted bits based on the *a posteriori* probability of each legitimate $\mathcal{M}$-QAM symbol. The *a posteriori* LLR of a data bit u_k is denoted by $\mathcal{L}(u_k|\mathbf{v}_k)$, which was defined as the log of the ratio of the probabilities of the bit being a logical 1 or a logical 0, conditioned on the received sequence $\mathbf{v}_k$:

$$\begin{aligned} \mathcal{L}(u_k|\mathbf{v}_k) &= \ln\left(\frac{P(u_k=+1|\mathbf{v}_k)}{P(u_k=-1|\mathbf{v}_k)}\right), \\ &= L(u_k=+1|\mathbf{v}_k) - L(u_k=-1|\mathbf{v}_k), \end{aligned} \tag{4.8}$$

where the term $L(u_k = \pm 1|\mathbf{v}_k) = ln(P(u_k = \pm 1|\mathbf{v}_k))$ is the log-likelihood of the data bit u_k having the value ± 1 conditioned on the received sequence $\mathbf{v}_k$.

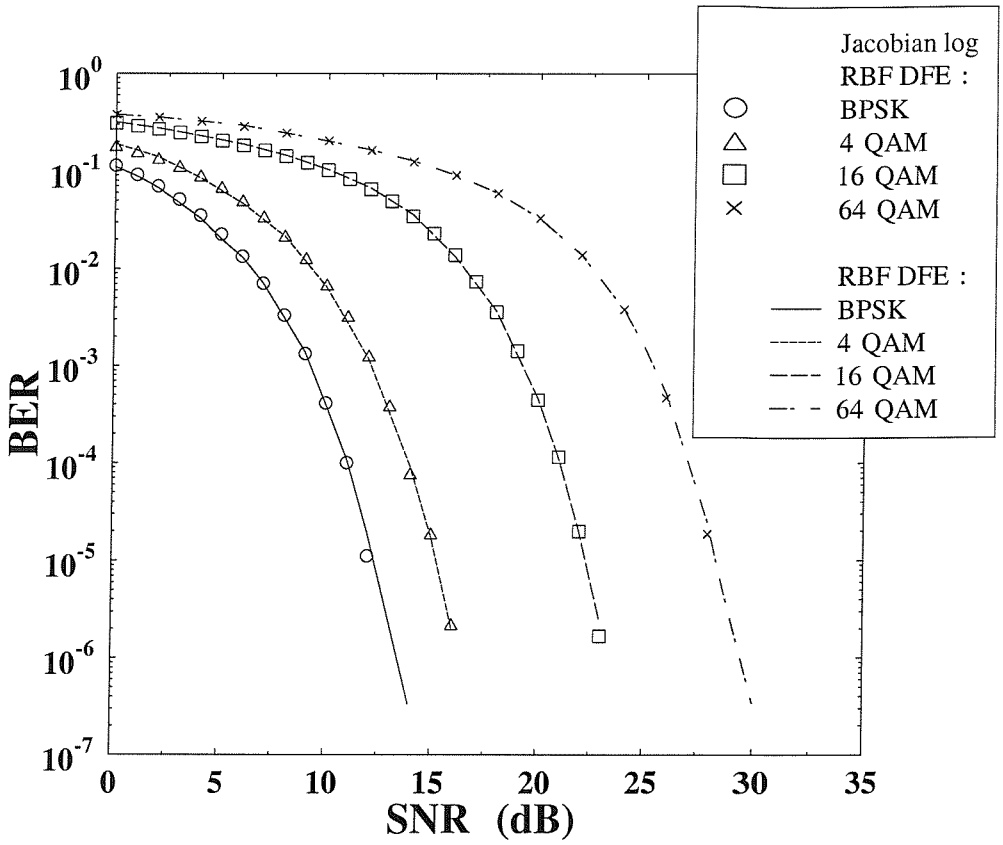


Figure 4.4: BER versus signal to noise ratio performance of the RBF DFE and the Jacobian logarithmic RBF DFE over the dispersive **two-path Gaussian channel** of Figure 2.21(a) for different $\mathcal{M}$-QAM modes. Both equalisers have a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol.

The LLR of the bits representing the QAM symbols can be obtained from the *a posteriori* log-likelihood of the symbol. Below we provide an example for the 4-QAM mode of our AQAM scheme. The *a posteriori* log-likelihood L_1, L_2, L_3 and L_4 of the four possible 4-QAM symbols is given by the Jacobian RBF networks. A 4-QAM symbol is denoted by the bits U_0U_1 and the symbols $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{I}_3$ and $\mathcal{I}_4$ correspond to 00, 01, 10 11, respectively. Thus, the *a posteriori* LLRs of the bits are obtained as follows:

$$\begin{aligned} \mathcal{L}(U_0|\mathbf{v}_k) &= L(U_0 = 1|\mathbf{v}_k) - L(U_0 = 0|\mathbf{v}_k), \\ \mathcal{L}(U_1|\mathbf{v}_k) &= L(U_1 = 1|\mathbf{v}_k) - L(U_1 = 0|\mathbf{v}_k), \end{aligned} \tag{4.9}$$

where,

$$L(U_0 = 1|\mathbf{v}_k) = L(U_0U_1 = 11 \cup U_0U_1 = 10|\mathbf{v}_k) = \ln(e^{P(U_0U_1=11|\mathbf{v}_k)\cdot P(U_0U_1=10|\mathbf{v}_k)}) = J(L_4, L_3),$$

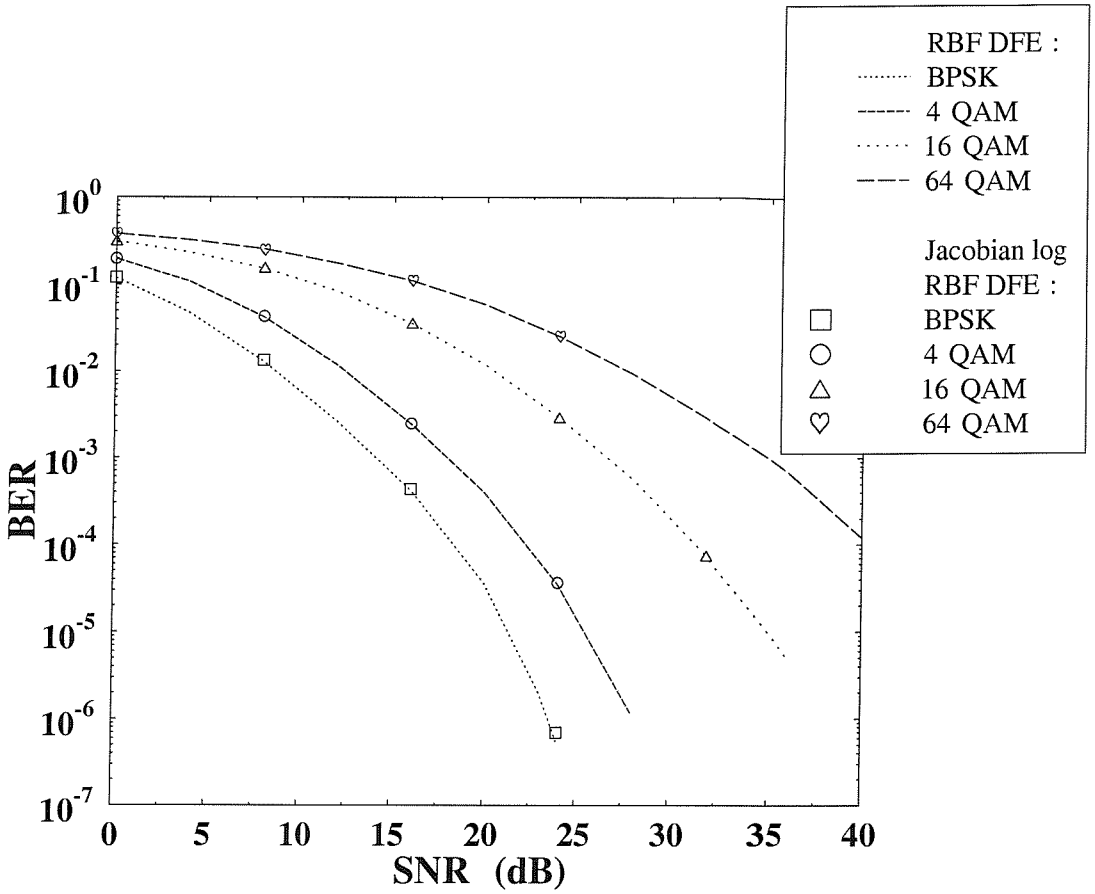


Figure 4.5: BER versus signal to noise ratio performance of the RBF DFE and the Jacobian logarithmic RBF DFE over the **two path equal weight, symbol-spaced Rayleigh fading channel** of Table 3.1 for different $\mathcal{M}$-QAM modes. Both equalisers have a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol. Correct symbols were fed back.

$$
\begin{aligned}
L(U_0 = 0|\mathbf{v}_k) &= L(U_0U_1 = 01 \cup U_0U_1 = 00|\mathbf{v}_k) = \ln(e^{P(U_0U_1=01|\mathbf{v}_k)\cdot P(U_0U_1=00|\mathbf{v}_k)}) = J(L_2, L_1), \\
L(U_1 = 1|\mathbf{v}_k) &= L(U_0U_1 = 11 \cup U_0U_1 = 01|\mathbf{v}_k) = \ln(e^{P(U_0U_1=11|\mathbf{v}_k)\cdot P(U_0U_1=01|\mathbf{v}_k)}) = J(L_4, L_2), \\
L(U_1 = 0|\mathbf{v}_k) &= L(U_0U_1 = 10 \cup U_0U_1 = 00|\mathbf{v}_k) = \ln(e^{P(U_0U_1=10|\mathbf{v}_k)\cdot P(U_0U_1=00|\mathbf{v}_k)}) = J(L_3, L_1),
\end{aligned}
\tag{4.10}
$$

and $J(\lambda_1, \lambda_2)$ denotes the Jacobian logarithmic relationship of Equation 4.1.

Note that the Jacobian RBF equaliser will provide $\log_2(\mathcal{M})$ number of LLR values for every $\mathcal{M}$-QAM symbol. These value are fed to the turbo decoder as its soft inputs. The turbo decoder will iteratively improve the BER of the decoded bits and the detected bits will be constituted by the sign of the turbo decoder's soft output.

The probability of error for the detected bit can be estimated on the basis of the soft

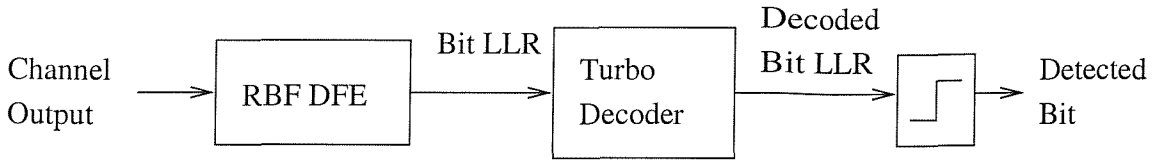


Figure 4.6: Joint RBF DFE and turbo decoder schematic

output of the turbo decoder. Referring to Equation 4.8 and assuming $P(u_k = +1|\mathbf{v}_k) + P(u_k = -1|\mathbf{v}_k) = 1$, the probability of error for the detected bit is given by

$$P_{error}(u_k) = \begin{cases} 1 - P(u_k = +1|\mathbf{v}_k) = P(u_k = -1|\mathbf{v}_k), & \text{if } \mathcal{L}(u_k|\mathbf{v}_k) \geq 0 \\ 1 - p(u_k = -1|\mathbf{v}_k) = P(u_k = +1|\mathbf{v}_k), & \text{if } \mathcal{L}(u_k|\mathbf{v}_k) < 0 \end{cases}. \quad (4.11)$$

With the aid of the definition in Equation 4.8 the probability of the bit having the value of +1 or -1 can be rewritten in terms of the *a posteriori* LLR of the bit, $\mathcal{L}(u_k|\mathbf{v}_k)$ as follows:

$$\begin{aligned} P(u_k = +1|\mathbf{v}_k) &= \frac{1}{1 + e^{-\mathcal{L}(u_k|\mathbf{v}_k)}}, \\ P(u_k = -1|\mathbf{v}_k) &= \frac{1}{1 + e^{\mathcal{L}(u_k|\mathbf{v}_k)}}. \end{aligned} \quad (4.12)$$

Upon substituting Equation 4.12 into Equation 4.11, we redefined the probability of error of a detected bit in terms of its LLR as:

$$P_{error}(u_k) = \frac{1}{1 + e^{|\mathcal{L}(u_k|\mathbf{v}_k)|}}, \quad (4.13)$$

where $|\mathcal{L}(u_k|\mathbf{v}_k)|$ is the magnitude of $\mathcal{L}(u_k|\mathbf{v}_k)$. Again, the average short-term probability of bit error within the decoded burst is given by:

$$P_{\text{bit, short-term}} = \frac{\sum_{i=0}^{L_b} P_{error}(u_i)}{L_b}, \quad (4.14)$$

where L_b is the number of decoded bits per transmitted burst and u_i is the ith decoded bit in the burst. This value, which we will refer to as the *estimated short-term BER* was found to give a good estimation of the actual BER of the burst, which will be demonstrated in Section 4.4. The actual BER is the ratio of the number of bit errors encountered in a data burst to the total number of bits transmitted in that burst.

In the next section we will investigate the performance of the turbo-coding assisted RBF DFE $\mathcal{M}$-QAM scheme based on our simulation results.

4.4 Turbo-coded RBF-equalized $\mathcal{M}$-QAM Performance

According to our BER versus BPS optimistion approach high code rates in excess of 2/3 are desirable, in order to maximise the BPS throughput of the system. Consequently, block

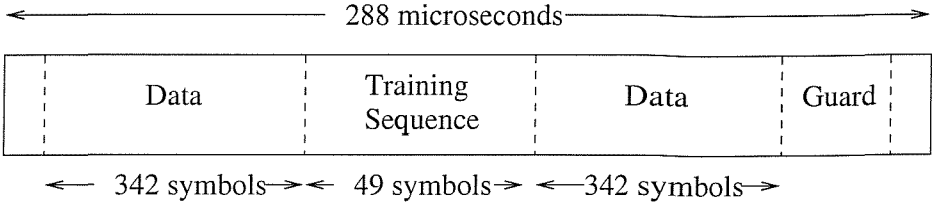


Figure 4.7: Transmission burst structure of the so-called FMA1 nonspread data mode as specified in the FRAMES proposal [101].

codes were favoured as the turbo component codes in preference to the more widely used Recursive Systematic Convolutional (RSC) code based turbo-coded benchmarker scheme, since turbo block coding has been shown to perform better for coding rates in excess of 2/3 [100]. This is demonstrated first in Figure 4.11, which will be discussed in more depth at a later stage. In our simulations, unless otherwise stated, we hence utilized the turbo coding parameters given in Table 4.3 and employed the transmission burst structure shown in Figure 4.7. The turbo encoder used two Bose-Chaudhuri-Hocquenghem BCH(31, 26) block codes in parallel. A 9984-bit random interleaver was used between the two component codes, unless otherwise stated. We used the Log-MAP decoder [13] throughout our simulations, since it offered the same performance as the optimal MAP decoder with a reduced complexity. The DFE used correct symbol feedback and we assumed perfect CIR estimation, hence the associated results indicate the system's upper-bound performance.

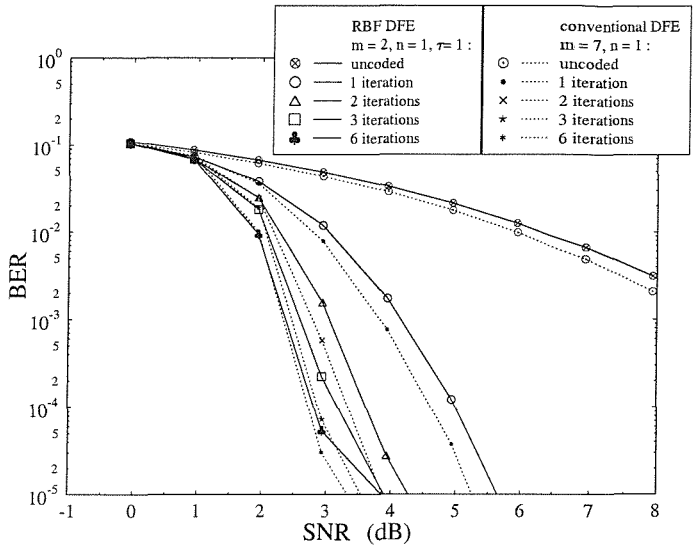
	BCH	RSC
Component code	BCH(31,26)	$K = 3$, $n = 2$, $k = 1$
Octal generator polynomial		$G[0] = 7_8$ $G[1] = 5_8$
Code rate, R	$0.72 = \frac{26}{36}$ [1]	0.75
Turbo interleaver type	Random	Random
Turbo interleaver size	9984-bit	9984-bit
Component decoders	Log-MAP	Log-MAP

Table 4.3: The turbo BCH and RSC coding parameters

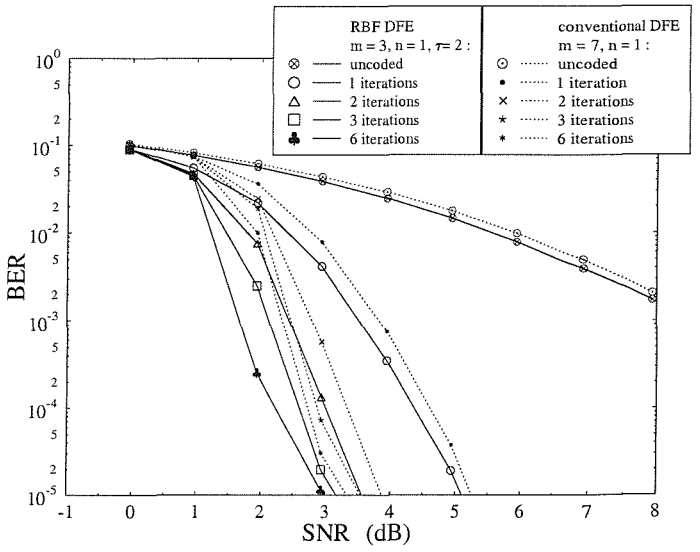
4.4.1 Results over Dispersive Gaussian Channels

We will first investigate the performance of the joint RBF DFE $\mathcal{M}$-QAM and turbo coding scheme over the two-path Gaussian channel of Figure 2.21(a). Figure 4.8 provides our BER performance comparison between the RBF DFE scheme and the conventional DFE scheme in conjunction with the turbo BCH codec of Table 4.3. The RBF DFE has a feedforward

[1] The parity bits were not punctured, since block turbo codes suffer from performance loss upon puncturing



(a) RBF DFE with feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol.



(b) RBF DFE with feedforward order of $m = 3$, feedback order of $n = 1$ and decision delay of $\tau = 2$ symbol.

Figure 4.8: BER versus SNR performance for the BPSK RBF DFE and for a conventional DFE using the turbo BCH codec of Table 4.3 with different number of iterations over the dispersive two-path Gaussian channel of Figure 2.21(a). The conventional DFE has a feedforward order of $m = 7$ and a feedback order of $n = 1$. The turbo interleaver size is 9984 bits.

order of 2, feedback order of 1 and decision delay of 1 symbol in Figure 4.8(a) and a feedforward order of 3, feedback order of 1 and decision delay of 1 symbol for Figure 4.8(b). The parameters of the conventional DFE were a feedforward order of 7 and feedback order of 1, which were assigned such that they gave the best possible BER performance according to our experiments and hence there was no significant BER improvement upon increasing the feedforward and feedback order. Figure 4.8 also demonstrates the effect of the number of decoding iterations used. The performance of the uncoded scheme is also provided as a comparison. Using turbo coding improves the performance by approximately 3.2dB at a BER of 10^{-2} for both the RBF DFE ($m = 2, \tau = 1$ and $m = 3, \tau = 2$) and for conventional DFE schemes. As the number of iterations used by the turbo decoder increases, both the turbo-coded RBF DFE and the turbo-coded conventional DFE scheme perform significantly better. However, the 'per-iteration' BER improvement is reduced, as the number of iterations increases. Hence, for complexity reasons, the number of decoding iterations was set to six for our forthcoming simulations.

Figure 4.8(a) indicates that the turbo-coded conventional DFE scheme performs slightly better than the turbo-coded RBF DFE ($m = 2, \tau = 1$) scheme, corresponding to approximate improvements of 0.5dB, 0.3dB and 0.1dB for one iteration, three iterations and six iterations, respectively, at a BER of 10^{-4}. However, the performance of the turbo-coded RBF DFE scheme can be further improved by increasing its feedforward order and decision delay, as demonstrated in Figure 4.8(b), unlike that of the turbo-coded conventional DFE where there is no further performance improvement upon increasing the equaliser order. The improved turbo-coded RBF DFE ($m = 3, \tau = 2$) scheme gives an SNR improvement of 0.2dB, 0.2dB and 0.5dB for one iteration, three iterations and six iterations, respectively, at a BER of 10^{-4} compared to the conventional DFE scheme. The SNR improvement at a BER of 10^{-4} compared to the uncoded conventional DFE is -0.5dB and 0.2dB for the RBF DFE using $m = 2, \tau = 1$ and $m = 3, \tau = 2$, respectively. We observed that the turbo-coded performance of the conventional DFE and RBF DFE follow the trends of their uncoded performances.

We will now extend our investigations to QAM schemes. Figure 4.9 shows the BER performance of the BCH turbo-coded RBF DFE system for various QAM modes over the two-path Gaussian channel. Introducing turbo coding into the system improves the performance by 8dB for BPSK, 4-QAM and 16-QAM and by about 9.5dB for 64-QAM at a BER of 10^{-4}. Note that turbo coding only starts to improve the uncoded performance after the uncoded BER drops below 10^{-1}, since coding could not improve the BER performance, if the number of errors in the undecoded burst exceeded a certain limit.

The Jacobian logarithmic RBF DFE introduced in Section 4.2 can be used to substitute

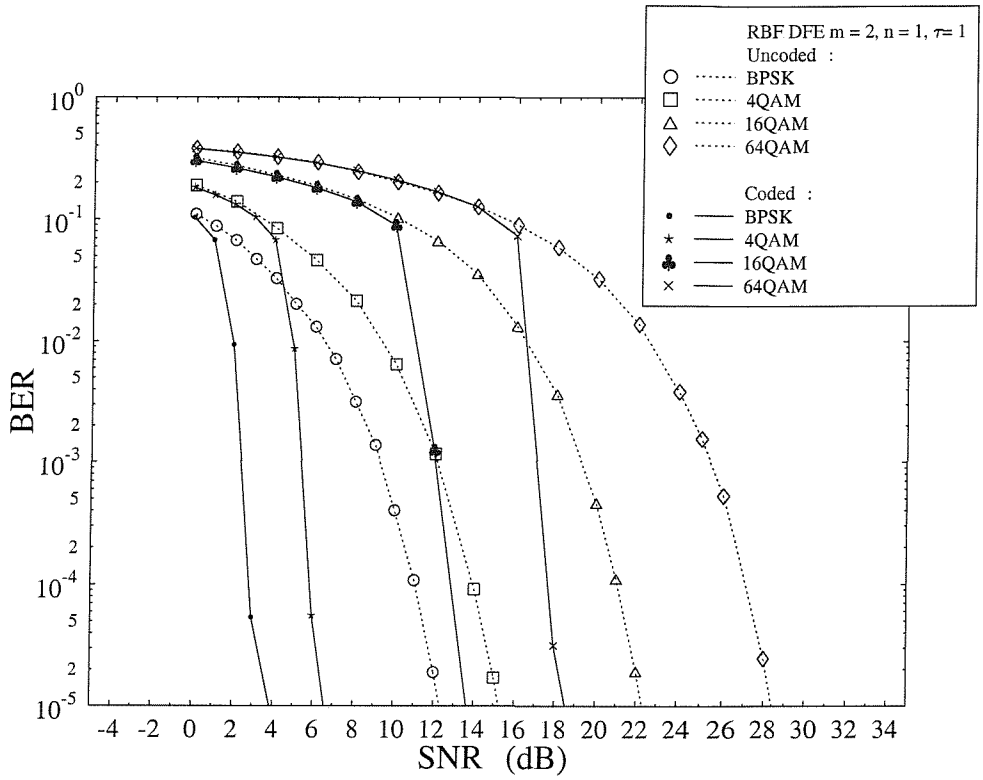


Figure 4.9: BER versus SNR performance for the RBF DFE using the turbo codec of Table 4.3 over **the dispersive two-path Gaussian channel** of Figure 2.21(a) in conjunction with various QAM modes. The RBF DFE has a feedforward order of $m = 2$, feedback order of $n = 1$ and a decision delay of $\tau = 1$ symbol. The number of turbo BCH(31,26) decoder iterations is six, while the random turbo interleaver size is 9984 bits.

the RBF DFE in order to reduce the computational complexity of the system. The turbo-coded performance of the Jacobian logarithmic RBF DFE is shown to be similar to that of the RBF DFE in Figure 4.10, since the Jacobian logarithmic algorithm is capable of giving a good approximation of the equalised channel output LLRs.

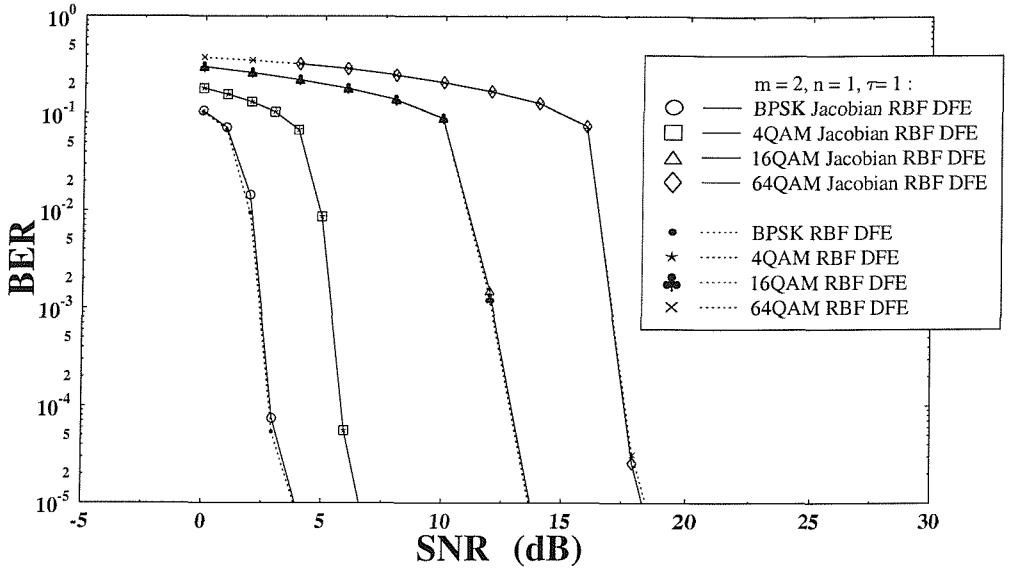


Figure 4.10: BER versus SNR performance for the RBF DFE and Jacobian logarithmic RBF DFE using the turbo codec of Table 4.3 over the dispersive two-path Gaussian channel of Figure 2.21(a) in conjunction with various QAM modes. The equaliser has a feedforward order of $m = 2$, feedback order of $n = 1$ and a decision delay of $\tau = 1$ symbol. The number of turbo BCH(31,26) decoder iterations is six, while the random turbo interleaver size is 9984 bits.

4.4.2 Results over Dispersive Fading Channels

We will now investigate the performance of the joint RBF DFE $\mathcal{M}$-QAM and turbo coding scheme over the wideband Rayleigh fading channel environment of Table 4.4, while the parameters of the turbo codec are given in Table 4.3.

As noted before, Figure 4.11 shows the performance of the Jacobian RBF DFE in conjunction with both BCH and RSC based turbo coding for various QAM modes. The BCH turbo-coded scheme improves the system performance by 5dB, 4dB, 7dB and 8dB using BPSK, 4-QAM, 16-QAM and 64-QAM, respectively, for a BER of 10^{-4}. By contrast, for the RSC turbo coded scheme the BER performance improves by 2dB for BPSK and 4-QAM, while 3dB for 16-QAM and 64-QAM. Similarly to the 2-path Gaussian channel, the

turbo-coded schemes only start to provide significant BER improvements with respect to the uncoded scheme, once the uncoded BER dips below 10^{-1}. Our performance comparison with the turbo convolutional codec of Table 4.3 given in Figure 4.11 demonstrates that the $R = 0.72$ turbo block code provides a better BER performance than the $R = 0.75$ RSC-turbo codec, at the cost of a higher computational complexity. As seen in Table 4.3, a half rate RSC encoder of constraint lenght $K = 3$ was used in the RSC turbo codec. The generator polynomials expressed in octal terms were set to seven (for the feedback path) and five. Similarly to the turbo BCH codec, the code rate was set to 0.75 by applying a random puncturing pattern in the RSC encoder. The turbo interleaver depth was also chosen to be 9984 bits.

Transmission Frequency	1.9GHz
Transmission Rate	2.6MBd
Vehicular Speed	30 mph
Normalised Doppler Frequency	3.3×10^{-5}
Channel weights	$0.707 + 0.707z^{-1}$

Table 4.4: Simulation parameters for the two-path Rayleigh fading channel

Modulation Mode	BPSK	4-QAM	16-QAM	64-QAM
Interleaver Size	494	988	1976	2964

Table 4.5: Corresponding random interleaver sizes for each modulation mode

4.5 Channel Quality Measure

In order to identify the potentially most reliable channel quality measure to be used in our BbB adaptive turbo-coded QAM modems to be designed during our forthcoming discourse, we will now analyse the relationship between the average burst LLR magnitude before and after channel decoding. For this reason, the random turbo interleaver size was reduced from the previously used 9984 bits and it was varied on a BbB basis, corresponding to the modulation mode used, as shown in Table 4.5, in order to enable BbB decoding so that we could obtain the average burst LLR magnitude of the coded data burst corresponding to the uncoded data burst. Explicitly, the interleaver size is set to be equivalent to the number of source bits in a data burst, in order to enable BbB decoding. Since the code rate is 0.72 and the number of coded bits is 684, 1368, 2736 and 4104 for BPSK, 4-QAM, 16-QAM and 64-QAM, respectively, for a burst length of 684 symbols, the interleaver size (= number of

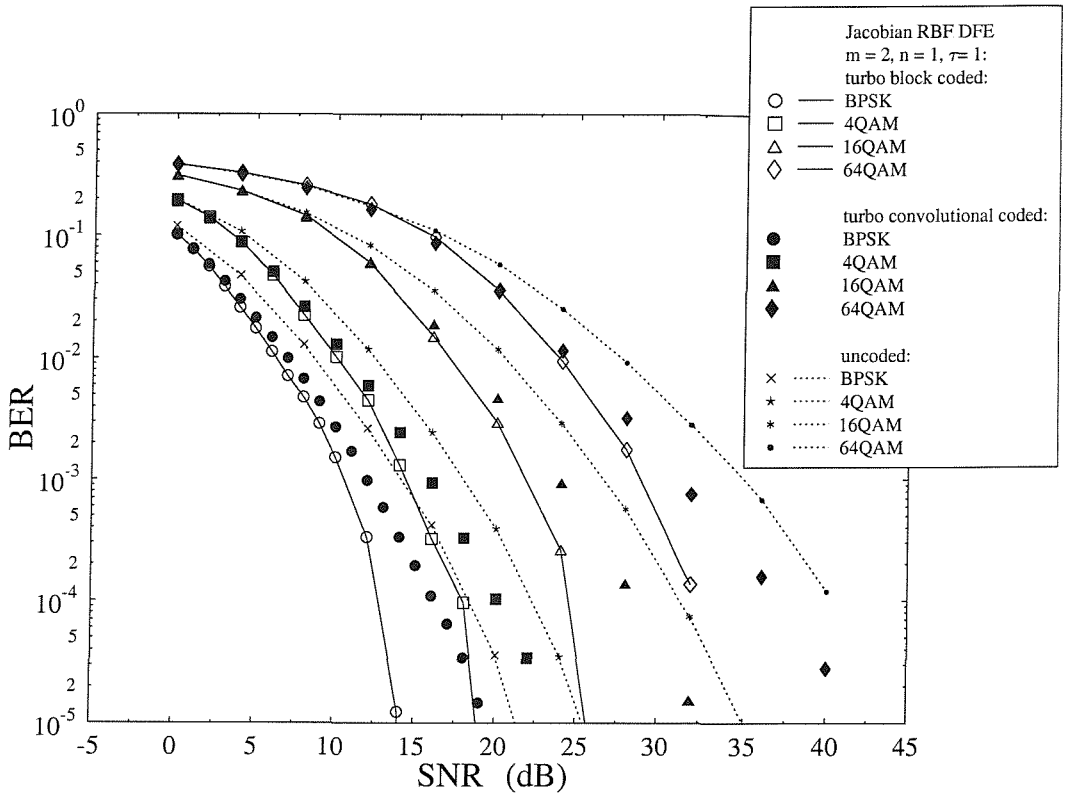


Figure 4.11: BER versus SNR performance for the Jacobian logarithmic RBF DFE using the turbo codec of Table 4.3 over **the dispersive two-path fading channel** of Table 4.4 for various QAM modes. The equaliser has a feedforward order of $m = 2$, feedback order of $n = 1$ and a decision delay of $\tau = 1$ symbol. The number of **convolutional and BCH** turbo decoder iterations is six, while the turbo interleaver size is fixed to 9984 bits.

source bits = number of coded bits - number of parity bits) is as shown in Table 4.5. The average burst LLR magnitude is defined as follows:

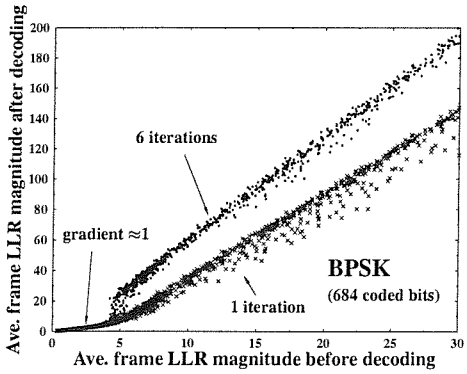
$$\mathcal{L}_{\text{average}} = \frac{\sum_{i=0}^{L_b} |\mathcal{L}(u_i|\mathbf{v}_k)|}{L_b}, \tag{4.15}$$

where L_b is the number of data bits per transmitted burst and u_i is the ith data bit in the burst. Figure 4.12 shows the improvement of the average burst LLR magnitude after turbo decoding for the turbo BCH codec of Table 4.3 over the wideband Rayleigh fading channel environment of Table 4.4. As seen in the figure, the gradient of the curve is approximately unity for the average burst LLR magnitude before decoding over the range of 0 to 5 for BPSK and 4-QAM, 0 to 6 for 16-QAM and 0 to 10 for 64-QAM. Thus, there is no average LLR magnitude improvement upon introducing turbo decoding in this low reliability range. This is in harmony with our previous observations in Figures 4.10 and 4.11, namely that there is no BER improvement for BERs below 10^{-1}. Beyond this range, there is a sharp increase in the decoded LLR magnitude due to turbo decoding. Figure 4.12(a) also shows the effect of increasing the number of decoder iterations on the average burst LLR magnitude. Increasing the number of decoder iterations improves not only the BER, but also the average confidence measure of the decoder's decisions.

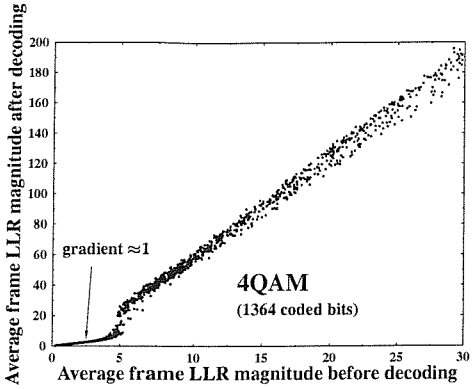
Figure 4.13 shows the relationship between the estimated short-term BER defined in Equation 4.14 and the average burst LLR magnitude after turbo decoding using six iterations. Note that the curves becomes more 'spread out', as the short-term BER decreases.

This is because the relationship between the probability of bit error in the decoded burst expressed in the logarithmic domain is inversely proportional to its LLR magnitude, as shown in Figure 4.14. The average of the burst LLR magnitude is dominated by the LLR values of the bits having lower probability of bit error, whereas the short-term BER of the burst is dominated by the bits with higher probability of bit error. The variance of the LLR values of the bits in the burst accounts for the 'spread' of the the estimated short-term BER versus average burst LLR magnitude curves in Figure 4.13 at low short-term BER values.

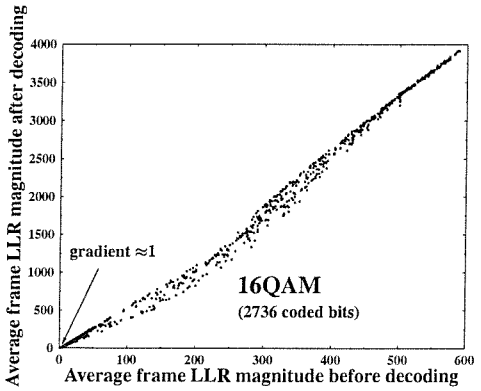
Since the average burst LLR magnitude is related to the estimated short-term BER, after accounting for the 'spread' at low short-term BERs, the average burst LLR magnitude can be used as the modem mode switching metric in our AQAM scheme, which will be discussed in Section 4.6. The average burst LLR magnitude is preferred instead of the short-term BER as the modem mode switching metric, because it can avoid the extra computational complexity of having to convert the output of the RBF DFE and the turbo decoder from the LLR values to BER values according to Equation 4.13, in order to obtain the short-term BER of the data burst.



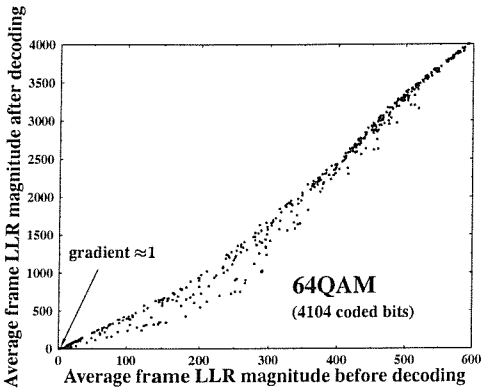
(a) BPSK (1 and 6 iterations)



(b) 4-QAM (6 iterations)

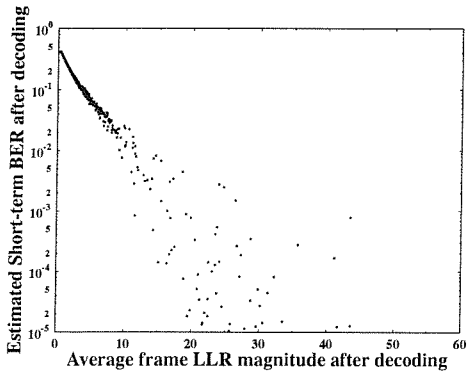


(c) 16-QAM (6 iterations)

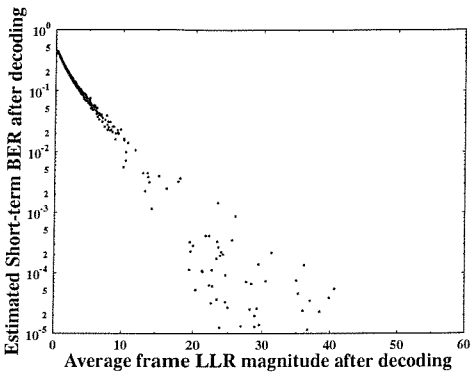


(d) 64-QAM (6 iterations)

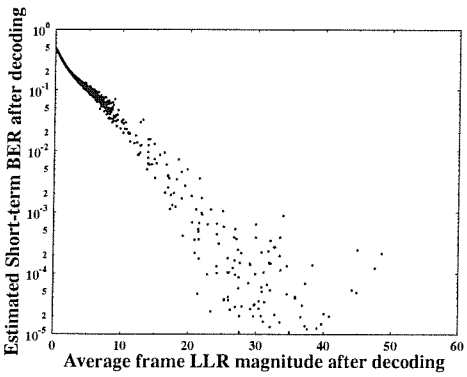
Figure 4.12: The average burst LLR magnitude after turbo decoding versus the average burst LLR magnitude before turbo decoding using BbB interleaving and turbo BCH decoding employing the parameters of Table 4.3 over the burst-invariant two-path fading channel of Table 4.4.



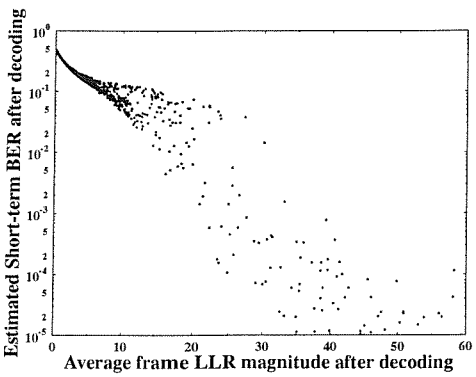
(a) BPSK



(b) 4-QAM



(c) 16-QAM



(d) 64-QAM

Figure 4.13: The estimated short-term BER versus the average burst LLR magnitude after turbo BCH(31,26) decoding using six iterations over the burst-invariant two-path fading channel of Table 4.4.

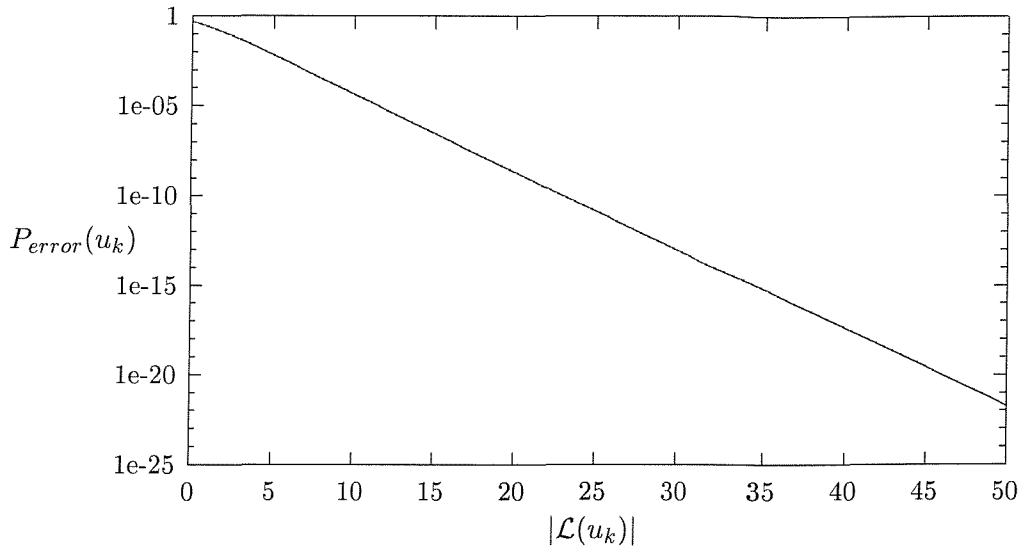


Figure 4.14: The probability of error of the detected bit versus the magnitude of its LLR.

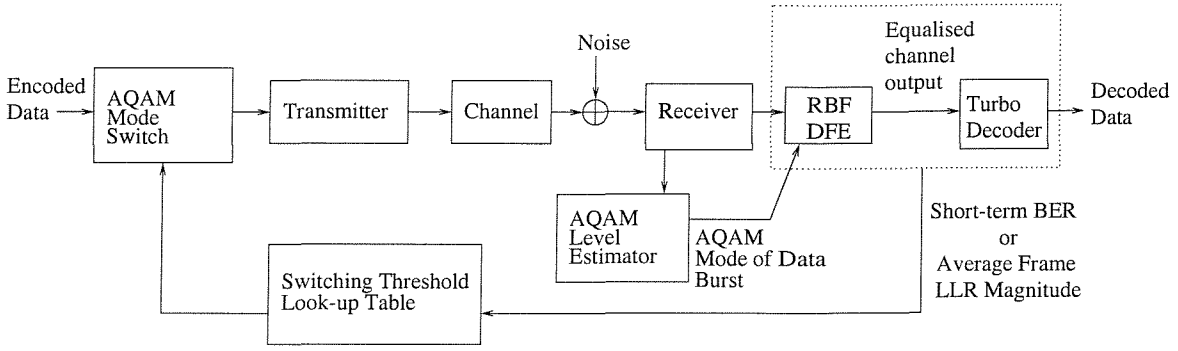


Figure 4.15: System schematic of the joint adaptive modulation and RBF equaliser scheme using turbo coding

4.6 Turbo Coding and RBF Equaliser Assisted AQAM

4.6.1 System Overview

The schematic of the joint AQAM and RBF network based equalisation scheme using turbo coding is depicted in Figure 4.15. The switching thresholds can be based on the switching metric either before or after turbo decoding. In this section we will investigate the performance of the AQAM scheme using either the short-term BER or the average burst LLR magnitude as our switching metric.

For our experiments in the following sections, the simulation parameters are listed in Table 4.4, noting that we analysed the joint AQAM and RBF DFE scheme in conjunction with turbo coding over the two-path Rayleigh fading channel of Table 4.4. The wideband fading channel was burst-invariant, implying that during a transmission burst the channel

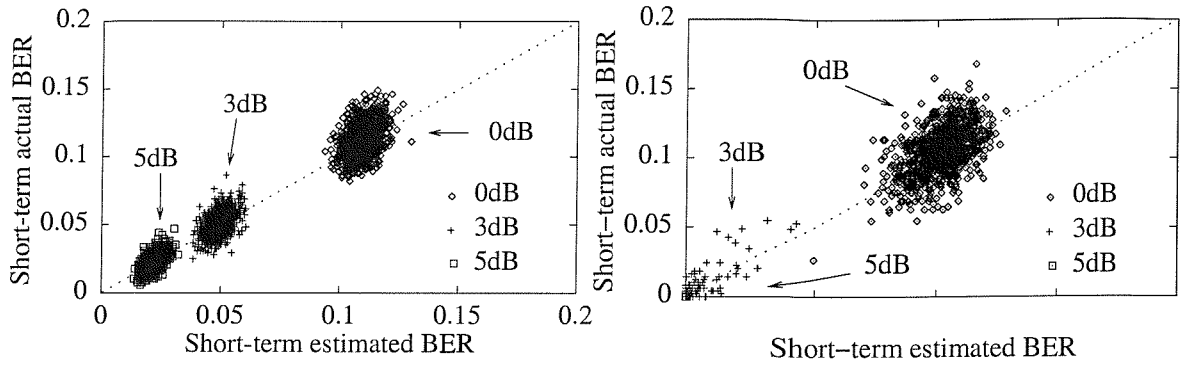
impulse response was considered time-invariant. In our simulations, we used the Jacobian RBF DFE of Section 4.2, which gave a similar turbo-coded BER performance to the RBF DFE but at a lower computational complexity, as it was demonstrated in Figure 4.10. The Jacobian RBF DFE had a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$. We used the BCH(31, 26) code of Table 4.3 as the turbo component code and the BbB random interleavers depending on the modulation mode were employed, as given in Table 4.5. The modulation modes utilized in our system are BPSK, 4-QAM, 16-QAM, 64-QAM and NO TX.

4.6.2 Performance of the AQAM Jacobian RBF DFE Scheme: Switching Metric Based on the Short-Term BER Estimate

Following from Section 3.4, where the *uncoded* AQAM RBF DFE scheme used the estimated short-term BER to switch the modem mode, we will now investigate the performance of the *turbo-coded* AQAM RBF DFE scheme based on the same switching metric. The estimated short-term BER can be obtained both before or after turbo BCH(31,26) decoding for the coded system. The estimated short-term BER before decoding can be obtained with the aid of the RBF DFE based on Equation 3.15, while that after turbo decoding can be obtained with the aid of the decoder based on Equation 4.14.

The plot of the estimated BER versus actual BER before and after turbo BCH(31,26) decoding and their corresponding PDFs of the BER estimation error for the Jacobian RBF DFE and for various channel SNRs is shown in Figures 4.16, 4.17, 4.18 and 4.19, for BPSK transmission bursts over the dispersive two-path Gaussian channel of Figure 2.21(a) and the two-path Rayleigh fading channel of Table 4.4, respectively. The actual burst-BER is the ratio of the number of bit errors encountered in a data burst to the total number of bits transmitted in that burst. The figures suggest that the Jacobian RBF DFE and the turbo BCH(31,26) decoder provide a good BER estimation, especially at higher channel SNRs. We note, however again that the accuracy of the actual BER evaluation is limited by the burst-length of 684 bits and 494 bits for the undecoded and decoded bursts, respectively. Therefore, for high SNRs the actual number of errors registered is often 0, which portrays the estimation algorithm in a less accurate light in the PDF of Figure 4.18 and 4.19 than it is in reality, since the 'resolution' of the reference BER is $1/684$ or $1/494$.

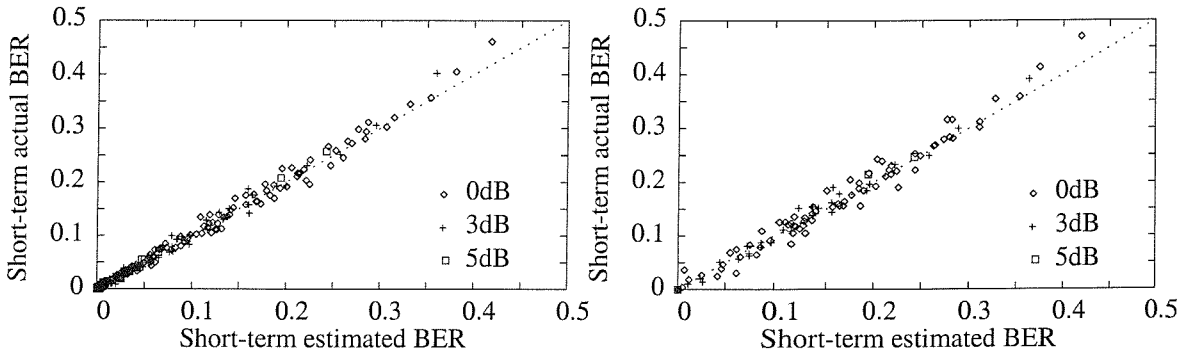
We shall refer to the AQAM scheme that utilised the switching thresholds based on the short-term BER before and after decoding, 'before decoding'-scheme and 'after decoding'-scheme, respectively. The short-term BER $P_{\text{bit, short-term}}$, obtained from either the RBF DFE or the turbo BCH(31,26) decoder is compared to a set of switching BER thresholds, $P_i^{\mathcal{M}}, i = 2, 4, 16, 64$, corresponding to the various $\mathcal{M}$-QAM modes, and the modulation



(a) Before decoding.

(b) After decoding.

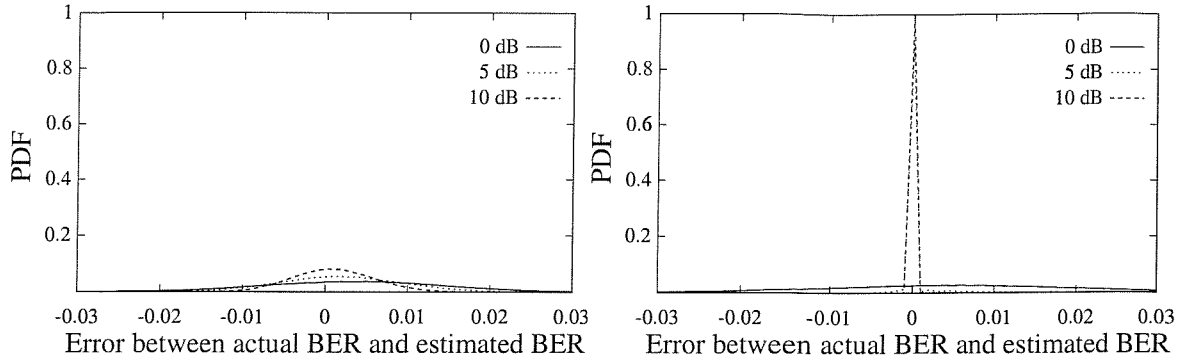
Figure 4.16: The actual BER versus estimated BER before and after turbo BCH(31,26) decoding with the error PDF given in Figure 4.18 for the **dispersive two-path Gaussian channel** of Figure 2.21(a) using BPSK. The number of turbo BCH(31,26) decoder iterations is six while the random turbo interleaver size is 494.



(a) Before decoding.

(b) After decoding.

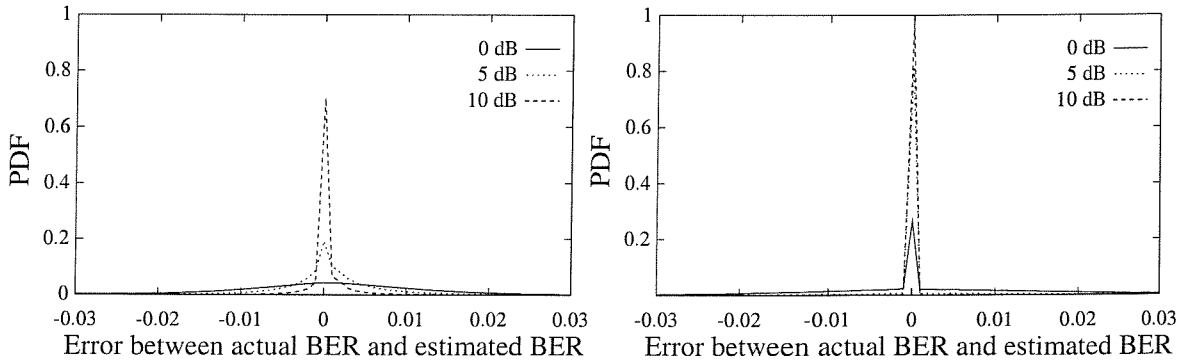
Figure 4.17: The actual BER versus estimated BER before and after turbo BCH(31,26) decoding with error PDF given in Figure 4.19 for the **dispersive two-path Rayleigh fading channel** of Table 4.4 using BPSK. The number of turbo decoder iterations is six, while the turbo interleaver size is 494.



(a) Before turbo BCH(31,26) decoding.

(b) After turbo BCH(31,26) decoding.

Figure 4.18: Discretised PDF of the error between the actual BER of BPSK bursts and the BER estimated by the Jacobian RBF DFE before and after turbo BCH(31,26) decoding for the **dispersive two-path AWGN channel** of Figure 2.21(a) using BPSK. The number of turbo BCH(31,26) decoder iterations is six, while the random turbo interleaver size is 494.



(a) Before turbo BCH(31,26) decoding.

(b) After turbo BCH(31,26) decoding.

Figure 4.19: Discretised PDF of the error between the actual BER of BPSK bursts and the BER estimated by the Jacobian RBF DFE before and after turbo BCH(31,26) decoding for the **two-path Rayleigh fading channel** of Table 4.4 using BPSK. The number of turbo decoder iterations is six, while the turbo interleaver size is 494.

mode is switched according to Equation 3.7.

As discussed in Section 3.4, the switching BER thresholds can be obtained by estimating the BER degradation/improvement, when the modulation mode is switched from $\mathcal{M}$-QAM to a higher/lower value of $\mathcal{M}$. We obtain this BER degradation/improvement measure from the estimated short-term BER of every modulation mode used under the same channel scenario.

In our experiments used to obtain the switching BER thresholds, pseudo-random symbols were transmitted in a fixed-length burst for all modulation modes across the burst-invariant wideband channel. The receiver receives each data burst having different modulation modes, equalises and turbo BCH(31,26) decodes each one of them independently. The estimated short-term BER before and after turbo BCH(31,26) decoding for all modulation modes was obtained according to Equation 3.15 and Equation 4.14, respectively. Thus, we have the estimated short-term BER of the received data burst before and after decoding for every modulation mode under the same channel conditions, which we could use to observe the BER degradation/improvement, when we switch from $\mathcal{M}$-QAM to a higher/lower value of $\mathcal{M}$. We could not use the BER performance versus SNR curve of Figure 4.11 generated over the dispersive two-path fading channel of Table 4.4 for the various QAM modes to estimate the BER improvement/degradation, since the BER in that figure was an average of the time-varying short-term BER of all the transmitted bursts over the faded channel. For the switching mechanism we need the 'short-term' BER measure and not the 'long-term' BER measure to configure the modem for the next transmission burst.

The switching BER thresholds for the 'before decoding'-scheme can be obtained by estimating the degradation/improvement of the short-term BER *before decoding*, when the modulation mode is switched from $\mathcal{M}$-QAM to a higher/lower value of $\mathcal{M}$ to achieve the target BER after decoding. Figure 4.20 shows the estimated short-term BER *after decoding* for all the possible modulation modes that can be switched to versus the estimated short-term BER of 16-QAM *before decoding* under the same channel conditions. The figure shows how each switching BER threshold $P_i^{16}, i = 2, 4, 16, 64$ is obtained. For example, in order to maintain the target BER of 10^{-4}, the short-term BER of the 16-QAM transmission burst before turbo decoding has to be approximately 2.5×10^{-1}, 2×10^{-1}, 5×10^{-2} and 1×10^{-3}, when switching to BPSK, 4-QAM and 64-QAM, respectively, under the same channel conditions. Using the same method for the other modulation modes, the switching BER thresholds are obtained, as listed in Table 4.6. For the 'after decoding' switching scheme, the short-term BER thresholds $P_i^{\mathcal{M}}, i = 2, 4, 16, 64$, listed in Table 4.7 were obtained. However, for NO TX bursts, where only dummy data are transmitted, turbo decoding is not necessary. Thus, for NO TX bursts we use the short-term BER *before decoding* as the switching metric.

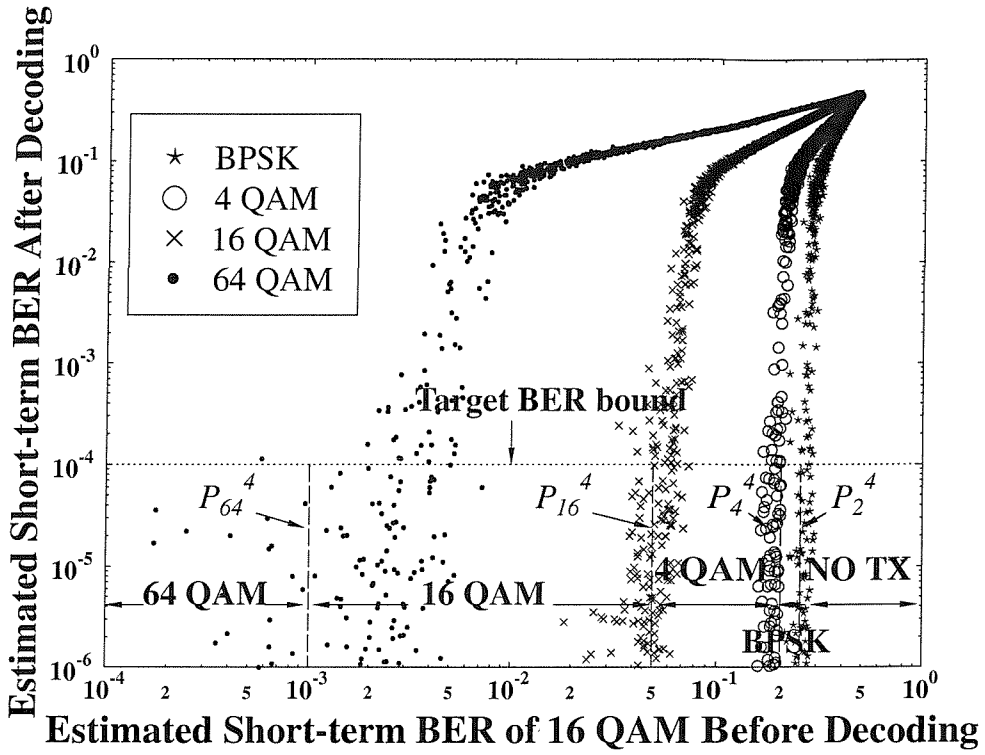


Figure 4.20: The estimated short-term BER after turbo BCH(31,26) decoding for all the possible modulation modes that can be invoked, assuming that current mode is 16-QAM – versus the estimated short-term BER of 16-QAM **before** decoding over the two-path Rayleigh fading channel of Table 3.1.

	$P_2^{\mathcal{M}}$	$P_4^{\mathcal{M}}$	$P_{16}^{\mathcal{M}}$	$P_{64}^{\mathcal{M}}$
NO TX	2.5×10^{-2}	2×10^{-3}	1×10^{-32}	0.0
BPSK	2.5×10^{-2}	2×10^{-3}	1×10^{-32}	0.0
4-QAM	1×10^{-1}	4×10^{-2}	4×10^{-5}	0.0
16-QAM	2.5×10^{-1}	2×10^{-1}	5×10^{-2}	1×10^{-3}
64-QAM	3.2×10^{-1}	2.5×10^{-1}	1.3×10^{-1}	5×10^{-2}

Table 4.6: The switching BER thresholds $P_i^{\mathcal{M}}$ of the joint adaptive modulation and RBF DFE scheme for the turbo-decoded target BER of 10^{-4} over the two-path Rayleigh fading channel of Table 3.1. The switching metric is based on the estimated short-term BER obtained before turbo decoding from the **RBF DFE**. This table explicitly indicates the uncoded modem BER that has to be maintained by the modem modes shown at the top of the table, in order to achieve the 10^{-4} turbo-decoded BER after switching to the various modem modes seen in the left-most column.

	$P_2^{\mathcal{M}}$	$P_4^{\mathcal{M}}$	$P_{16}^{\mathcal{M}}$	$P_{64}^{\mathcal{M}}$
NO TX	3×10^{-2}	2.5×10^{-3}	1×10^{-35}	0.0
BPSK	1×10^{-5}	1×10^{-30}	0.0	0.0
4-QAM	8×10^{-2}	1×10^{-5}	1×10^{-50}	0.0
16-QAM	2×10^{-1}	1.6×10^{-1}	1×10^{-5}	1×10^{-34}
64-QAM	3.2×10^{-1}	2.7×10^{-1}	1.3×10^{-1}	1×10^{-5}

Table 4.7: The switching BER thresholds $P_i^{\mathcal{M}}$ of the joint adaptive modulation and RBF DFE scheme for the turbo-decoded target BER of 10^{-4} over the two-path Rayleigh fading channel of Table 3.1. The switching metric is based on the estimated short-term BER obtained after turbo decoding from the **decoder**. This table explicitly indicates the coded modem BER that has to be maintained by the modem modes shown at the top of the table, in order to achieve the 10^{-4} turbo-decoded BER after switching to the various modem modes seen in the left-most column.

Figure 4.21 shows the performance of the 'before decoding'-scheme and 'after decoding'-scheme using the switching thresholds given in Tables 4.6 and 4.7, respectively. Both schemes have similar BPS performances. However, the 'before decoding'-scheme performs better, than the 'after decoding'-scheme in terms of its BER performance. Note that the 'after decoding'-scheme could only achieve the target BER of 10^{-4} beyond the SNR of 32dB. The performance degradation of the 'after decoding'-scheme can be explained by observing Figure 4.22, which shows the short-term BER fluctuation obtained before and after decoding at an SNR of 10dB for 4-QAM – the dominant modulation mode at 10dB. The BER fluctuation after decoding is more spurious and hence exhibits a higher variance than before decoding. Our modem mode switching mechanism assumes that the BER of the transmission burst is slowly varying and the estimated short-term BER of the *current* received burst is used to select the modulation mode for the *next* transmission burst. The spurious nature of the short-term BER after decoding, which is used as the switching metric, defies the BER predictability assumptions made and hence degrades the performance of the modulation mode switching mechanism. We also note from Table 4.7 that the thresholds required for the modem to switch to a higher-order modulation mode are extremely low. For example, when the BER must be lower than $P_{16}^4 = 1 \times 10^{-50}$ for the modem to switches from 4-QAM to 16-QAM. The extremely low values of the thresholds associated with the 'after decoding'-scheme degrade the performance of the mode switching mechanism. The 'before decoding'-scheme has a more reasonable set of thresholds, as shown in Table 4.6 and therefore performs better.

In the following section, we will investigate the performance of the coded adaptive scheme using the average burst LLR magnitude, defined by Equation 4.15 as an alternative switching metric.

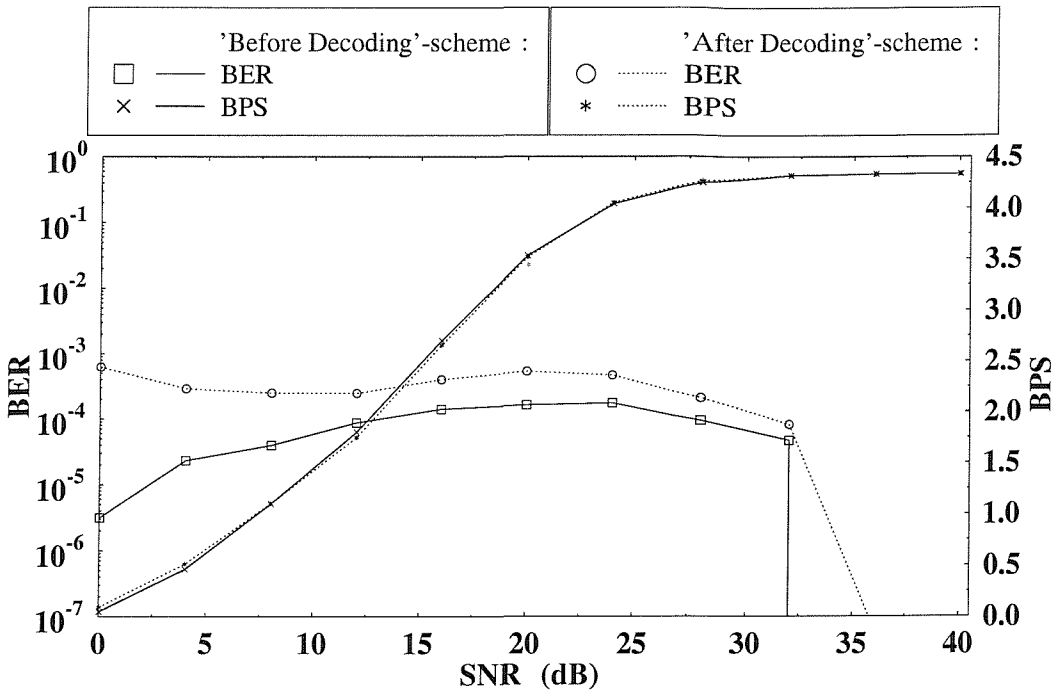


Figure 4.21: The BER and BPS performance of the turbo BCH(31,26) coded AQAM Jacobian RBF DFE aiming for a target BER of 10^{-4} for data-transmission using the parameters listed in Table 4.4. The 'before decoding'-scheme and 'after decoding'-scheme uses the estimated short-term BER of Equation 3.15 and 4.14 before and after decoding, respectively, as switching metric. The modem mode switching BERs used for both schemes are listed in Table 4.6 and 4.7. The Jacobian RBF DFE had a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol. The turbo coding parameters are given in Table 4.3 and the number of turbo decoder iterations is six. The BbB turbo interleaver size was fixed according to the modulation mode used as shown in Table 4.5.

4.6.3 Performance of the AQAM Jacobian RBF DFE Scheme: Switching Metric Based on the Average Burst LLR Magnitude

As discussed in Section 4.3, the probability of bit error is related to the magnitude of the bit LLR according to Equation 4.13. Thus, in addition to the BER-based switching criteria of the previous section, the magnitude of the bit LLR can also be used as the modem mode switching metric. The turbo decoder iteratively improves the BER of the decoded bits. Since the average burst LLR magnitude before and after decoding has an approximately linear relationship, as demonstrated by Figure 4.12 in Section 4.4.2, the average probability of error for the decoded burst can be inferred from the average burst LLR magnitude provided by the RBF equaliser using Equation 4.15. Thus, this parameter can also be used as the switching metric of the turbo-coded BbB AQAM scheme.

Figure 4.23 and 4.24 portray the average burst LLR magnitude fluctuation before and

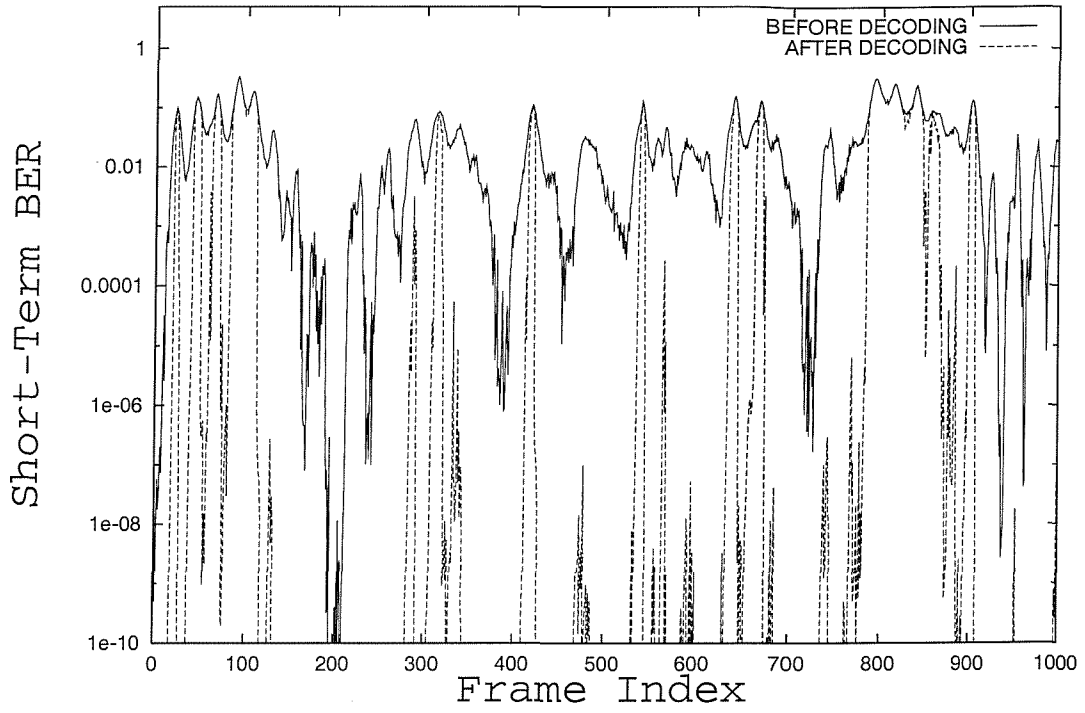
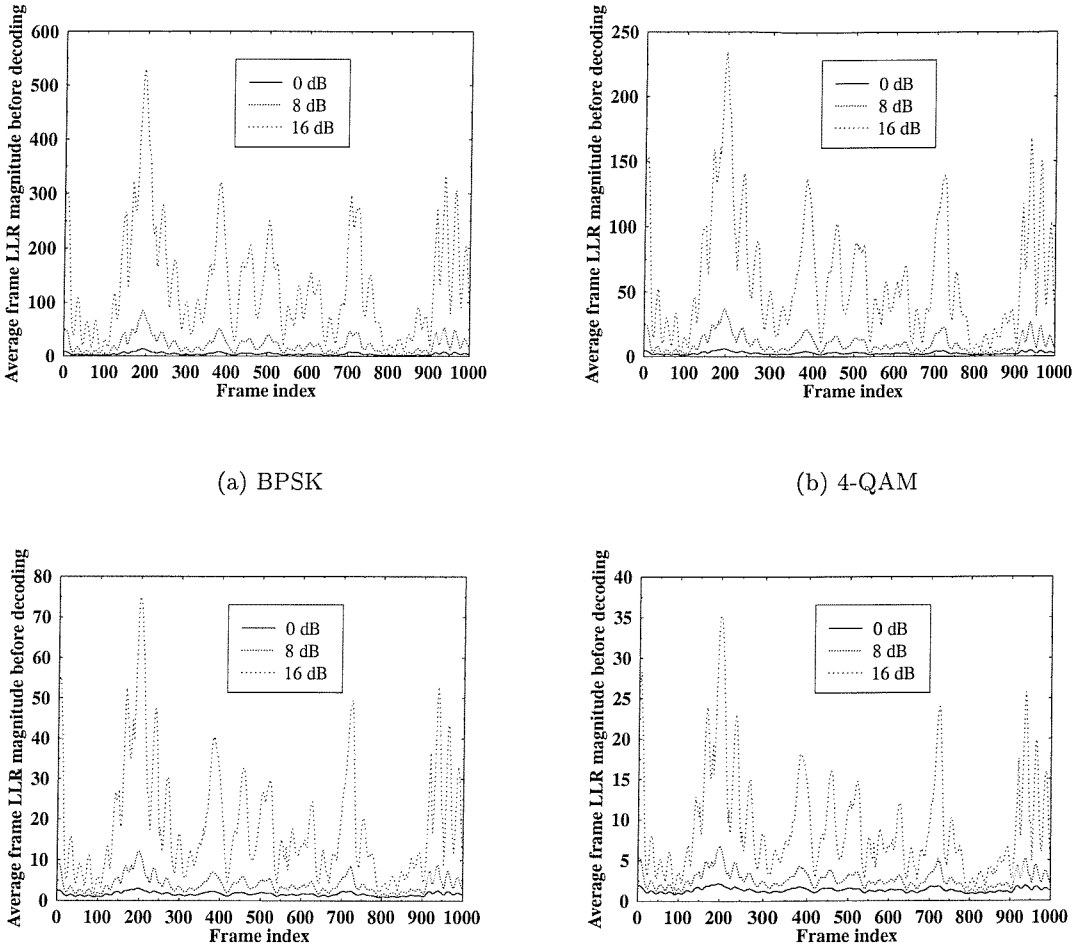


Figure 4.22: Short-term BER before and after turbo BCH(31,26) decoding versus symbol index for 4-QAM and for a channel SNR of 10dB over the two-path equal-weight, symbol-spaced Rayleigh fading channel of Table 3.1. The RBF DFE had a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol. Perfect CIR estimation is assumed and decision fedback error propagation is ignored. These low short-term BER estimates were obtained from the average values of Equation 4.13, which was plotted in Figure 4.14.

after turbo decoding, respectively, over the burst-invariant channel of Table 4.4 versus the symbol index for various QAM modes, as given by the RBF DFE and the turbo decoder, which is slowly varying and predictable for a number of consecutive data bursts. Therefore in our simulated channel scenario the average burst LLR magnitude both before and after turbo decoding constitute suitable metrics for the AQAM switching mechanism.

The average burst LLR magnitude obtained from either the RBF DFE or the turbo decoder is compared to a set of switching LLR magnitudes corresponding to the modulation mode of that data burst. Consequently, a modulation mode is selected for the next transmission burst, based on the current estimated BER upon assuming slowly fading channels. More explicitly, this implies that the similarity of the average burst LLR magnitude of consecutive data bursts can be exploited, in order to set the next modulation mode. Again, the modulation modes utilized in our system are BPSK, 4-QAM, 16-QAM, 64-QAM and no transmission (NO TX). Therefore, the modulation mode is switched according to the



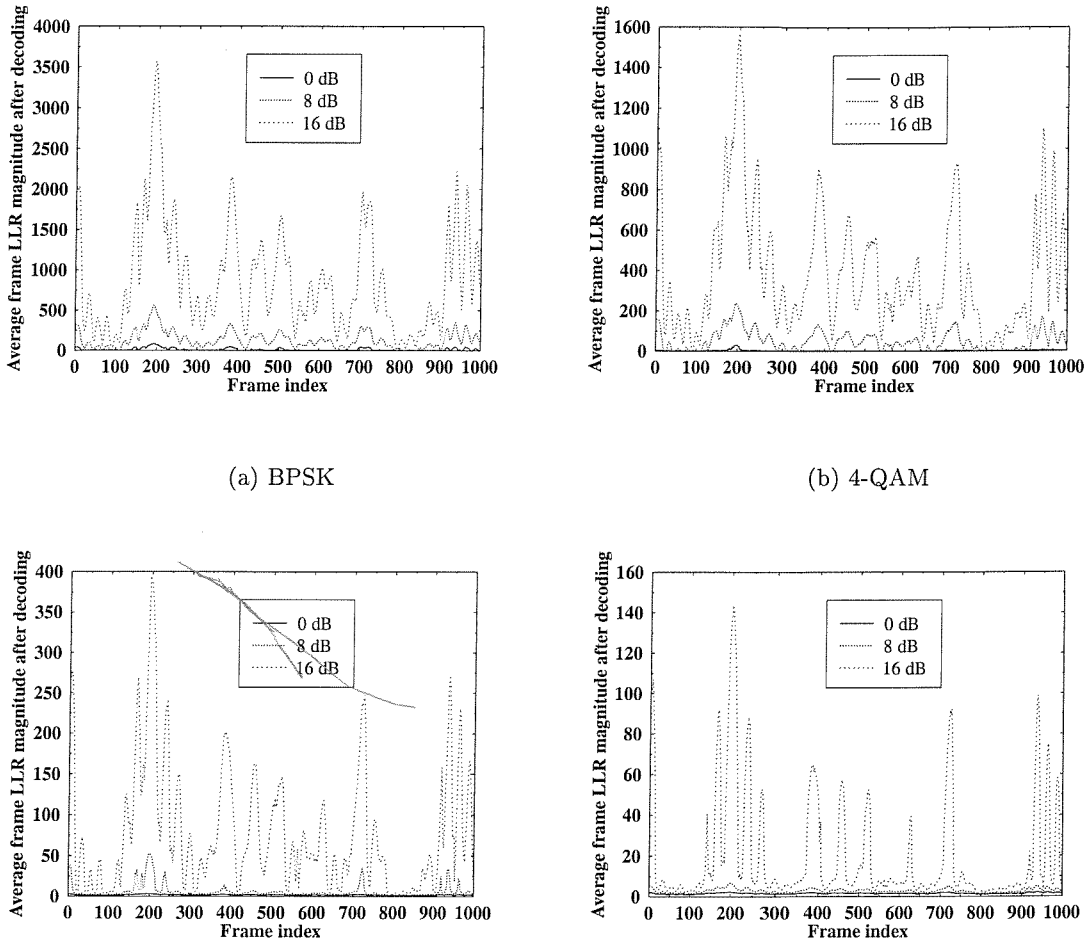
(a) BPSK

(b) 4-QAM

(c) 16-QAM

(d) 64-QAM

Figure 4.23: Average burst LLR magnitude **before turbo decoding** versus transmission burst index for various QAM modes as given by the RBF DFE over the two-path equal-weight, symbol-spaced Rayleigh fading channel of Table 3.1. Perfect CIR estimation is assumed and error propagation in decision feedback is ignored.



(a) BPSK

(b) 4-QAM

(c) 16-QAM

(d) 64-QAM

Figure 4.24: Average burst LLR magnitude **after turbo decoding** versus transmission burst index for various QAM modes as given by the RBF DFE over the two-path equal-weight, symbol-spaced Rayleigh fading channel of Table 3.1. Perfect CIR estimation is assumed and error propagation in decision feedback is ignored.

average burst LLR magnitude as follows:

$$\text{Modulation Mode} = \begin{cases} \text{NO TX} & \text{if } \mathcal{L}_{\text{average}} \leq \mathcal{L}_2^{\mathcal{M}} \\ \text{BPSK} & \text{if } \mathcal{L}_2^{\mathcal{M}} < \mathcal{L}_{\text{average}} \leq \mathcal{L}_4^{\mathcal{M}} \\ \text{4-QAM} & \text{if } \mathcal{L}_4^{\mathcal{M}} < \mathcal{L}_{\text{average}} \leq \mathcal{L}_{16}^{\mathcal{M}} \\ \text{16-QAM} & \text{if } \mathcal{L}_{16}^{\mathcal{M}} < \mathcal{L}_{\text{average}} \leq P_{64}^{\mathcal{M}} \\ \text{64-QAM} & \text{if } \mathcal{L}_{64}^{\mathcal{M}} < \mathcal{L}_{\text{average}}, \end{cases} \tag{4.16}$$

where $\mathcal{L}_i^{\mathcal{M}}, i = 2, 4, 16, 64$ are the switching LLR magnitude thresholds corresponding to the $\mathcal{M}$-QAM mode.

The LLR magnitude switching thresholds corresponding to $\mathcal{M}$-QAM, $\mathcal{L}_i^{\mathcal{M}}, i = 2, 4, 16, 64$, can be obtained by estimating the average burst LLR magnitude degradation/improvement, upon switching the modulation mode from $\mathcal{M}$-QAM to a higher/lower number of bits per symbol. The target BER requirement can be met by obtaining the average burst LLR magnitude of each modulation mode corresponding to the estimated channel quality and by activating the specific mode satisfying this target BER.

In our experiments, we obtained the LLR magnitude degradation/improvement upon switching from each modem mode to all other legitimate modes under the same instantaneous channel conditions. As an example for the 'before decoding'-scheme, Figure 4.25 shows the short-term BER – defined in Equation 4.14 – that would be encountered upon switching to all possible AQAM modes after BCH(31,26) turbo decoding versus the average burst LLR magnitude of 4-QAM before decoding, which was the current AQAM mode. In order to maintain the target BER of 10^{-4}, Figure 4.25 demonstrates how each switching LLR magnitude $\mathcal{L}_i^4, i = 2, 4, 16, 64$ is obtained after averaging the LLR magnitude occurances seen in the figure. More explicitly, the average burst LLR magnitudes before decoding encountered in the 4-QAM transmission burst would have to be 4.0, 7.5, 40.0 and 100.0, before switching to BPSK, 4-QAM, 16-QAM and 64-QAM AQAM bursts under the same channel conditions, leading to an estimated BER of 10^{-4} after BCH(31,26) turbo decoding. For example, if the average LLR magnitude $\mathcal{L}_{\text{average}}$ before decoding of the received 4-QAM transmission burst is in the range of $100 > \mathcal{L}_{\text{average}} \geq 40$, the modulation mode is switched from 4-QAM to 16-QAM for the next AQAM burst, since the BER of this 16-QAM transmission burst is estimated to be below the target BER of 10^{-4}. Note that due to the 'spreading' of the average burst LLR magnitude before decoding versus the short-term BER curve – especially for higher-order AQAM modes, as seen in Figure 4.25 – the threshold is estimated from the mean of this dynamic range. Using the same method for the other modulation modes, the 'before decoding' switching LLR magnitude thresholds were obtained for the turbo-decoded target BER of 10^{-4}, as listed in Table 4.8. For the 'after decoding' switching scheme, a similar method was implemented, in order to obtain

the switching thresholds listed in Table 4.8. Similar to the 'after decoding' short-term BER switching metric described in Section 4.6.2, the average burst LLR magnitude *before decoding* is used as the switching metric for the NO TX bursts, since turbo decoding is not performed in this mode.

	Before Decoding				After Decoding			
	$\mathcal{L}_2^{\mathcal{M}}$	$\mathcal{L}_4^{\mathcal{M}}$	$\mathcal{L}_{16}^{\mathcal{M}}$	$\mathcal{L}_{64}^{\mathcal{M}}$	$\mathcal{L}_2^{\mathcal{M}}$	$\mathcal{L}_4^{\mathcal{M}}$	$\mathcal{L}_{16}^{\mathcal{M}}$	$\mathcal{L}_{64}^{\mathcal{M}}$
NO TX	8.0	17.0	90.0	380.0	8.0	17.0	90.0	380.0
BPSK	8.0	17.0	90.0	380.0	40.0	100.0	∞	∞
4QAM	4.0	7.5	40.0	140.0	6.0	32.0	230.0	∞
16QAM	2.0	3.0	11.5	55.0	3.2	4.0	40.0	200.0
64QAM	1.7	2.2	6.2	30.0	24.0	3.0	6.5	45.0

Table 4.8: The switching LLR magnitude thresholds $\mathcal{L}_i^{\mathcal{M}}$ before and after decoding of the RBF DFE BbB AQAM scheme with turbo coding for the target BER of 10^{-4} over the two-path Rayleigh fading channel of Table 4.4.

Figure 4.26 compares the performance of the adaptive schemes using the short-term BER estimate based on Equation 4.14 and the average burst LLR magnitude before and after decoding as the switching metric. Both the 'before' and 'after decoding' LLR schemes of this section have similar BER and BPS performances to the 'before decoding' short-term BER scheme of Section 4.6.2, although the scheme using the average burst LLR magnitude as the switching metric has a lower computational complexity. This is because the output of the Jacobian RBF DFE is in a logarithmic form and obtaining the short-term BER values requires us to convert the logarithmic output to the non-logarithmic domain using exponential functions, in order to acquire the probability of bit error according to Equation 4.14.

Figure 4.27 shows our performance comparison of the AQAM Jacobian RBF DFE scheme in conjunction with turbo BCH(31,26) coding for the target BER of 10^{-4} with the 'before decoding' LLR magnitude switching metric along with its constituent turbo-coded fixed QAM modes. Figure 4.27 also shows the BER and BPS performance of the AQAM RBF DFE scheme *without turbo coding*, using the *short-term BER* as the switching metric, as described in Section 3.4 for performance comparison. The switching BER thresholds of the AQAM RBF DFE scheme *without turbo coding* were listed in Table 3.5.

Referring to Figure 4.27, the coded BPS performance was better than that of the uncoded scheme for the channel SNR range of 0dB to 26dB, with a maximum SNR gain of 4dB at a channel SNR of 0dB. However, at high SNRs, the BPS performance is limited by the coding rate of the system to achieve a maximum BPS throughput of $\frac{26}{36} \cdot 6 = 4.33$. The turbo BCH(31,26) coded AQAM system also exhibited a superior BER performance, when

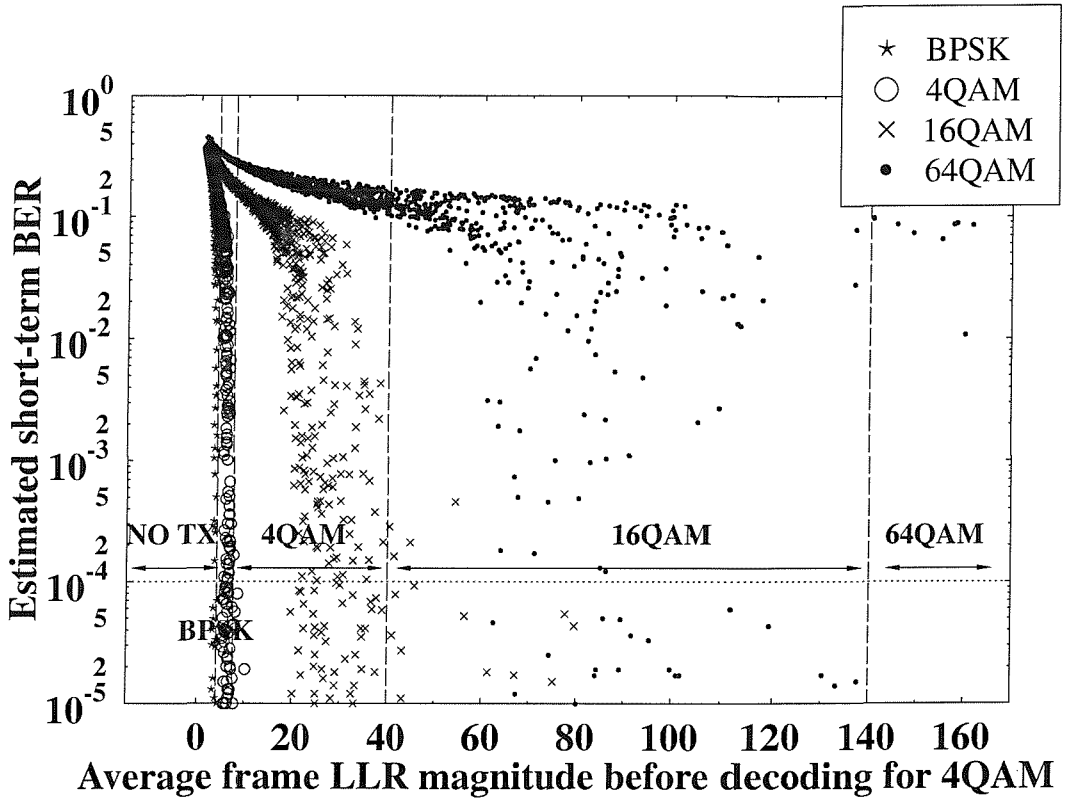


Figure 4.25: The estimated short-term BER for all the possible turbo BCH(31,26) decoded AQAM modes versus the average burst LLR magnitude of 4-QAM over the two-path Rayleigh fading channel of Table 3.1. The figure illustrates the expected spread of the short-term BER of all turbo decoded modem modes given a certain average burst LLR magnitude value in conjunction with 4-QAM as the current modem mode.

compared to the uncoded system for the channel SNR range of 0dB to 16dB and for the range above 28dB. However, the coded AQAM system failed to achieve the target BER of 10^{-4} for the SNR range of 16dB to 28dB. This was because the spread nature of the short-term BER versus LLR magnitude curves observed in Figure 4.25 leads to inaccuracies in obtaining these LLR magnitude thresholds, especially for $\mathcal{L}_{16}^{\mathcal{M}}$ and $\mathcal{L}_{64}^{\mathcal{M}}$, as demonstrated in Figure 4.25. These inaccuracies affect the switching performance for the SNR range of 16dB to 28dB. The spread nature of the short-term BER versus LLR magnitude curves in Figure 4.25 is due to a number of factors and these investigations are set aside for future work.

	$\mathcal{L}_2^{\mathcal{M}}$	$\mathcal{L}_4^{\mathcal{M}}$	$\mathcal{L}_{16}^{\mathcal{M}}$	$\mathcal{L}_{64}^{\mathcal{M}}$
NO TX	10.0	30.0	280.0	1000.0
BPSK	10.0	30.0	280.0	1000.0
4-QAM	8.0	12.0	100.0	350.0
16-QAM	3.0	5.0	30.0	120.0
64-QAM	2.5	3.0	13.0	70.0

Table 4.9: The switching LLR magnitude thresholds $\mathcal{L}_i^{\mathcal{M}}$ before decoding of the RBF DFE BbB AQAM scheme using turbo coding for the zero-error target performance over the two-path Rayleigh fading channel of Table 4.4.

Since the estimated short-term BER is a somewhat eratic function of the turbo decoder's input LLR, the switching LLR values have to be conservative, if the target BER cannot be exceeded. For the BER = 10^{-4} scenario the switching LLR was adjusted experimentally to be near the upper end of the LLR-range observed in Figure 4.25. When aiming for virtually error-free communications, an even more conservative LLR threshold has to be chosen, in order not to precipitate a plethora of transmission errors, even at the cost of thereby reducing the achievable, BPS throughput of the system. Figure 4.28 shows the BER and BPS performance of the near-error-free, turbo-coded AQAM Jacobian RBF DFE scheme with the more conservative, increased LLR magnitude switching thresholds listed in Table 4.9. The BER and BPS performance of the uncoded AQAM RBF DFE system is also given in the figure for comparison. The BPS performance of the error-free coded system was better, than that of the uncoded AQAM system for the channel range of 0dB to 15dB, as evidenced by Figure 4.28. However, the BPS performance is limited by the coding rate of the system to a maximum value of 4.33 at high channel SNRs. This suggests that the best overall BER/BPS performance is achieved by our system, if we add the AQAM option of switching off the turbo BCH(31,26) code under high SNR conditions, namely around 25dB. This allows us to attain a BPS of 6 in this SNR region.

Wong [83] introduced the concept of variable rate turbo coding AQAM schemes with

the aim of improving the throughput of turbo block coded AQAM scheme at high channel SNRs. Two types of variable code rate schemes were implemented:

1. **Partial turbo block coded adaptive modulation scheme**: The switching mechanism is capable of disabling and enabling the channel encoder for a chosen modulation mode.

2. **Variable rate turbo block coded adaptive modulation scheme**: The coding rate is varied by utilizing different BCH component codes for the different modulation modes. The higher-order modulation modes are assigned a higher code rate, in order to improve the effective data throughput at medium to high average channel SNRs and conversely, the lower-order modulation modes will be accompanied by lower code rates, in order to ensure maximum error protection at low average channel SNRs, where these modes have a high selection probability.

These methods can similarly be implemented for our turbo-coded AQAM RBF DFE system, in order to improve the throughput performance at high channel SNRs.

4.6.4 Switching Metric Selection

The choice of the switching metric depends on a variety of factors, which are discussed here with reference to Figures 4.21 – 4.27. The most reliable channel quality metric is the BER of a given transmitted burst, since this metric is capable of quantifying all channel impairments, irrespective of the effects of its source. Explicitly, the BER of the transmission burst quantifies the influence of reduced received signal strength or reduced SNR, that of increased ISI or co-channel interference, etc. The short-term BER of a transmission burst can be estimated for example with the aid of the RBF DFE using Equation 4.14.

In conjunction with turbo FEC coding also, the LLR of Equation 4.8 at the input or output of the turbo decoder can be used with the aid of Equation 4.13 and 4.14, in order to estimate the BER. Explicitly, the probability of a specific bit being in error is given by Equation 4.13, which can be averaged according to Equation 4.14 for a transmission burst. The corresponding short-term BER versus transmission burst index was plotted using both the channel decoder's input and output SNRs in Figure 4.22 over the two-path equal-weight symbol-spaced Rayleigh channel of Table 4.4. Observe that due to the higher fluctuation of the FEC decoder's output LLRs the output BER fluctuates over a wider range. The corresponding turbo decoder LLRs both before and after turbo decoding are plotted in Figure 4.23 and 4.24, respectively, for 0dB, 8dB and 16dB channel SNRs. As expected, the evolution of the LLRs is similar, although the output LLR fluctuates over a wider dynamic range, since the turbo decoder typically improves the input LLRs upon each iteration, unless the LLR changes polarity several times, which is the sign of a low-reliability decision.

Since the BER curves of the turbo-coded constituent AQAM modem modes seen in Figure 4.9 are extremely steep, upon switching for example from the BPSK mode to 4-QAM the BER increases dramatically, by several orders of magnitude. Hence, for example the BPSK BER has to become significantly lower than 10^{-5} in Figure 4.9, before switching to 4-QAM can take place. This justifies the extreme BER differences observed in Table 4.6 and Table 4.7. In conclusion of our discussions on the choice of switching metric we infer from Figure 4.21 that whilst the BER of the AQAM switching regime using the LLRs before turbo-decoding attains a lower BER, this is not associated with any reduction of the BPS throughput, and hence this switching metric was deemed more beneficial to invoke. This is because due to the higher steepness of the turbo-decoded BER curve, the BER is more often misjudged on the basis of the output LLRs. This then often results in using an 'optimistic' high BPS AQAM mode, which increases the BER. When the channel quality is under-estimated, a reduced number of bits per symbol is used, however the associated BER reduction is insufficient for compensating for the increase of BER of the 'over-estimated' channel quality scenario. These under- and over-estimated BERs result in the high spread of the curves seen in Figure 4.25.

4.7 Conclusions

In this chapter, we have investigated the performance of the RBF equaliser using turbo coding. We have also demonstrated the application of turbo BCH coding in conjunction with AQAM in a wideband fading channel. The use of different switching criteria – namely the short-term BER and average burst LLR magnitude before and after decoding – was discussed. We observed that the performance of the switching mechanism depends on the fluctuation of the switching metric, since the AQAM scheme assumes that the channel quality is slowly varying. The turbo-coded AQAM RBF DFE system exhibited a better BPS performance, when compared to the uncoded system at low to medium channel SNRs, as evidenced by Figure 4.27. The same figure also showed an improved coded BER performance at higher channel SNRs. A virtually error-free turbo-coded AQAM scheme was also characterized in Figure 4.28.

In the next chapter we will explore the recently developed family of iterative equalisation and channel decoding techniques, a scheme which is termed as turbo equalisation. We will investigate the employment of RBF equaliser in the equaliser component.

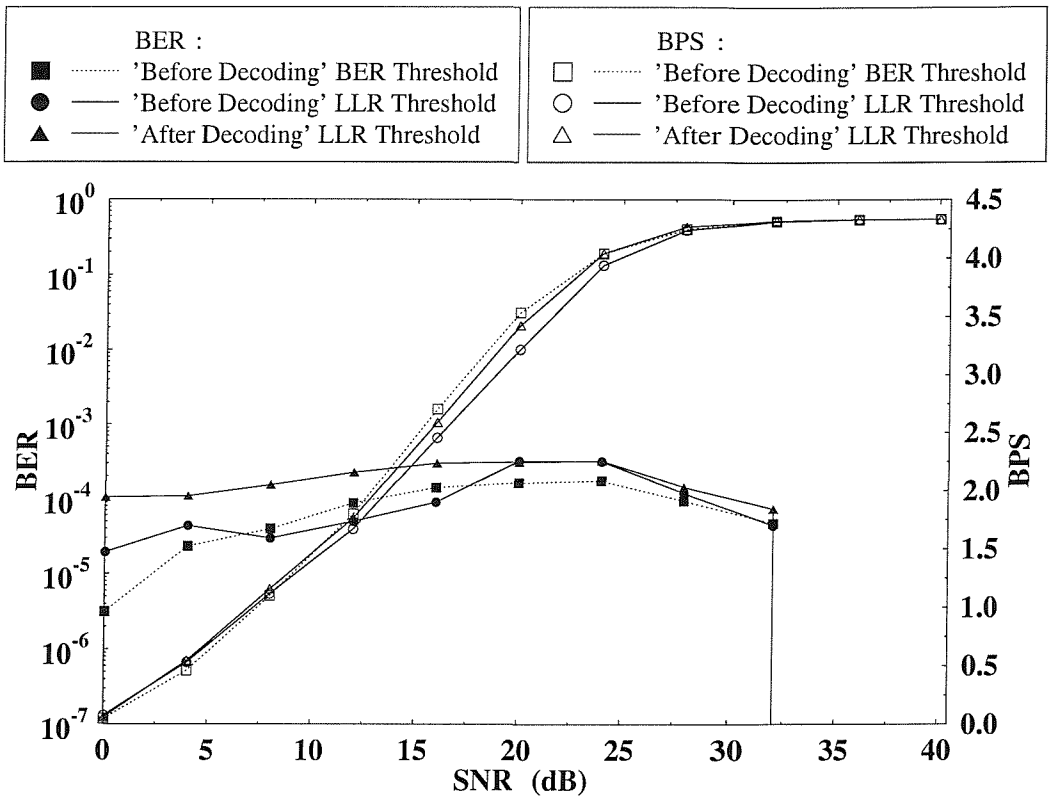


Figure 4.26: The BER and BPS performance of the turbo BCH(31,26) coded AQAM Jacobian RBF DFE using different switching metrics for a data-transmission target BER of 10^{-4} over the two-path Rayleigh fading channel of Table 4.4. The modem mode switching thresholds used for both scheme are listed in Tables 4.6 and 4.8, respectively. The Jacobian RBF DFE had a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol. The turbo coding parameters were given in Table 4.3 and the number of turbo decoder iterations was six. The BbB turbo interleaver size was fixed according to the modulation mode used, as shown in Table 4.5.

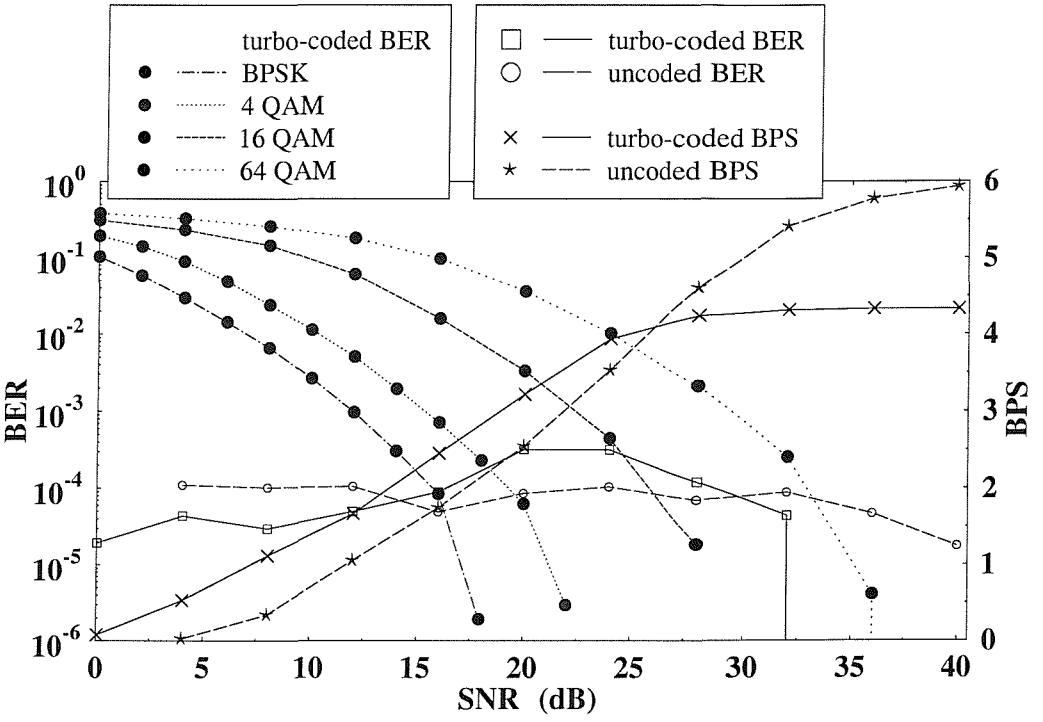


Figure 4.27: The BER and BPS performance of the uncoded and turbo BCH(31,26) coded AQAM Jacobian RBF DFE for a data-transmission target BER of 10^{-4} over the two-path Rayleigh fading channel of Table 4.4. The average LLR magnitude modem mode switching thresholds before decoding used for this scheme are listed in Table 4.8. The Jacobian RBF DFE had a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol. The turbo coding parameters were given in Table 4.3 and the number of turbo decoder iterations was six. The BbB turbo interleaver size was fixed according to the modulation modes used, as shown in Table 4.5.

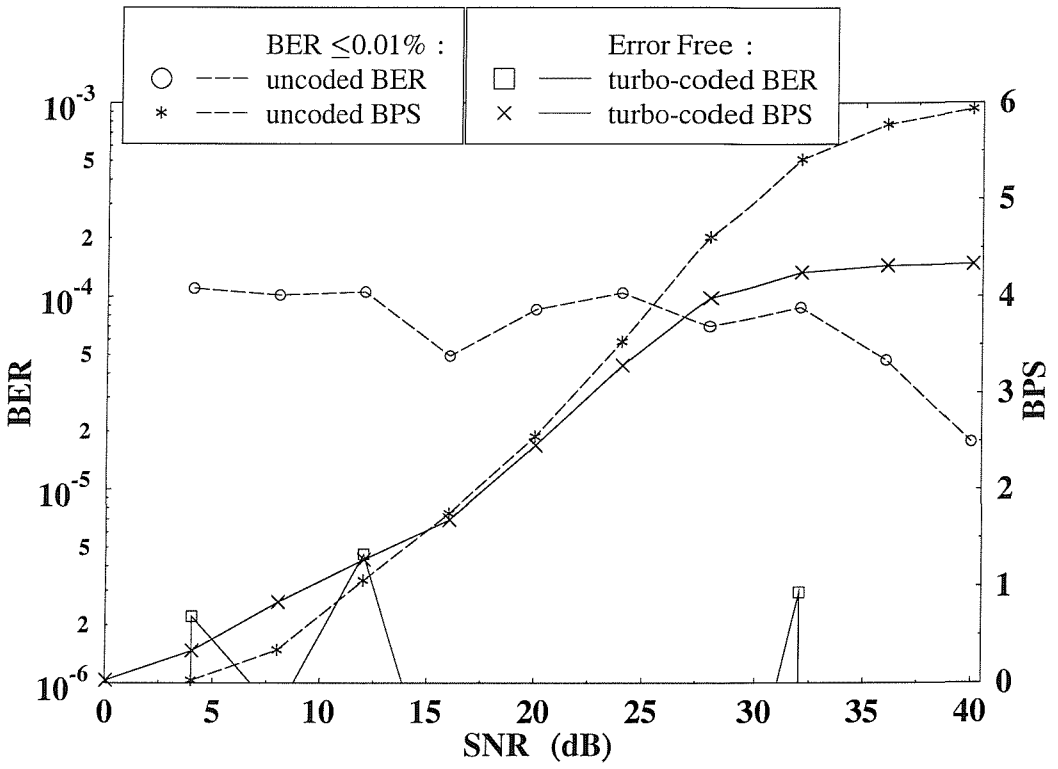


Figure 4.28: The BER and BPS performance of the turbo BCH(31,26) coded AQAM Jacobian RBF DFE for targetted no error transmission over the two-path Rayleigh fading channel of Table 4.4. The average LLR magnitude modem mode switching thresholds before decoding used for this scheme are listed in Table 4.9. The Jacobian RBF DFE had a feedforward order of $m = 2$, feedback order of $n = 1$ and decision delay of $\tau = 1$ symbol. The turbo coding parameters were given in Table 4.3 and the number of turbo decoder iterations is six. The BbB turbo interleaver size was fixed according to the modulation mode used as shown in Table 4.5.

Chapter 5

RBF Turbo Equalisation

This chapter presents a novel turbo equalisation scheme, which employs a RBF equaliser instead of the conventional trellis-based equaliser of Douillard *et. al.* [10]. The basic principles of turbo equalisation will be highlighted. Structural, computational cost and performance comparisons of the RBF-based and trellis-based turbo equalisers are provided. A novel element of our design is that in order to reduce the computational complexity of the RBF turbo equaliser (TEQ), we propose invoking further iterations only, if the decoded symbol has a high error probability. Otherwise we curtail the iterations, since a reliable decision can be taken. Let us now introduce the concept of turbo equalisation.

5.1 Introduction to Turbo Equalisation

In the conventional RBF DFE based systems discussed in Chapter 4 equalisation and channel decoding ensued independently. However, it is possible to improve the receiver's performance, if the equaliser is fed by the channel outputs plus the soft decisions provided by the channel decoder, invoking a number of iterative processing steps. This novel receiver scheme was first proposed by Douillard *et. al.* [10] for a convolutional coded binary phase shift keying (BPSK) system, using a similar principle to that of turbo codes and hence it was termed *turbo equalisation*. This scheme is illustrated in Figure 5.1, which will be detailed during our forthcoming discourse. Gertsman and Lodge [102] extended this work and showed that the iterative process of turbo equalisation can compensate for the performance degradation due to imperfect channel estimation. Turbo equalisation was implemented in conjunction with turbo coding, rather than conventional convolutional coding by Raphaeli and Zarai [103], demonstrating an increased performance gain due to turbo coding as well as with advent of enhanced ISI mitigation achieved by turbo equalisation.

The principles of iterative turbo decoding [69] were modified appropriately for the coded $\mathcal{M}-QAM$ system of Figure 5.2. The channel encoder is fed with independent binary data

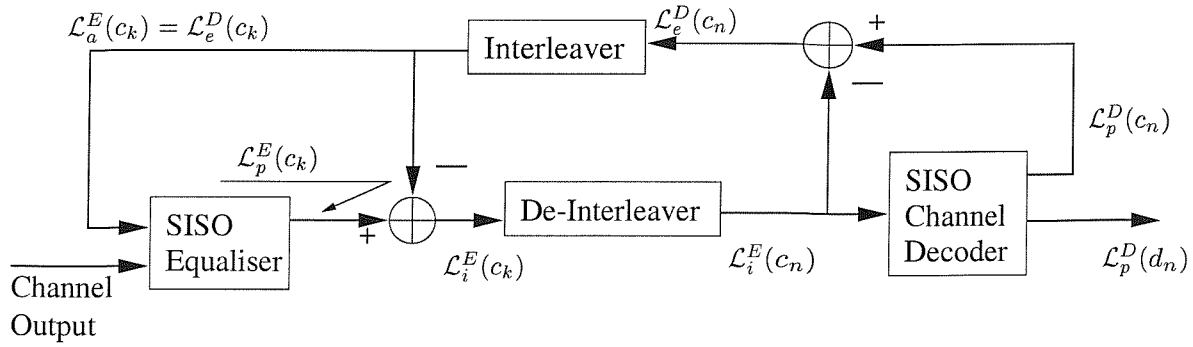


Figure 5.1: Iterative turbo equalisation schematic

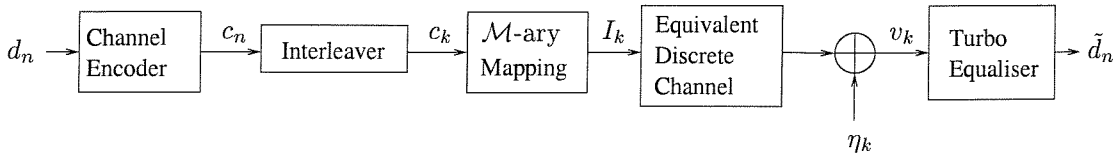


Figure 5.2: Serially concatenated coded $\mathcal{M}$-ary system using the turbo equaliser, which performs the equalisation, demodulation and channel decoding iteratively.

d_n and every $\log_2(\mathcal{M})$ number of bits of the interleaved, channel encoded data c_k is mapped to an $\mathcal{M}$-ary symbol before transmission. In this scheme the channel is viewed as an 'inner encoder' of a serially concatenated arrangement, since it can be modelled with the aid of a tapped delay line similar to that of a convolutional encoder [10, 104]. At the receiver the equaliser and decoder employ a Soft-In/Soft-Out (SISO) algorithm, such as the optimal Maximum *A Posteriori*(MAP) algorithm [96] or the Log-MAP algorithm [13]. The SISO equaliser processes the *a priori* information associated with the coded bits c_k transmitted over the channel and – in conjunction with the channel output values v_k – computes the *a posteriori* information concerning the coded bits. The soft values of the channel coded bits c_k are typically quantified in the form of the log-likelihood ratio defined in Equation 4.8. Note that in the context of turbo decoding – which was discussed in Chapter 4 – the SISO decoders compute the *a posteriori* information of the *source* bits only, while in turbo equalisation the *a posteriori* information concerning all the *coded* bits is required.

In our description of the turbo equaliser depicted in Figure 5.1, we have used the notation $\mathcal{L}^E$ and $\mathcal{L}^D$ to indicate the LLR values output by the SISO equaliser and SISO decoder, respectively. The subscripts e, i, a and p were used to represent the extrinsic LLR, the combined channel and extrinsic LLR, the *a priori* LLR and the *a posteriori* LLR, respectively. Referring to Figure 5.1, the SISO equaliser processes the channel outputs and the *a priori* information $\mathcal{L}_a^E(c_k)$ of the coded bits, and generates the *a posteriori* LLR values $\mathcal{L}_p^E(c_k)$ of the interleaved coded bits c_k seen in Figure 5.2. Before passing the above *a posteriori* LLRs generated by the SISO equaliser to the SISO decoder of Figure 5.1, the

contribution of the decoder — in the form of the *a priori* information $\mathcal{L}_a^E(c_k)$ — from the previous iteration must be removed, in order to yield the combined channel and extrinsic information $\mathcal{L}_i^E(c_k)$ seen in Figure 5.1. They are referred to as 'combined', since they are intrinsically bound and cannot be separated. However, note that at the initial iteration stage, no *a priori* information is available yet, hence we have $\mathcal{L}_a^E(c_k) = 0$. To elaborate further, the *a priori* information $\mathcal{L}_a^E(c_k)$ was removed at this stage, in order to prevent the decoder from processing its own output information, which would result in overwhelming the decoder's current reliability-estimation characterising the coded bits, *i.e.* the extrinsic information. The combined channel and extrinsic LLR values are channel-deinterleaved – as seen in Figure 5.1 – in order to yield $\mathcal{L}_i^E(c_n)$, which is then passed to the SISO channel decoder. Subsequently, the channel decoder computes the *a posteriori* LLR values of the coded bits $\mathcal{L}_p^D(c_n)$. The *a posteriori* LLRs at the output of the channel decoder are constituted by the extrinsic LLR $\mathcal{L}_e^D(c_n)$ and the channel-deinterleaved combined channel and extrinsic LLR $\mathcal{L}_i^E(c_n)$ extracted from the equaliser's *a posteriori* LLR $\mathcal{L}_p^E(c_k)$. The extrinsic part can be interpreted as the incremental information concerning the current bit obtained through the decoding process from all the information available due to all other bits imposed by the code constraints, but excluding the information directly conveyed by the bit. This information can be calculated by subtracting bitwise the LLR values $\mathcal{L}_i^E(c_n)$ at the input of the decoder from the *a posteriori* LLR values $\mathcal{L}_p^D(c_n)$ at the channel decoder's output, as seen also in Figure 5.1, yielding:

$$\mathcal{L}_e^D(c_n) = \mathcal{L}_p^D(c_n) - \mathcal{L}_i^E(c_n). \tag{5.1}$$

The extrinsic information $\mathcal{L}_e^D(c_n)$ of the coded bits is then interleaved in Figure 5.1, in order to yield $\mathcal{L}_e^D(c_k)$, which is fed back in the required bit-order to the equaliser, where it is used as the *a priori* information $\mathcal{L}_a^E(c_k)$ in the next equalisation iteration. This constitutes the first iteration. Again, it is important that only the channel-interleaved extrinsic part – *i.e.* $\mathcal{L}_e^D(c_k)$ of $\mathcal{L}_p^D(c_n)$ – is fed back to the equaliser, since the interdependence between the *a priori* information $\mathcal{L}_a^E(c_k) = \mathcal{L}_e^D(c_k)$ used by the equaliser and the previous decisions of the equaliser should be minimized. This independence assists in obtaining the equaliser's reliability-estimation of the coded bits for the current iteration, without being 'influenced' by its previous estimations. Ideally, the *a priori* information should be based on an independent estimation. As argued above, this is the reason that the *a priori* information $\mathcal{L}_a^E(c_k)$ is subtracted from the *a posteriori* LLR value $\mathcal{L}_p^E(c_k)$ at the output of the equaliser in Figure 5.1, before passing the LLR values to the channel decoder. In the final iteration, the *a posteriori* LLRs $\mathcal{L}_p^D(d_n)$ of the source bits are computed by the channel decoder. Subsequently, the transmitted bits are estimated by comparing $\mathcal{L}_p^D(d_n)$ to the threshold value of 0. For $\mathcal{L}_p^D(d_n) < 0$ the transmitted bit d_n is deemed to be a logical 0, while $d_n = +1$ or

a logical 1 is output, when $\mathcal{L}_p^D(d_n) \geq 0$.

Previous turbo equalisation research has implemented the SISO equaliser using the Soft-Output Viterbi Algorithm (SOVA) [10], the optimal MAP algorithm [105] and linear filters [106]. We will now introduce the proposed RBF based equaliser as the SISO equaliser in the context of turbo equalisation. The following sections will discuss the implementational details and the performance of this scheme, benchmarked against the optimal MAP turbo equaliser scheme of [105].

5.2 RBF Assisted Turbo Equalisation

The RBF network based equaliser is capable of utilizing the *a priori* information $\mathcal{L}_a^E(c_k)$ provided by the channel decoder of Figure 5.1, in order to improve its performance. This *a priori* information can be assigned namely to the weights of the RBF network [107]. We will describe this in more detail in this section. For convenience, we will rewrite Equation 2.81, describing the conditional probability density function (PDF) of the ith symbol, $i = 1, \ldots, \mathcal{M}$, associated with the ith subnet of the $\mathcal{M}$-ary RBF equaliser:

$$\begin{aligned} f_{RBF}^i(\mathbf{v}_k) &= \sum_{j=1}^{n_s^i} w_j^i \varphi(\|\mathbf{v}_k - \mathbf{c}_j^i\|), \\ \varphi(x) &= \exp\left(\frac{-x^2}{\rho}\right) \\ & \qquad i = 1, \ldots, \mathcal{M}, \qquad j = 1, \ldots, n_s^i \end{aligned} \tag{5.2}$$

where $\mathbf{c}_j^i$, w_j^i, $\varphi(\cdot)$ and ρ are the RBF's centres, weights, activation function and width, respectively. In order to arrive at the Bayesian equalisation solution [8] – which was highlighted in Section 2.9 – the RBF centres are assigned the values of the channel states $\mathbf{r}_j^i$ defined in Equation 2.84, the RBF weights defined in Section 2.7.1 correspond to the *a priori* probability of the channel states $p_j^i = P(\mathbf{r}_j^i)$ and the RBF width introduced in Section 2.7.1 is given the value of $2\sigma_\eta^2$ where σ_η^2 is the channel noise variance. The actual number of channel states n_s^i is determined by the specific design of the algorithm invoked, reducing the number of channel states from the optimum number of $\mathcal{M}^{m+L-1}$, where m is the equaliser feedforward order and $L+1$ is the CIR duration [26, 67, 68]. The probability p_j^i of the channel states $\mathbf{r}_j^i$, and therefore the weights of the RBF equaliser can be derived from the LLR values of the transmitted bits, as estimated by the channel decoder.

Expounding further from Equation 2.7 and 2.11, the channel output can be defined as

$$\mathbf{r}_j = \mathbf{F}\mathbf{s}_j, \tag{5.3}$$

where $\mathbf{F}$ is the CIR matrix defined in Equation 2.12 and $\mathbf{s}_j$ is the jth possible combination of the $(L+m)$ transmitted symbol sequence, $\mathbf{s}_j = \begin{bmatrix} s_{j1} & \ldots & s_{jp} & \ldots & s_{j(L+m)} \end{bmatrix}^T$. Hence

– for a time-invariant CIR and assuming that the symbols in the sequence $\mathbf{s}_j$ are statistically independent of each other – the probability of the received channel output vector $\mathbf{r}_j$ is given by:

$$\begin{aligned} P(\mathbf{r}_j) &= P(\mathbf{s}_j) \\ &= P(s_{j1} \cap \ldots s_{jp} \cap \ldots s_{j(L+m)}) \\ &= \prod_{p=1}^{L+m} P(s_{jp}) \qquad j = 1, \ldots, n_s^i. \end{aligned} \tag{5.4}$$

The transmitted symbol vector component s_{jp} – i.e. the pth symbol in the vector – is given by $\mathrm{m} = \log_2 \mathcal{M}$ number of bits $c_{jp1}, c_{jp2}, \ldots, c_{jp\mathrm{m}}$. Therefore,

$$\begin{aligned} P(s_{jp}) &= P(c_{jp1} \cap \ldots c_{jpq} \cap \ldots c_{jp\mathrm{m}}) \\ &= \prod_{q=1}^{\mathrm{m}} P(c_{jpq}) \qquad j = 1, \ldots, n_s^i, \qquad p = 1, \ldots, L+m. \end{aligned} \tag{5.5}$$

We have to map the bits c_{jpq} representing the $\mathcal{M}$-ary symbol s_{jp} to the corresponding bit $\{c_k\}$. Note that the probability $P(\mathbf{r}_j)$ of the channel output states and therefore also the RBF weights defined in Equation 5.2 are time-variant, since the values of $\mathcal{L}_p(c_k)$ are time-variant. Based on the definition of the bit LLR of Equation 4.8, the probability of bit c_k having the value of +1 or -1 can be obtained after a few steps from the *a priori* information $\mathcal{L}_a^E(c_k)$ provided by the channel decoder of Figure 5.1, according to:

$$P(c_k = \pm 1) = \frac{\exp(-\mathcal{L}_a^E(c_k)/2)}{1 + \exp(-\mathcal{L}_a^E(c_k))} \cdot \exp(\pm \mathcal{L}_a^E(c_k)/2). \tag{5.6}$$

Hence, referring to Equation 5.4, 5.5 and 5.6, the probability $P(\mathbf{r}_j)$ of the received channel output vector can be represented in terms of the bit LLRs $\mathcal{L}_a^E(c_{jpq})$ as follows:

$$\begin{aligned} P(\mathbf{r}_j) &= P(\mathbf{s}_j) \\ &= \prod_{p=1}^{L+m} P(s_{jp}) \\ &= \prod_{p=1}^{L+m} \prod_{q=1}^{\mathrm{m}} P(c_{jpq}) \\ &= \prod_{p=1}^{L+m} \prod_{q=1}^{\mathrm{m}} \frac{\exp(-\mathcal{L}_a^E(c_{jpq})/2)}{1 + \exp\left(-\mathcal{L}_a^E(c_{jpq})\right)} \cdot \exp\left(\frac{1}{2} \cdot c_{jpq} \cdot \mathcal{L}_a^E(c_{jpq})\right) \\ &= C_{\mathcal{L}_a^E(\mathbf{s}_j)} \cdot \prod_{p=1}^{L+m} \prod_{q=1}^{\mathrm{m}} \exp\left(\frac{1}{2} \cdot c_{jpq} \cdot \mathcal{L}_a^E(c_{jpq})\right) \\ &= C_{\mathcal{L}_a^E(\mathbf{s}_j)} \cdot \exp\left(\frac{1}{2} \sum_{p=1}^{L+m} \sum_{q=1}^{\mathrm{m}} c_{jpq} \cdot \mathcal{L}_a^E(c_{jpq})\right) \qquad j = 1, \ldots, n_s^i, \end{aligned} \tag{5.7}$$

where the constant $C_{\mathcal{L}_a^E(\mathbf{s}_j)} = \prod_{p=1}^{L+m} \prod_{q=1}^{\mathrm{m}} \frac{\exp(-\mathcal{L}_a^E(c_{jpq})/2)}{1+\exp(-\mathcal{L}_a^E(c_{jpq}))}$ is independent of the bit c_{jpq}.

Therefore, we have demonstrated how the soft output $\mathcal{L}_a^E(c_k)$ of the channel decoder of Figure 5.1 can be utilized by the RBF equaliser. Another way of viewing this process is that the RBF equaliser is trained by the information generated by the channel decoder. The RBF equaliser provides the *a posteriori* LLR values of the bits c_k according to

$$\mathcal{L}_p^E(c_k) = \ln\left(\frac{\sum_{\substack{i \\ c_k=+1}} f_{RBF}^i(\mathbf{v}_k)}{\sum_{\substack{i \\ c_k=-1}} f_{RBF}^i(\mathbf{v}_k)}\right), \tag{5.8}$$

where $f_{RBF}^i(\mathbf{v}_k)$ was defined by Equation 5.2 and the received sequence $\mathbf{v}_k$ is shown in Figure 5.2. In the next section we will provide a comparative study of the RBF equaliser and the conventional MAP equaliser of [108].

5.3 Comparison of the RBF and MAP Equaliser

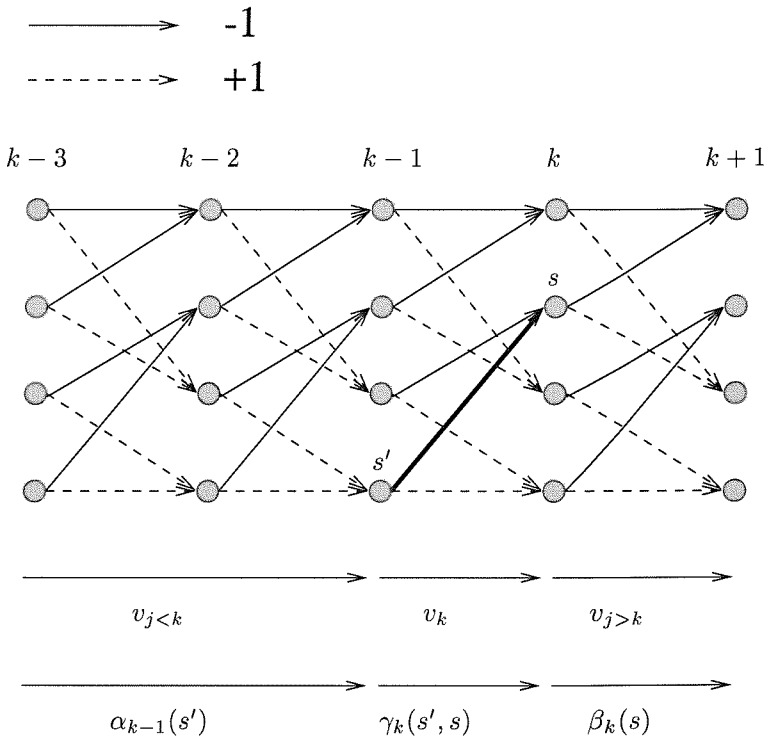


Figure 5.3: Example of a binary ($\mathcal{M} = 2$) system's trellis structure

The *a posteriori* LLR value $\mathcal{L}_p^E$ of the coded bit c_k, given the received sequence $\mathbf{v}_k$ of Figure 5.2, can be calculated according to [105]:

$$\mathcal{L}_p^E(c_k) = \ln\left(\frac{\sum_{(s',s)\Rightarrow c_k=+1} p(s', s, \mathbf{v}_k)}{\sum_{(s',s)\Rightarrow c_k=-1} p(s', s, \mathbf{v}_k)}\right), \tag{5.9}$$

where s' and s denote the states of the trellis seen in Figure 5.3 at trellis stages $k-1$ and k, respectively. The joint probability $p(s', s, \mathbf{v}_k)$ is the product of three factors [105]:

$$p(s', s, \mathbf{v}_k) = \underbrace{p(s', v_{j<k})}_{\alpha_{k-1}(s')} \cdot \underbrace{P(s|s') \cdot p(v_k|s', s)}_{\gamma_k(s',s)} \cdot \underbrace{p(v_{j>k}|s)}_{\beta_k(s)}, \tag{5.10}$$

where the term $\alpha_{k-1}(s')$ and $\beta_k(s)$ are the so-called forward- and backward oriented transition probabilities, respectively, which can be obtained recursively, as follows [105]:

$$\alpha_k(s) = \sum_{s'} \gamma_k(s', s) \cdot \alpha_{k-1}(s') \tag{5.11}$$

$$\beta_{k-1}(s) = \sum_{s} \gamma_k(s', s) \cdot \beta_k(s). \tag{5.12}$$

Furthermore, $\gamma_k(s', s), k = 1, \ldots, \mathcal{F}$ represents the trellis transitions between the trellis stages $(k-1)$ and k. The trellis has to be of finite length and for the case of MAP equalisation, this corresponds to the length $\mathcal{F}$ of the received sequence or the transmission burst. The branch transition probability $\gamma_k(s', s)$ can be expressed as the product of the *a priori* probability $P(s|s') = P(c_k)$ and the transition probability $p(v_k|s', s)$:

$$\gamma_k(s', s) = P(c_k) \cdot p(v_k|s', s). \tag{5.13}$$

The transition probability is given by:

$$p(v_k|s', s) = \frac{1}{\sqrt{2\pi\sigma_\eta^2}} \exp\left(-\frac{(v_k - \tilde{v}_k)^2}{2\sigma_\eta^2}\right), \tag{5.14}$$

where $\tilde{v}_k$ is the noiseless channel output, and the *a priori* probability of bit c_k being a logical 1 or a logical 0 can be expressed in terms of its LLR values according to Equation 5.6. Since the term $\frac{1}{\sqrt{2\pi\sigma_\eta^2}}$ in the transition probability expression of Equation 5.14 and the term $\frac{\exp(-\mathcal{L}_a^E(c_k)/2)}{1+\exp(-\mathcal{L}_a^E(c_k))}$ in the *a priori* probability formula of Equation 5.6 are constant over the summation in the numerator and denominator of Equation 5.9, they cancel out. Hence, the transition probability is calculated according to [105]:

$$\gamma_k(s', s) = w_k \cdot \gamma_k^*(s', s), \tag{5.15}$$

$$\gamma^*(s', s) = \exp\left(-\frac{|v_k - \tilde{v}_k|^2}{2\sigma_\eta^2}\right) \tag{5.16}$$

$$w_k = \exp\left(\frac{1}{2} \cdot c_k \cdot \mathcal{L}_a^E(c_k)\right). \tag{5.17}$$

Note the similarity of the transition probability of Equation 5.15 with the PDF of the RBF equaliser's ith symbol described by Equation 4.3, where the terms w_k and $\gamma^*(s', s)$ are the RBF's weight and activation function, respectively, while the number of RBF nodes n_s^i

is one. We also note that the computational complexity of both the MAP and the RBF equalisers can be reduced by representing the output of the equalisers in the logarithmic domain, utilizing the Jacobian logarithmic relationship [13] described in Equation 4.1. The RBF equaliser based on the Jacobian logarithm – highlighted in Section 4.2 – was hence termed as the Jacobian RBF equaliser.

The memory of the MAP equaliser is limited by the length of the trellis, provided that decisions about the kth transmitted symbol I_k are made in possession of the information related to all the received symbols of a transmission burst. In the MAP algorithm the recursive relationships of the forward and backward transition probabilities of Equation 5.11 and 5.12, respectively, allow us to avoid processing the entire received sequence $\mathbf{v}_k$ everytime the a *posteriori* LLR $\mathcal{L}_p^E(c_k)$ is evaluated from the joint probability $p(s', s, \mathbf{v}_k)$ according to Equation 5.9. This approach is different from that of the RBF based equaliser having a feedforward order of m, where the received sequence $\mathbf{v}_k$ of m-symbols is required each time the a *posteriori* LLR $\mathcal{L}_p^E(c_k)$ is evaluated using Equation 5.8. However, the MAP algorithm has to process the received sequence both in a forward and backward oriented fashion and store both the forward and backward recursively calculated transition probabilities $\alpha_k(s)$ and $\beta_k(s)$, before the LLR values $\mathcal{L}_p^E(c_k)$ can be calculated from Equation 5.9. The equaliser's delay facilitates invoking information from the 'future' samples $v_k, \ldots, v_{k-\tau+1}$ in the detection of the transmitted symbol $I_{k-\tau}$. In other words, the delayed decision of the MAP equaliser provides the necessary information concerning the 'future' samples $v_{j>k}$ – relative to the delayed kth decision – to be utilised and the information of the future samples is generated by the backward recursion of Equation 5.12.

The MAP equaliser exhibits optimum performance. However, if decision feedback is used in the RBF subset centre selection as in [26] or in the RBF space-translation as in Section 2.11.2, the performance of the RBF DFE TEQ in conjunction with the idealistic assumption of *correct* decision feedback is better, than that of the MAP TEQ due to the increased Euclidean distance between channel states, as it will be demonstrated in Section 5.5. However, this is not so for the more practical RBF DFE feeding back the detected symbols, which may be erroneous.

5.4 Comparison of the Jacobian RBF and Log-MAP Equaliser

Building on Section 5.3, in this section the Jacobian logarithmic algorithm is invoked, in order to reduce the computational complexity of the MAP algorithm. We denote the forward, backward and transition probability in the logarithmic form as follows:

$$A_k(s) = \ln(\alpha_k(s)) \tag{5.18}$$

$$B_k(s) = \ln(\beta_k(s)) \tag{5.19}$$

$$\Gamma_k(s', s) = \ln(\gamma_k(s', s)), \tag{5.20}$$

which we also used in Section 5.3. Thus, we could rewrite Equation 5.11 as:

$$\begin{aligned} A_k(s) &= \ln\left(\sum_{s'} \gamma_k(s', s) \cdot \alpha_{k-1}(s')\right) \\ &= \ln\left(\sum_{s'} \exp\left(\Gamma_k(s', s) + A_{k-1}(s')\right)\right), \end{aligned} \tag{5.21}$$

and Equation 5.12 as:

$$\begin{aligned} B_{k-1}(s') &= \ln\left(\sum_{s} \gamma_k(s', s) \cdot \beta_k(s)\right) \\ &= \ln\left(\sum_{s} \exp\left(\Gamma_k(s', s) + B_k(s)\right)\right). \end{aligned} \tag{5.22}$$

From Equation 5.21 and 5.22, the logarithmic-domain forward and backward recursion can be evaluated, once $\Gamma_k(s', s)$ was obtained. In order to evaluate the logarithmic-domain branch metric $\Gamma_k(s', s)$, Equations 5.15–5.17 and 5.20 are utilized to yield:

$$\Gamma_k(s', s) = -\frac{|v_k - \tilde{v}_k|^2}{2\sigma_\eta^2} + \frac{1}{2} \cdot c_k \cdot \mathcal{L}_a^E(c_k). \tag{5.23}$$

By transforming $\alpha_k(s)$, $\gamma_k(s', s)$ and $\beta_k(s)$ into the logarithmic domain in the Log-MAP algorithm, the expression for the LLR, $\mathcal{L}_p^E(c_k)$ in Equation 5.9 is also modified to yield:

$$\begin{aligned} \mathcal{L}_p^E(c_k) &= \ln\left(\frac{\sum_{(s',s)\Rightarrow c_k=+1} \alpha_{k-1}(s') \cdot \gamma_k(s', s) \cdot \beta_k(s)}{\sum_{(s',s)\Rightarrow c_k=-1} \alpha_{k-1}(s') \cdot \gamma_k(s', s) \cdot \beta_k(s)}\right) \\ &= \ln\left(\frac{\sum_{(s',s)\Rightarrow c_k=+1} \exp\left(A_{k-1}(s') + \Gamma_k(s', s) + B_k(s)\right)}{\sum_{(s',s)\Rightarrow c_k=-1} \exp\left(A_{k-1}(s') + \Gamma_k(s', s) + B_k(s)\right)}\right) \\ &= \ln\left(\sum_{(s',s)\Rightarrow c_k=+1} \exp\left(A_{k-1}(s') + \Gamma_k(s', s) + B_k(s)\right)\right) \\ &\quad - \ln\left(\sum_{(s',s)\Rightarrow c_k=-1} \exp\left(A_{k-1}(s') + \Gamma_k(s', s) + B_k(s)\right)\right). \end{aligned} \tag{5.24}$$

In the trellis of Figure 5.3 there are $\mathcal{M}$ possible transitions from state s' to all possible states s or to state s from all possible states s'. Hence, there are $\mathcal{M} - 1$ summations of the exponentials in the forward and backward recursion of Equation 5.21 and 5.22, respectively. Using the Jacobian logarithmic relationship of Equation 4.2, $\mathcal{M} - 1$ summations of the exponentials requires 2($\mathcal{M}$-1) additions/subtractions, ($\mathcal{M} - 1$) maximum search operations

and $(\mathcal{M}-1)$ table look-up steps. Together with the $\mathcal{M}$ additions necessitated to evaluate the term $\Gamma_k(s',s)+A_{k-1}(s')$ and $\Gamma_k(s',s)+B_k(s)$ in Equation 5.21 and 5.22, respectively, the forward and backward recursion requires a total of $(6\mathcal{M}-4)$ additions/subtractions, $2(\mathcal{M}$-1) maximum search operations and $2(\mathcal{M}$-1) table look-up steps. Assuming that the term $\frac{1}{2}\cdot c_k\cdot\mathcal{L}_a^E(c_k)$ in Equation 5.23 is a known weighting coefficient, evaluating the branch metrics given by Equation 5.23 requires a total of 2 additions/subtractions, 1 multiplication and 1 division.

By considering a trellis having χ number of states at each trellis stage and $\mathcal{M}$ legitimate transitions leaving each state, there are $\frac{1}{2}\mathcal{M}\chi$ number of transitions due to the bit $c_k=+1$. Each of these transitions belongs to the set $(s',s)\Rightarrow c_k=+1$. Similarly, there will be $\frac{1}{2}\mathcal{M}\chi$ number of $c_k=-1$ transitions, which belong to the set $(s',s)\Rightarrow c_k=-1$. Evaluating $A_k(s)$, $B_{k-1}(s')$ and $\Gamma_k(s',s)$ of Equation 5.21, 5.22 and 5.23, respectively, at each trellis stage k associated with a total of $\mathcal{M}\chi$ transitions requires $\mathcal{M}\chi(6\mathcal{M}-2)$ additions/subtractions, $\mathcal{M}\chi(2\mathcal{M}-2)$ maximum search operations, $\mathcal{M}\chi(2\mathcal{M}-2)$ table look-up steps, plus $\mathcal{M}\chi$ multiplications and $\mathcal{M}\chi$ divisions. With the terms $A_k(s)$, $B_{k-1}(s')$ and $\Gamma_k(s',s)$ of Equations 5.21, 5.22 and 5.23 evaluated, computing the LLR $\mathcal{L}_p^E(c_k)$ of Equation 5.24 using the Jacobian logarithmic relationship of Equation 4.2 for the summation terms $\ln(\sum_{(s',s)\Rightarrow c_k=+1}\exp(\cdot))$ and $\ln(\sum_{(s',s)\Rightarrow c_k=-1}\exp(\cdot))$ requires a total of $4(\frac{1}{2}\mathcal{M}\chi-1)+2\mathcal{M}\chi+1$ additions/subtractions, $\mathcal{M}\chi-2$ maximum search operations and $\mathcal{M}\chi-2$ table look-up steps. The number of states at each trellis stage is given by $\chi=\mathcal{M}^L=n_{s,f}/\mathcal{M}$. Therefore, the total computational complexity associated with generating the *a posteriori* LLRs using the Jacobian logarithmic relationship for the Log-MAP equaliser is given in Table 5.1.

	Log-MAP	Jacobian RBF
subtraction and addition	$n_{s,f}(6\mathcal{M}+2)-3$	$n_{s,f}+\mathcal{M}n_s^i(m+2)-4$
multiplication	$n_{s,f}$	$n_{s,f}$
division	$n_{s,f}$	$n_{s,f}$
max	$n_{s,f}(2\mathcal{M}-1)-2$	$\mathcal{M}n_s^i-2$
table look-up	$n_{s,f}(2\mathcal{M}-1)-2$	$\mathcal{M}n_s^i-2$

Table 5.1: Computational complexity of generating the *a posteriori* LLR $\mathcal{L}_p^E$ for the Log-MAP equaliser and the Jacobian RBF equaliser [9]. The RBF equaliser order is denoted by m and the number of RBF centres is n_s^i. The notation $n_{s,f}=\mathcal{M}^{L+1}$ indicates the number of trellis states for the Log-MAP equaliser and also the number of scalar channel states for the Jacobian RBF equaliser.

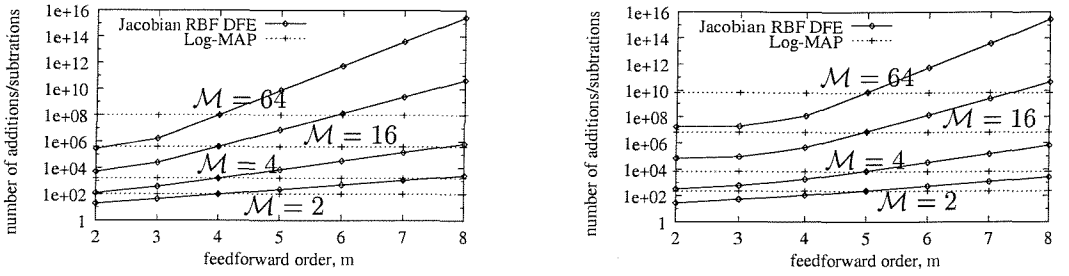
For the Jacobian RBF equaliser, the LLR expression of Equation 5.8 is rewritten in terms

of the logarithmic form $\ln\left(f^i_{RBF}(\mathbf{v}_k)\right)$ to yield:

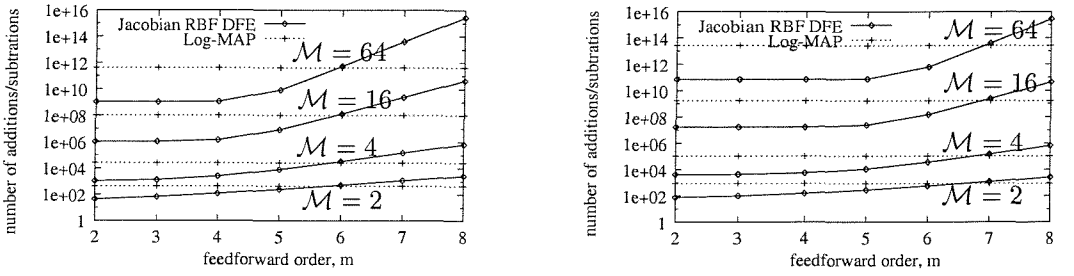
$$\begin{aligned}
\mathcal{L}_p^E(c_k) &= \ln\left(\frac{\sum_{\substack{i\\ c_k=+1}} f^i_{RBF}(\mathbf{v}_k)}{\sum_{\substack{i\\ c_k=-1}} f^i_{RBF}(\mathbf{v}_k)}\right) \\
&= \ln\left(\frac{\sum_{\substack{i\\ c_k=+1}} \exp\left(\ln(f^i_{RBF}(\mathbf{v}_k))\right)}{\sum_{\substack{i\\ c_k=-1}} \exp\left(\ln(f^i_{RBF}(\mathbf{v}_k))\right)}\right) \\
&= \ln\left(\sum_{\substack{i\\ c_k=+1}} \exp\left(\ln(f^i_{RBF}(\mathbf{v}_k))\right)\right) - \ln\left(\sum_{\substack{i\\ c_k=-1}} \exp\left(\ln(f^i_{RBF}(\mathbf{v}_k))\right)\right) \quad (5.25)
\end{aligned}$$

The summation of the exponentials in Equation 5.25 requires $2(\mathcal{M}\text{-}2)$ additions/subtractions, $(\mathcal{M}\text{-}2)$ table look-up and $(\mathcal{M}-2)$ maximum search operations. The associated complexity of evaluating the conditional PDF of $\mathcal{M}$ symbols in logarithmic form according to Equation 4.4 was given in Table 4.1. Therefore, – similarly to the Log-MAP equaliser – the computational complexity associated with generating the *a posteriori* LLR $\mathcal{L}_p^E$ for the Jacobian RBF equaliser is given in Table 5.1. Figure 5.4 compares the number of additions/subtractions per turbo iteration involved in evaluating the *a posteriori* LLRs $\mathcal{L}_p^E$ for the Log-MAP equaliser and Jacobian RBF equaliser according to Table 5.1. More explicitly, the complexity is evaluated upon with varying the feedforward order m for different values of L, where $(L+1)$ is the CIR duration under the assumption that the feedback order $n = L$ and the number of RBF centres is $n_s^i = \mathcal{M}^{m+L-n}/\mathcal{M}$. Since the number of multiplications and divisions involved is similar, and by comparison, the number of maximum search and table look-up stages is insignificant, the number of additions/subtractions incurred in Figure 5.4 approximates the relative computational complexities involved. Figure 5.4 shows significant computational complexity reduction upon using Jacobian RBF equalisers of relatively low feedforward order, especially for higher-order modulation modes, such as $\mathcal{M} = 64$. The figure also shows an exponential increase of the computational complexity, as the CIR length increases. Observe in Figure 5.4 that as a rule of thumb, the feedforward order of the Jacobian RBF DFE must not exceed the CIR length $(L+1)$ in order to achieve a computational complexity improvement relative to the Log-MAP equaliser, provided that we use the optimal number of RBF centres, namely $n_s^i = \mathcal{M}^{m+L-n}/\mathcal{M}$.

The length of the trellis determines the storage requirements of the Log-MAP equaliser, since the Log-MAP algorithm has to store both the forward- and backward-recursively calculated metrics $A_k(s)$ and $B_{k-1}(s')$ before the LLR values $\mathcal{L}_p^E(c_k)$ can be calculated. For the Jacobian RBF DFE, we have to store the value of the RBF centres and the storage requirements will depend on the CIR length $L+1$ and on the modulation mode characterised by $\mathcal{M}$.



(a) L = 2 (b) L = 3



(c) L = 4 (d) L = 5

Figure 5.4: Number of additions/subtractions per iteration for the Jacobian RBF DFE of varying equaliser order m and the Log-MAP equaliser for various values of L, where $L+1$ is the CIR length. The feedback order of the Jacobian RBF DFE is set to $n = L$ and the number of RBF centres is set to $n_s^i = \mathcal{M}^{m+L-n}/\mathcal{M}$.

5.5 RBF Turbo Equaliser Performance

The schematic of the entire system was shown in Figure 5.2, where the transmitted source bits are convolutionally encoded, channel-interleaved and mapped to an $\mathcal{M}$-ary modulated symbol. The encoder utilized a half-rate recursive systematic convolutional (RSC) code, having a constraint length of $K = 5$ and octal generator polynomials of $G_0 = 35$ and $G_1 = 23$. A random channel interleaver of 20 000-bit memory was invoked. The transmission burst structure used in this system is the FMA1 non-spread speech burst, as specified in the Pan-European FRAMES proposal [101], which is seen in Figure 5.5. We have assumed that perfect knowledge of the CIR was available, which implies that our results portray the best-case performance.

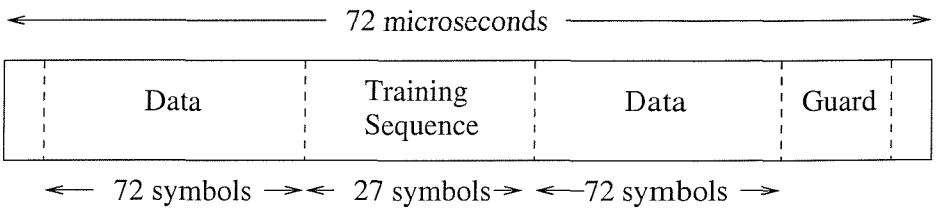
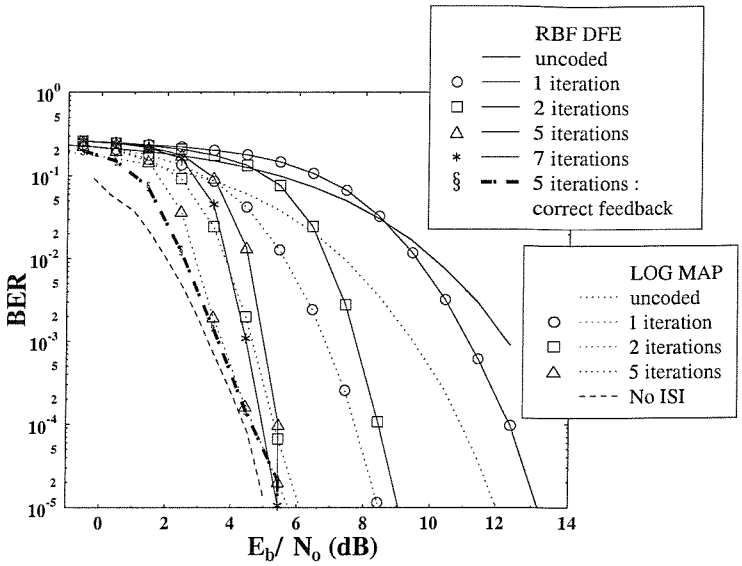


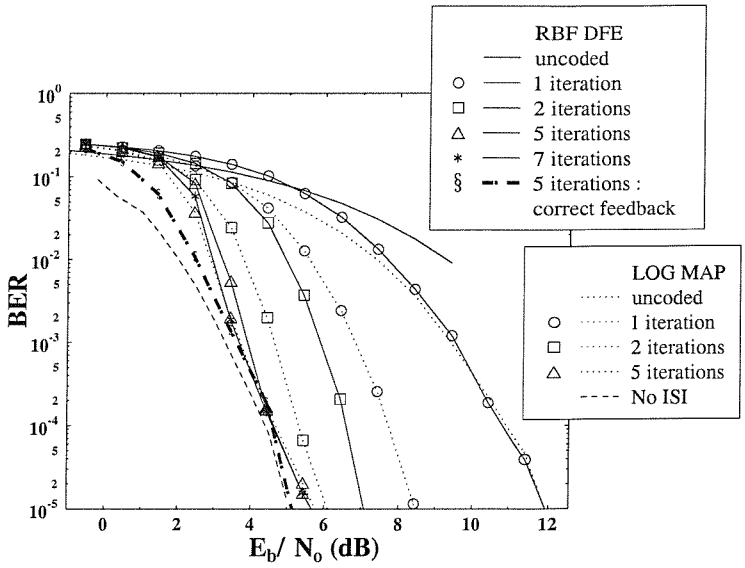
Figure 5.5: Transmission burst structure of the so-called FMA1 nonspread speech mode as specified in the FRAMES proposal [101].

5.5.1 Dispersive Gaussian Channels

The performance of the Jacobian RBF DFE TEQ was initially investigated over a dispersive Gaussian channel. Figure 5.6 provides the BER performance comparison of the Log-MAP and Jacobian RBF DFEs in the context of turbo equalisation. Various equaliser orders were used over a three-path Gaussian channel having a z-domain transfer function of $F(z) = 0.5773 + 0.5773z^{-1} + 0.5773z^{-2}$ and employing BPSK. Figure 5.6(b) shows that when the feedback information is not error-free, the Log-MAP TEQ outperforms the Jacobian RBF DFE TEQ for the same number of iterations. The corresponding uncoded systems using the Log-MAP equaliser and the Jacobian RBF DFE exhibit similar performance trends. Comparing Figure 5.6(a) for the equaliser parameters of $m = 3$, $n = 2$ and $\tau = 2$, as well as Figure 5.6(b) for the equaliser parameters of $m = 4$, $n = 2$ and $\tau = 3$, we observe that the performance of the Jacobian RBF DFE TEQ improves, as the feedforward order and the decision delay of the equaliser increases. This is achieved at the expense of increased computational complexities as evidenced by Figure 5.4. The above trend is a consequence of the enhanced DFE performance in conjunction with increasing feedforward order and decision delay, as it was demonstrated and justified in Section 2.11. However, as seen in Table 5.1, the approximate number of additions/subtrations for the Jacobian RBF DFE



(a) The Jacobian RBF DFE has a feedforward order of $m = 3$, feedback order of $n = 2$ and decision delay of $\tau = 2$ symbols.



(b) The Jacobian RBF DFE has a feedforward order of $m = 4$, feedback order of $n = 2$ and decision delay of $\tau = 3$ symbols.

Figure 5.6: Performance of the Log-MAP TEQ and Jacobian RBF DFE TEQ over the three-path Gaussian channel having a z-domain transfer function of $F(z) = 0.5773 + 0.5773z^{-1} + 0.5773z^{-2}$ for BPSK.

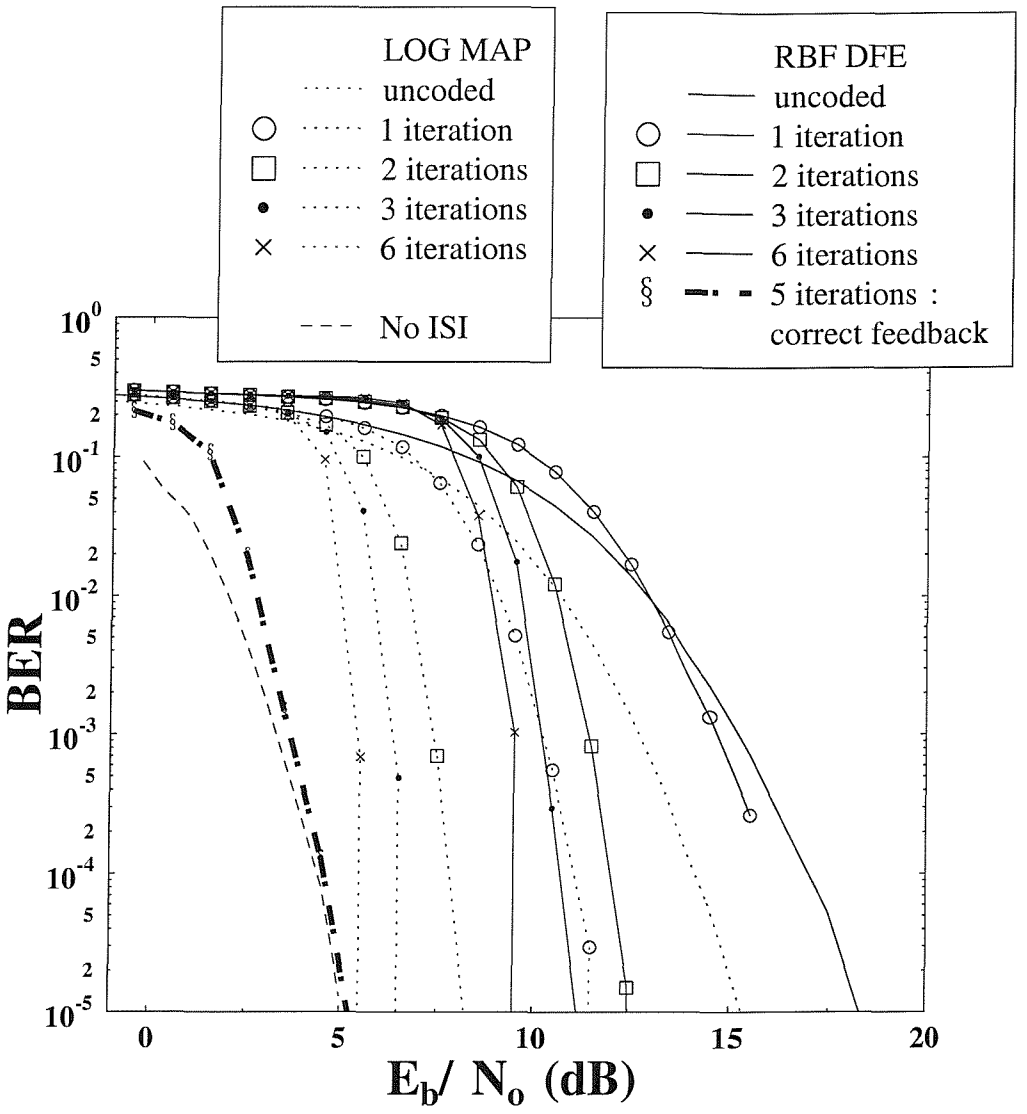


Figure 5.7: Performance of the Log-MAP TEQ and Jacobian RBF DFE TEQ over the five-path Gaussian channel having a z-domain transfer function of $F(z) = 0.227 + 0.46z^{-1} + 0.688z^{-2} + 0.46z^{-3} + 0.227z^{-4}$ for BPSK. The Jacobian RBF DFE has a feedforward order of $m = 5$, feedback order of $n = 4$ and decision delay of $\tau = 4$ symbols.

increased from 44 to 100 for a feedforward order increase from $m = 3$ to $m = 4$. Both the Log-MAP and the Jacobian RBF DFE TEQs converge to a similar BER performance upon increasing the number of iterations. The Log-MAP TEQ performs better, than the Jacobian RBF DFE TEQ at a lower number of iterations, as shown in Figure 5.6. This is, because effectively the Log-MAP equaliser has a higher feedforward order, which is equivalent to the length of the trellis and also exhibits a longer decision delay, as discussed in Section 5.3. The performance of the Log-MAP TEQ in the zero-ISI – i.e. non-dispersive – Gaussian channel environment was also presented in Figure 5.6(b) for comparison. The Log-MAP TEQ, the Jacobian RBF DFE TEQ using $m = 4$, $n = 2$, $\tau = 3$ and the Jacobian RBF DFE TEQ employing $m = 3$, $n = 2$, $\tau = 2$ performed within approximately 0.2dB, 0.2dB and 0.5dB, respectively, from this zero-ISI, i.e. non-dispersive AWGN benchmarker at BER of 10^{-4}. The BER performance of the RBF DFE TEQ using *correct* decision fedback is also shown in Figure 5.6, which exhibits a better performance than the Log-MAP TEQ. This is possible – although the Log-MAP equaliser is known to approximate the optimal performance – because the RBF DFE's subset centre selection mechanism creates an increased Euclidean distance between the channel states [26] and effectively eliminates the postcursor ISI, which improves the performance of the Jacobian RBF DFE TEQ.

The performance of the TEQs was then investigated over a dispersive Gaussian channel having an increased CIR length. Figure 5.7 compares the performance of the Log-MAP TEQ and the Jacobian RBF DFE ($m = 5$, $n = 4$, $\tau = 4$) TEQ over the five-path Gaussian channel associated with the transfer function of $F(z) = 0.227 + 0.46z^{-1} + 0.688z^{-2} + 0.46z^{-3} + 0.227z^{-4}$. The performance of both the Log-MAP and Jacobian RBF DFE TEQs degrades with increasing CIR lengths, especially at lower SNRs, when we compare Figures 5.6 and 5.7. This is due to the increased number of multipath components to be resolved, when the CIR length is increased, a phenomenon which was also demonstrated in Figures 2.32 and 2.33 for an uncoded RBF DFE over the three-path and five-path channels, respectively. For the five-path channel, the Log-MAP TEQ and the Jacobian RBF DFE TEQ using $m = 5$, $n = 4$, $\tau = 4$ performed within about 1dB and 5dB, respectively, from the zero-ISI, non-dispersive Gaussian limit at a BER of 10^{-4}. We observed from Figure 5.6(b) and 5.7, that the coded BERs only start to decrease once the uncoded BERs reached approximately 2×10^{-1}.

5.5.2 Dispersive Rayleigh Fading Channels

Let us now investigate the performance of the TEQs in a dispersive Rayleigh fading channel environment. A three-path, symbol-spaced fading channel of equal weights was utilized, where the Rayleigh fading statistics obeyed a normalised Doppler frequency of 1.5×10^{-4}.

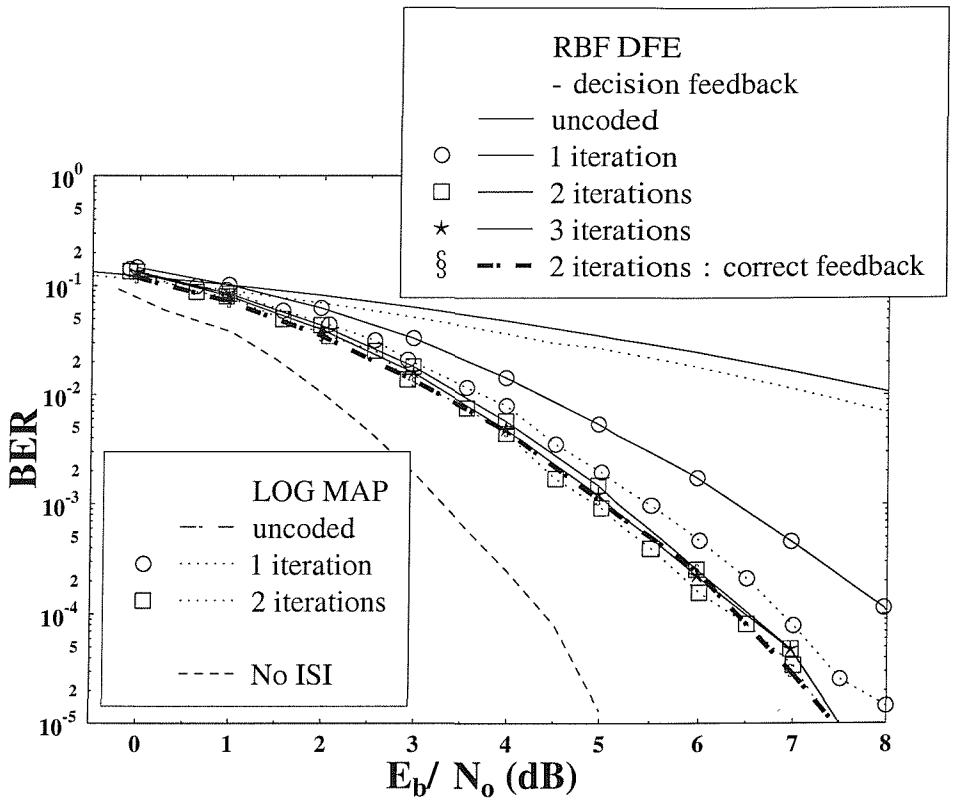


Figure 5.8: Performance of the Log-MAP TEQ and Jacobian RBF DFE TEQ over the three-path Rayleigh fading channel for **BPSK**. The Jacobian RBF DFE has a feedforward order of $m = 3$, feedback order of $n = 2$ and decision delay of $\tau = 2$ symbols.

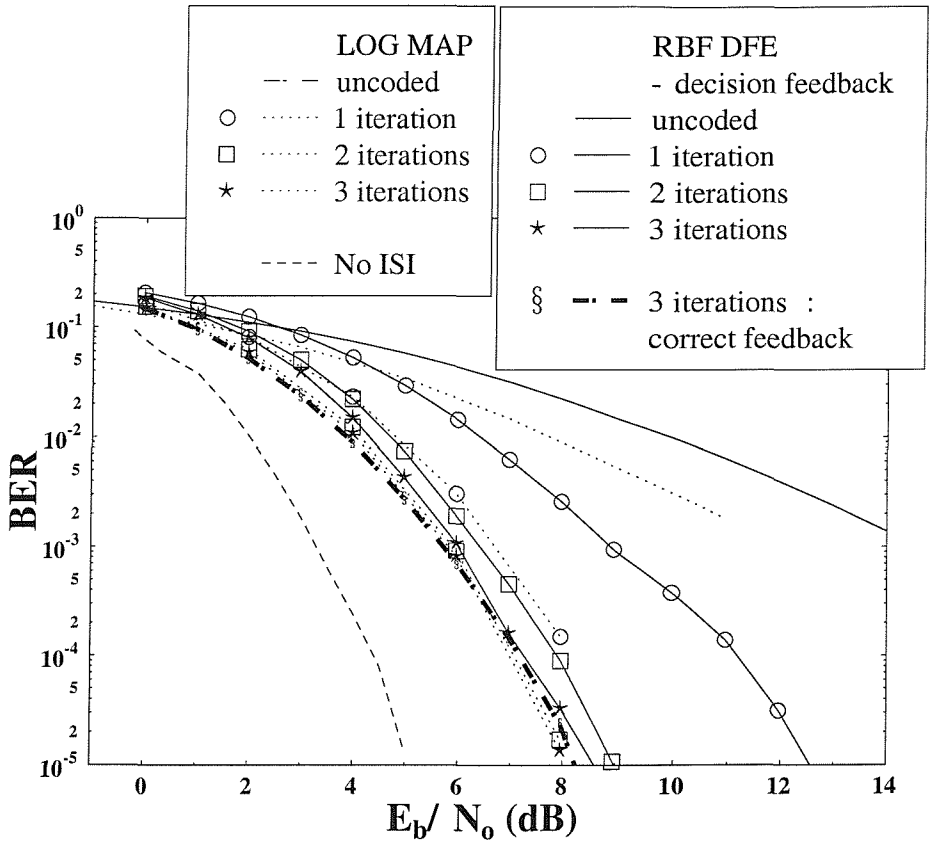


Figure 5.9: Performance of the Log-MAP TEQ and Jacobian RBF DFE TEQ over the three-path Rayleigh fading channel for **4-QAM**. The Jacobian RBF DFE has a feedforward order of $m = 3$, feedback order of $n = 2$ and decision delay of $\tau = 2$ symbols.

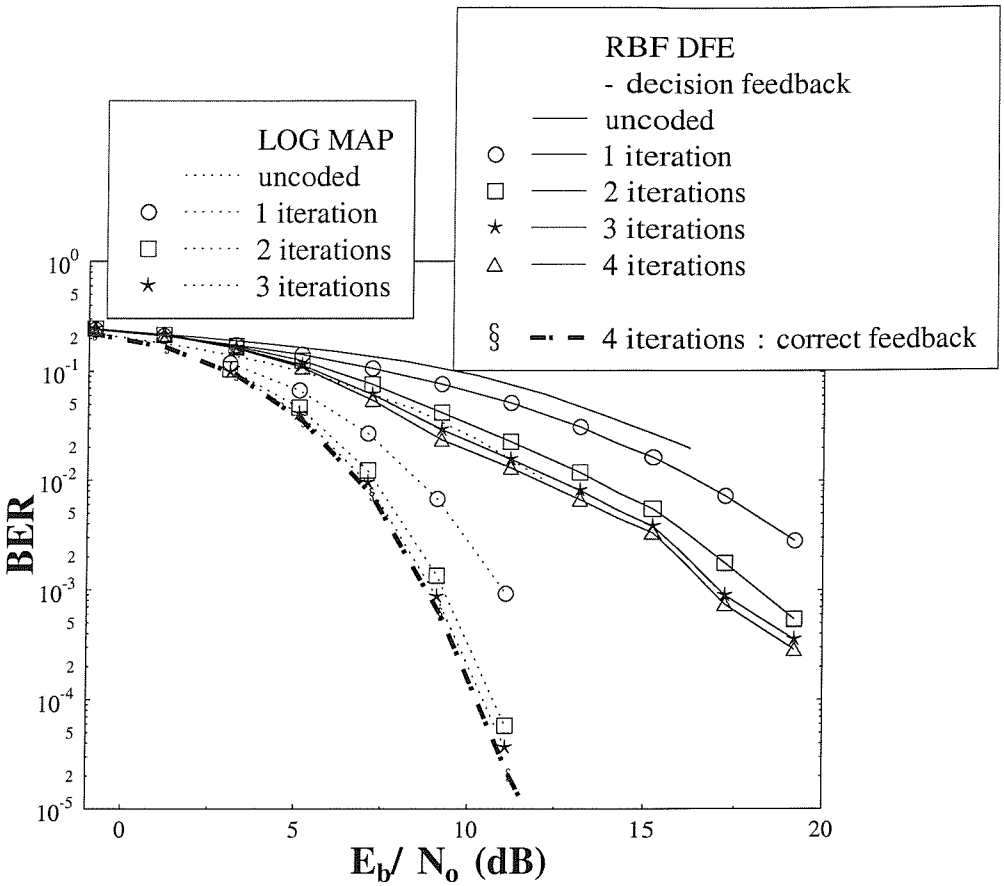


Figure 5.10: Performance of the Log-MAP TEQ and Jacobian RBF DFE TEQ over the three-path Rayleigh fading channel for **16-QAM**. The Jacobian RBF DFE has a feedforward order of $m = 3$, feedback order of $n = 2$ and decision delay of $\tau = 2$ symbols.

The CIR was assumed to be burst-invariant. Figure 5.8, 5.9 and 5.10 portray the performance of the Log-MAP TEQ and that of the Jacobian RBF DFE TEQ for BPSK, 4-QAM and 16-QAM, respectively. The Jacobian RBF DFE has a feedforward order of $m = 3$, feedback order of $n = 2$ and decision delay of $\tau = 2$ symbols. Figure 5.8 and Figure 5.9 show for BPSK and 4-QAM, that the Log-MAP TEQ and the Jacobian RBF DFE TEQ converge to a similar BER performance, but the Log-MAP TEQ requires a lower number of iterations. Specifically, two iterations are required for the Log MAP TEQ and three iterations for the Jacobian RBF DFE TEQ to achieve near-perfect convergence, since the Log-MAP TEQ exhibited a better BER performance for an uncoded system than the Jacobian RBF DFE. The performance of the Log-MAP TEQ at two iterations and that of the Jacobian RBF DFE TEQ at three iterations is about 2dB and 2.5dB away from the zero-ISI Gaussian BER curve for BPSK and 4-QAM, respectively, at a BER of 10^{-4}. For 16-QAM, the effect of error propagation degrades the performance of the Jacobian RBF DFE TEQ by 10dB at BER of 10^{-4}, when we compare the Jacobian RBF DFE TEQ's correct feedback based and decision feedback assisted performance after 4 iterations, as seen in Figure 5.10. Again, the performance can be improved by increasing the equaliser feedforward order at the expense of higher computational complexity, as discussed in Section 5.5.1.

The iteration gain of the Jacobian RBF DFE TEQ after 3 iterations at a BER of 10^{-3} was 1.3dB, 3dB and 4dB for the modulation modes of BPSK, 4-QAM and 16-QAM, respectively. By contrast, for the Log-MAP TEQ the corresponding iteration gains were 0.5dB, 0.9dB and 2dB for the modulation modes of BPSK, 4-QAM and 16-QAM, respectively. Explicitly, the iteration gain was defined as the difference between the channel SNR required in order to achieve a certain BER after one iteration and the corresponding channel SNR required after n number of iterations. The iteration gain was higher for the higher-order modulation modes, since the distance between two neighbouring points in the higher-order constellations was lower and hence it was more gravely affected by ISI and noise.

Since the computation of the associated implementational complexity summarised in Table 5.1 is quite elaborate, here we only give an estimate of the Log-MAP TEQ's and the Jacobian RBF DFE TEQ's complexity in the context of both BPSK and 4-QAM, employing the parameters used in our simulations. Specifically, in the BPSK scheme the approximate number of additions/subtractions and multiplications/divisions for the Log-MAP TEQ was 109 and 16 per iteration, respectively, whereas for the Jacobian RBF DFE TEQ ($m = 3$, $n = 2$, $\tau = 2$) the corresponding figures were 44 and 16, respectively. The 'per iteration' complexity of the Jacobian RBF DFE TEQ was approximately a factor of (109/44 =)2.5, 4.4 and 16.3 lower, than that of the Log-MAP TEQ, for BPSK, 4-QAM and 16-QAM, respectively.

Overall, due to the error propagation that gravely degrades the performance of the Jacobian RBF DFE TEQ when using 16-QAM, the Jacobian RBF DFE TEQ could only provide a practical performance versus complexity advantage for lower order modulation modes, such as BPSK and 4-QAM. It is worth noting here that we have attempted using the LLR values output by the decoder in the previous iteration as the feedback information for the feedback section of the RBF DFE. However, we attained an inferior performance compared to the scenario using the RBF DFE outputs as the feedback information. This is because the BER improves on every iterations and the BER of the input of the equaliser fed back from the decoder was improved after equalisation. Therefore the output of the equaliser was more reliable, than the output of the decoder in the previous iteration. Turbo equalisation research has been focused on developing reduced complexity equalisers, such as the receiver structure proposed by Glavieux *et. al.* [106], where the equaliser is constituted by two linear filters. Motivated by this trend, Yeap *et. al.* [109, 110, 111] proposed a reduced complexity trellis-based equaliser scheme based on equalising the in-phase and quadrature-phase component of the transmitted signal independently. This novel reduced complexity equaliser is termed as the In-Phase/Quadrature-phase Equaliser (I/Q EQ). When a channel having a memory of L symbol durations was encountered, the trellis-based equaliser must consider $\mathcal{M}^{L+1}$ total number of transitions at each trellis stage, as discussed in Section 5.4. The complexity of the complex-valued trellis-based equaliser increased rapidly with L. However, by removing the associated cross-coupling of the in-phase and quadrature-phase signal components and hence rendering the channel output to be only dependent on either quadrature component, the number of transitions considered was reduced to $(\sqrt{\mathcal{M}})^{L+1}$. Therefore, there will be an I/Q EQ for each I/Q component, subtituting the original trellis-based equaliser and giving a complexity reduction factor of $\frac{\mathcal{M}^{L+1}}{2\times\sqrt{\mathcal{M}}^{L+1}} = 0.5 \times \sqrt{\mathcal{M}^{L+1}}$. The TEQ using I/Q EQs was capable of achieving the same performance as the Log-MAP TEQ for 4-QAM and 16-QAM, while maintaining a complexity reduction factor of 2.67 and 16, respectively, over the equally-weighted three-path Rayleigh fading channel using a normalised Doppler frequency of 3.3×10^{-5} [109, 110, 111]. The complexity of the RBF DFE could be similarly reduced to that of the I/Q EQ by equalising the in-phase and quadrature-phase components of the transmitted signal separately. In the following section, we proposed another novel method of reducing the complexity of TEQ by making use of the fact that the RBF DFE evaluates its output on a symbol-by-symbol basis.

5.6 Reduced-complexity RBF Assisted Turbo Equalisation

The Log-MAP algorithm requires forward and backward recursions through the entire sequence of symbols in the received burst in order to evaluate the forward and backward

transition probability of Equation 5.11 and 5.12, before calculating the *a posteriori* LLR values $\mathcal{L}_p(c_k)$. Therefore, effectively the computation of the *a posteriori* LLRs $\mathcal{L}_p(c_k)$ is performed on a burst-by-burst basis. The RBF based equaliser, however, performs the evaluation of the *a posteriori* LLRs $\mathcal{L}_p(c_k)$ on a symbol-by-symbol basis. Therefore, in order to reduce the associated computational complexity, the RBF based TEQ may skip evaluating the symbol LLRs according to Equation 5.8 in the current iteration, when the symbol has a low error probability or high *a priori* LLR magnitude $|\mathcal{L}_a^E(c_k)|$ after channel decoding in the previous iteration. If, however this is not the case, the equaliser invokes a further iteration and attempts to improve the decoder's reliablility estimation of the coded bits. The output $f_{RBF}^i(\mathbf{v}_k)$ of the RBF equaliser provides the likelihood of the ith symbol at instant k. The log-likelihood values of the ith symbol provided by the channel decoder in the previous iteration obey an approximately linear relationship versus the log-likelihood values from the equaliser in the current iteration, as demonstrated in Figure 5.11 for the BPSK mode over a three-path, symbol-spaced fading channel of equal CIR tap weights, where the Rayleigh fading statistics obeyed a normalised Doppler frequency of 1.5×10^{-4}. Therefore, the logarithmic domain output $\ln\left(\tilde{f}_{RBF}^i(\mathbf{v}_k)\right)$ of the RBF equaliser can be estimated based on this near-linear relationship portrayed in Figure 5.11 according to:

$$\ln\left(\tilde{f}_{RBF}^i(\mathbf{v}_k)\right) = g \cdot \ln(L_a(I_k = \mathcal{I}_i)) + c, \tag{5.26}$$

where $\ln(L_a(I_k = \mathcal{I}_i))$ is the log-likelihood of the transmitted symbol I_k being the ith QAM symbol $\mathcal{I}_i$ based on the decoder's soft output, g is the log-likelihood gradient and c is the log-likelihood intercept point. Both g and c can be inferred from Figure 5.11. As our next action, we have to set the LLR magnitude threshold $|\mathcal{L}|_{threshold}$, where the estimated coded bits c_k output by the decoder in the previous iteration become sufficiently reliable for refraining from further iterations. Hence the symbols exhibiting an LLR value above this threshold are not fed back to the equaliser for futher iterations, since they can be considered sufficiently reliable for subjecting them to hard decision. The LLRs passed to the decoder from the equaliser are calculated from the symbols' log-likelihood values based on the linear relationship of Equation 5.26 instead of the more computationally demanding Equation 5.8, in order to reduce the computational complexity. We refer to this RBF based-TEQ as the *reduced-complexity RBF TEQ*.

In our experiments, the above mentioned log-likelihood gradient and the intercept point were found to be $g = 1.2$ and $c = -7.5$, respectively, according to the near-linear relationship of Figure 5.11. We set the LLR magnitude threshold $|\mathcal{L}|_{threshold}$ such that the symbols in the burst that were not fed back to the equaliser for further iterations became sufficiently reliable and hence exhibited a low probability of decoding error. The threshold was initially set to $|\mathcal{L}|_{threshold} = 10$ based on our experiments, such that the symbols that were not fed back

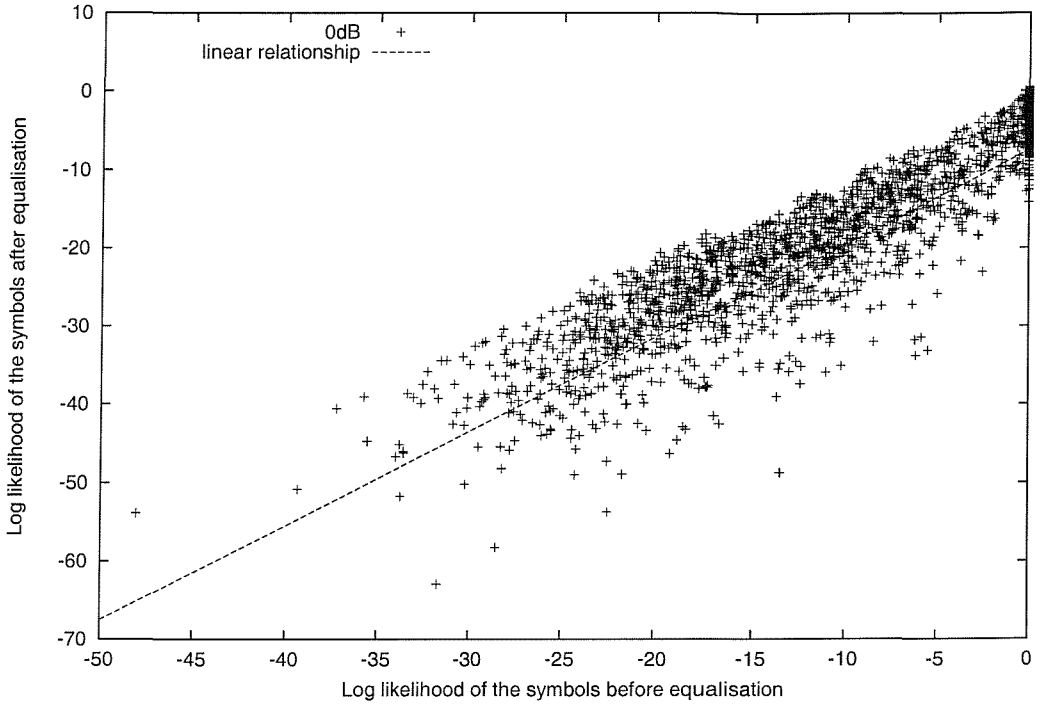


Figure 5.11: The log-likelihood of the RBF turbo equalised symbols before and after equalisation over the three-path, symbol-spaced fading channel of equal CIR tap weights, where the Rayleigh fading statistics obeyed a normalised Doppler frequency of 1.5×10^{-4}, at an SNR of 0dB using BPSK.

to the decoder exhibited a probability of error below 5×10^{-5} according to Equation 4.13. Figures 5.12 and 5.13 compare the performance of the reduced-complexity Jacobian RBF DFE TEQ to that of the Jacobian RBF DFE TEQ of Section 5.4 over the three-path Gaussian channel having a transfer function of $F(z) = 0.5773 + 0.5773z^{-1} + 0.5773z^{-2}$. The reduced-complexity Jacobian RBF DFE TEQ provides an equivalent BER performance to that of the Jacobian RBF DFE TEQ of Section 5.4, while exhibiting a reduced computational complexity, which is proportional to the percentage of the BPSK symbols fed back for further iterations in Figure 5.12 and 5.13. We note that in our experiments the reduced-complexity Jacobian RBF DFE TEQ using the detected decision feedback – rather than error-free feedback – required a higher LLR magnitude threshold of $|\mathcal{L}|_{threshold} = 26$ (which guaranteed a probability of error of 5×10^{-12} according to Equation 4.13), in order to provide an equivalent BER performance to that of the Jacobian RBF DFE TEQ, since the decision feedback error propagation reduced the decoder's reliability estimation of the coded bits. The higher the LLR magnitude threshold, the higher the percentage of bits fed back, resulting in a higher complexity. According to Figure 5.12 depicting the performance of the reduced-complexity Jacobian RBF DFE TEQ relying on correct decision feedback, the average percentage of bits not requiring further iterations for a channel SNR of 4dB was

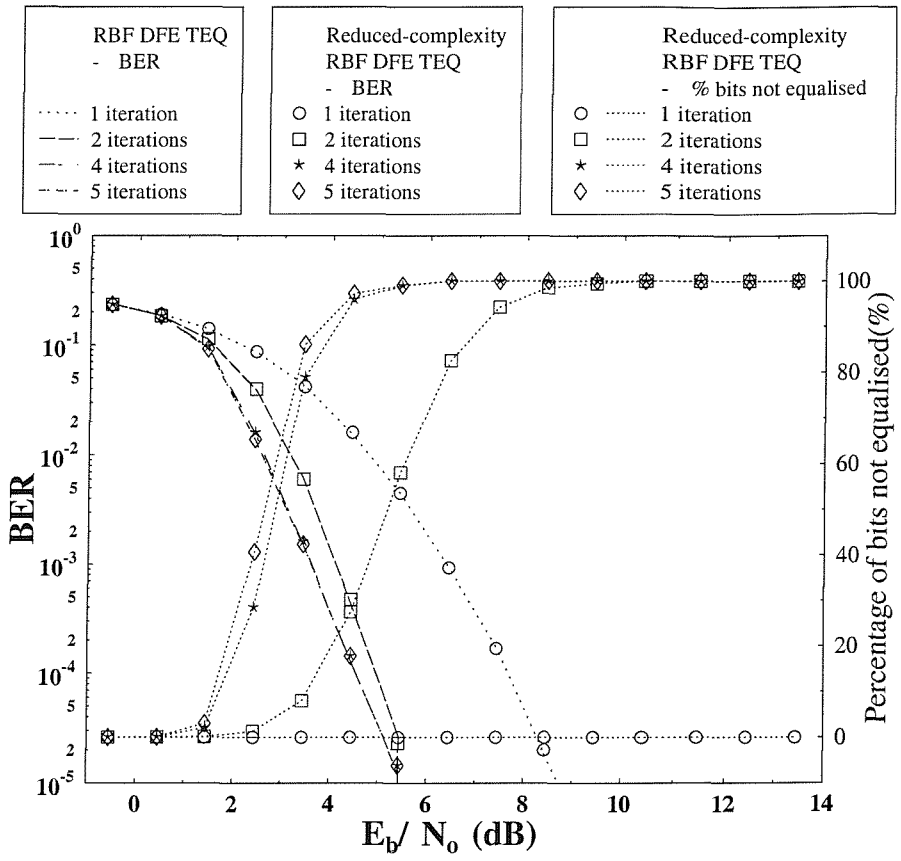


Figure 5.12: BER performance and the percentage of symbols not requiring equalisation by the reduced-complexity RBF TEQ using **correct** decision feedback over the three-path Gaussian channel having a z-domain transfer function of $F(z) = 0.5773 + 0.5773z^{-1} + 0.5773z^{-2}$ for BPSK. The LLR magnitude threshold, the log-likelihood gradient and the log-likelihood intercept point were set to $|\mathcal{L}|_{threshold} = 10$, $g = 1.2$ and $c = -7.5$, respectively.

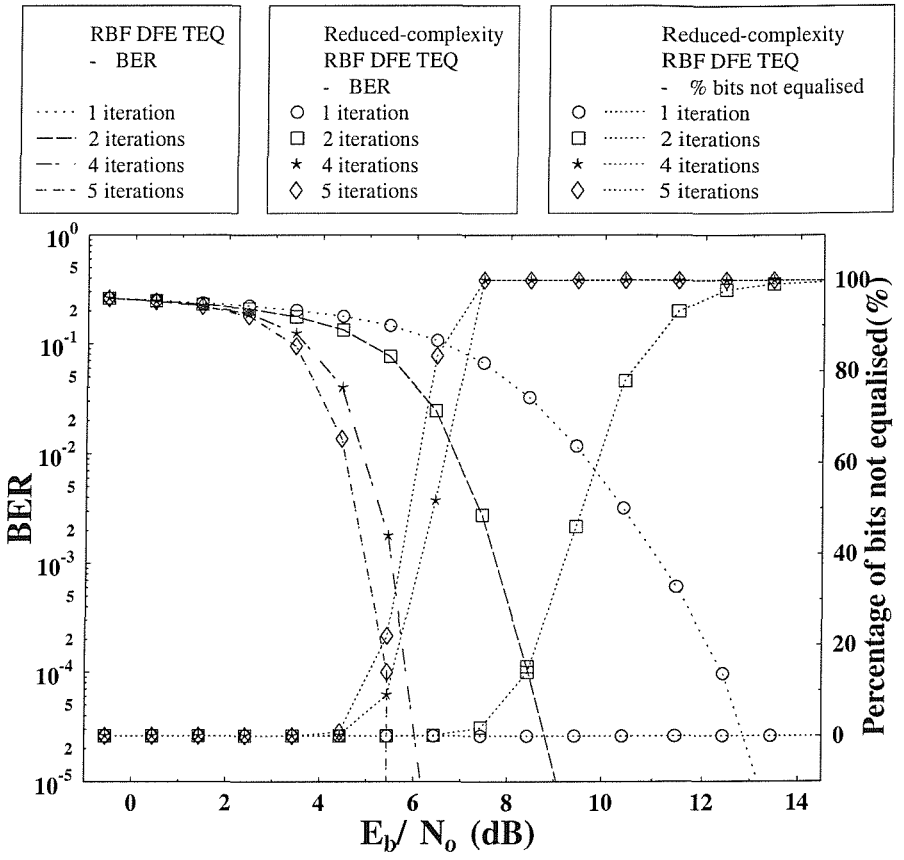


Figure 5.13: BER performance and the percentage of symbols not requiring equalisation by the reduced-complexity RBF TEQ using **detected** decision feedback over the three-path Gaussian channel having a z-domain transfer function of $F(z) = 0.5773 + 0.5773z^{-1} + 0.5773z^{-2}$ for BPSK. The LLR magnitude threshold, the log-likelihood gradient and the log-likelihood intercept point were set to $|\mathcal{L}|_{threshold} = 26$, $g = 1.2$ and $c = -7.5$, respectively.

20% for the second iteration, 70% for third iteration and approximately 90% for the consecutive iterations. This amounts to a total of approximately 54% computational complexity reduction at the SNR of 4dB. Referring to Figure 5.13, the reduced-complexity Jacobian RBF DFE TEQ relying on detected symbol-based – rather than perfect – decision feedback with its associated higher LLR magnitude threshold provides a total of approximately 21% computational reduction at an SNR of 6dB. Figure 5.14 depicts the performance of the reduced-complexity Jacobian RBF DFE TEQ relying on detected decision feedback over the three-tap equal gain, symbol-spaced Rayleigh faded CIR obeying a Doppler frequency of 1.5×10^{-4}. A LLR magnitude threshold of 10 was sufficient for the reduced-complexity Jacobian RBF DFE TEQ in order to provide an equivalent BER performance to that of the Jacobian RBF DFE TEQ. The RBF DFE provided a better reliability-estimation over the dispersive burst-invariant Rayleigh fading channel compared to the dispersive Gaussian channel, since the uncoded BER performance was better over the Rayleigh fading channel, as it is seen from comparing Figures 5.13 and 5.14. [1] Hence less errors were propagated from the equaliser's decision feedback to future bits. Referring to Figure 5.14, the reduced-complexity Jacobian RBF DFE TEQ using decision feedback provides approximately 35% computational complexity reduction at an SNR of 4dB.

The reduced-complexity RBF DFE TEQ implementation can be used instead of the RBF DFE TEQ in order to provide substantial computational reductions without degrading the BER performance. Since the reliability of the symbols in the decoded burst is provided by the channel decoder in the previous iteration, we were capable of designing a system, where the percentage of bits not equalised in the decoded burst was set according to our design criteria for every iteration, such that each burst exhibited a predetermined fixed computational complexity reduction for the sake of practical, constant-complexity implementations.

5.7 Conclusions

In conclusion, in this chapter the Jacobian RBF DFE TEQ has been proposed and analysed comparatively in conjunction with the well-known Log-MAP TEQ [13, 105]. The associated performances and complexities have been compared in the context of BPSK, 4-QAM and 16-QAM. The computational complexity of the Jacobian RBF DFE TEQ is dependent on the number of RBF centres, the CIR length and modulation mode. The associated 'per iteration' implementational complexity of the Jacobian RBF DFE TEQ ($m = 3$, $n = 2$,

[1]The three-tap Rayleigh fading channel has a better BER performance than the three-path Gaussian channel, because the dispersive Gaussian channel has a bad spectral characteristic exhibiting spectral null. By contrast, for the Rayleigh fading channel, the CIR taps are faded and hence the frequency-domain transfer function does not exhibit a permanent null.

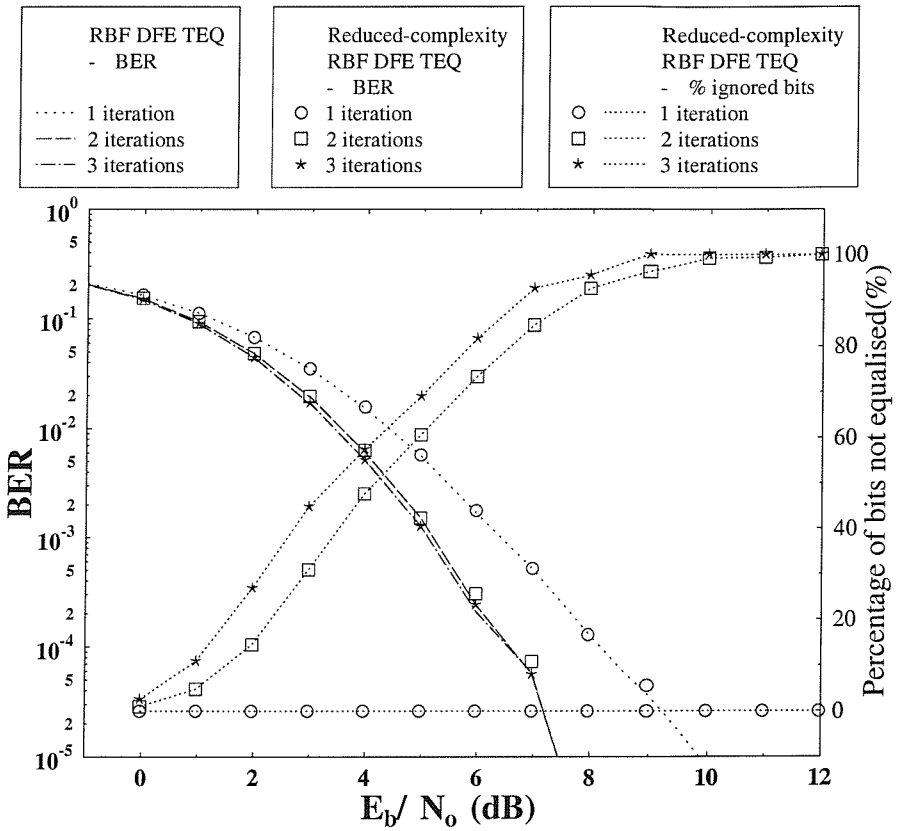


Figure 5.14: BER performance and the percentage of symbols not requiring equalisation by the reduced-complexity RBF TEQ using **detected** decision feedback over the three-tap equal-gain Rayleigh fading channel for BPSK. The LLR magnitude threshold, the log-likelihood gradient and the log-likelihood interception were set to $|\mathcal{L}|_{threshold} = 10$, $g = 1.2$ and $c = -7.5$, respectively.

$\tau = 2$) was approximately a factor 2.5, 4.4 and 16.3 lower in the context of BPSK, 4-QAM and 16-QAM, respectively, for the three-path channel considered. The performance degradation compared to the conventional Log-MAP TEQ [105] was negligible for BPSK and 4-QAM, but was approximately 10dB for 16-QAM over the three-path, equal-weight, symbol-spaced burst-invariant Rayleigh fading channel environment considered. The large performance degradation for the 16-QAM scheme is due to the error propagation effect of the DFE, which becomes more grave in conjunction with higher order constellations. Therefore, the Jacobian RBF DFE TEQ could only provide a practical performance versus complexity advantage over the conventional Log-MAP TEQ [105] for lower modulation modes. Our proposed reduced-complexity Jacobian RBF DFE TEQ was shown to provide an equivalent BER performance to that of the RBF DFE TEQ at a reduced computational load. The reduced-complexity Jacobian RBF DFE TEQ using detected decision feedback provided approximately 21% (at SNR of 6dB) and 35% (at SNR of 4dB) computational reduction for dispersive Gaussian and Rayleigh channels, respectively.

Chapter 6

Summary and Conclusions

This thesis investigated the application of neural networks in the context of channel equalisation. As an introduction, the family of established neural network based equaliser structures was reviewed. We opted for studying RBF network based equalisers in detail and investigated their implementation in conjunction with adaptive modulation and turbo channel coding, in order to improve the performance of the transceivers investigated. Below, the main findings of our investigations and suggestions for further research are presented.

6.1 Summary

Chapter 2 provided a brief overview of neural networks and described, how equalisation can be viewed as a classification problem. We studied the performance of the RBF equaliser assisted QAM schemes and their adaptive convergence performance in conjunction with both clustering algorithms and LMS channel estimators. The RBF equaliser provided superior performance compared to the linear MSE equaliser using an equivalent equaliser order at the expense of a higher computational complexity, as it was shown in Figure 2.28 and 2.29. According to Figure 2.28 and 2.29, the RBF equaliser ($m = 9$) provided performance improvements of 10dB and 20dB over the linear MSE equaliser over two-path and three-path Gaussian channels, respectively, at a BER of 10^{-3}. We note that both the linear MSE equaliser and the RBF equaliser exhibited residual BER characteristics, if the channel states corresponding to different transmitted symbols are inseparable in the channel observation space, as it was shown in Figure 2.30. The adaptive performance of the RBF equaliser employing the LMS channel estimator of Section 2.9.4, the vector centre clustering algorithm of Section 2.9.5 and the scalar centre clustering algorithm of Section 2.10 was compared. The convergence rate of the clustering algorithm depends on the number of channel coefficients to be adapted and therefore also on the modulation scheme used and the CIR length. However, the convergence of the LMS channel estimation technique only depends on the CIR length

and therefore this technique is preferred for high-order modulation schemes and high CIR lengths. In Section 2.11 decision feedback was introduced into the RBF equaliser, in order to reduce its computational complexity and to improve its performance, since due to its employment the Euclidean distance between the channel states corresponding to different transmitted symbols was increased. The performance degradation due to decision error propagation increase as the BER increased, which became more significant for higher-order QAM constellations, as it was shown in Figure 2.42. The performance degradation for higher-order modulation schemes was higher for fading channel conditions, since they are more sensitive to fades due to the reduced Euclidean distance between the neighbouring channel states. We note that even for relatively slow fading channels, the channel states value can change significantly on a symbol-by-symbol basis in a transmission burst duration. Inseparable channel state clusters were observed for symbol-invariant fading, as it was shown in Figure 2.48(b), which is due to the fading effects manifesting themselves across the burst duration. These phenomena, together with the non-ideal learnt channel states, explain the residual BERs present in our simulations.

Chapter 3 introduces the concept of adaptive modulation invoked, in order to improve the throughput of the system, while maintaining a certain target BER performance. The RBF DFE's 'on-line' BER estimation of the received data burst was used as the AQAM modem mode switching metric in order to quantify the channel's quality. Our simulation results of Section 3.3.5 showed that the proposed RBF DFE-assisted BbB adaptive modem outperformed the individual constituent fixed modulation modes in terms of the mean BER and BPS. The AQAM scheme employing RBF DFE was compared to the AQAM scheme using a conventional DFE, in terms of mitigating the effects of the dispersive wideband channel. Our results in Section 3.3.5 showed that the AQAM RBF DFE scheme was capable of performing as well as the conventional AQAM DFE at a lower decision delay and lower feedforward as well as feedback order. The performance of the AQAM RBF DFE can be improved by increasing both the decision delay τ and the feedforward order m, at the expense of increased computational complexity, while the performance of the conventional AQAM DFE cannot be improved significantly by increasing its equaliser order. However, the computational complexity of the RBF DFE is dependent on the AQAM mode and increases significantly for higher-order modulation modes. This is not so in the context of the conventional DFE, where the computational complexity is only dependent on the feedforward and feedback order. A practical method of obtaining the switching BER thresholds of the joint AQAM RBF DFE scheme was proposed in Section 3.4, which was shown to provide a near-identical performance in comparison to the achievable best-case performance for the target BER of 10^{-2}. However, for the lower target BER of 10^{-4}, the BER performance degradation in comparison to the best-case performance was more significant, since the

RBF DFE was unable to provide a BER estimate of such high accuracy and also because of the spread nature of the BER estimates seen for example in Figure 3.14. Overall, we have shown that our proposed AQAM scheme improved the throughput performance compared to the fixed modulation modes. On the whole, the RBF DFE provides a reliable channel quality measure, which quantifies all channel impairments, irrespective of their source for the AQAM scheme and at the same time improves the BER performance.

Chapter 4 proposed the Jacobian RBF equaliser that invoked the Jacobian logarithmic approximation, in order to reduce the computational complexity of the original RBF equaliser discussed in Section 2.9.1, while providing a similar BER performance. For example, the total complexity reduction was by a factor of about 2.1, when we considered a 16-QAM RBF DFE in conjunction with the equaliser parameters of $m = 3$, $n = 1$ and $\tau = 2$. The performance of the RBF DFE was investigated using turbo coding and it was compared to the turbo-coded conventional DFE scheme in Section 4.4. Introducing BCH(31,26) turbo coding into the system improved the SNR-performance by 9.5dB for BPSK and by about 8dB for 4-QAM, 16-QAM and 64-QAM at a BER of 10^{-4}. The performance of the conventional DFE and RBF DFE schemes depends on their uncoded performance. We have also investigated the application of turbo BCH coding in conjunction with AQAM in a wideband fading channel. We observed in Section 4.6.2 that the performance of the switching mechanism depends on the fluctuation of the switching metric since the AQAM switching regime assumed that the channel quality was slowly varying. This was demonstrated in Section 4.6.2, when we compared the performance of the AQAM scheme using the short-term BER before and after turbo decoding as the switching metric. The spurious nature of the short-term BER after turbo decoding was shown in Figure 4.22, which degraded the performance of the AQAM scheme, as it assumed that the channel quality was slowly varying. The turbo-coded AQAM RBF DFE system exhibited a better BPS performance, when compared to the uncoded system at low to medium channel SNRs – in the range of 0dB to 26 dB – as evidenced by Figure 4.27. The same figure also showed an improved coded BER performance at higher channel SNRs – in the range above 30dB. A virtually error-free turbo-coded AQAM scheme was also characterized in Figure 4.28. The BPS performance of the error-free coded system was better, than that of the uncoded AQAM system for the channel SNR range of 0dB to 15dB, as evidenced by Figure 4.28. Overall, we have presented the advantageous interactions of RBF-aided DFE and BbB AQAM in conjunction with turbo FEC.

Chapter 5 presented the Jacobian RBF DFE TEQ and comparatively analysed its associated performance and complexity with the well-known Log-MAP TEQ [105] in the context of BPSK, 4-QAM and 16-QAM. The computational complexity of the Jacobian RBF DFE TEQ was shown in Section 5.4 to be dependent on the number of RBF centres, on the

CIR length and on the modulation mode. The associated 'per iteration' implementational complexity of the Jacobian RBF DFE TEQ ($m = 3$, $n = 2$, $\tau = 2$) was approximately a factor 2.5, 4.4 and 16.3 lower in the context of BPSK, 4-QAM and 16-QAM, respectively, for the three-path channel considered as seen in Table 5.1. The associated performance degradation compared to the Log-MAP TEQ was shown in Figures 5.8, 5.9 and 5.10 to be approximately 0.2dB, 0.2dB and 10dB for BPSK, 4-QAM and 16-QAM, respectively over the three-path, equal-weight, symbol-spaced Rayleigh fading channel environment considered. The large performance degradation for the 16-QAM scheme was due to the error propagation effect of the DFE, which became more grave in conjunction with higher-order constellations. Therefore, the Jacobian RBF DFE TEQ of Section 5.2 could only provide a practical performance versus complexity advantage for lower modulation modes. In terms of storage requirements, the Jacobian RBF DFE is less demanding, as it only has to store the values of the RBF centres, while the Log-MAP equaliser has to store both the forward- and backward-recursively calculated metrics. Our proposed reduced-complexity RBF DFE TEQ – where the RBF DFE skips evaluating the symbol LLRs in the current iteration when the symbol is sufficiently reliable after channel decoding in the previous iteration – was shown in Section 5.6 to give significant computational complexity reductions, while providing an equivalent BER performance to the RBF DFE TEQ. The complexity reduction was approximately 21% (at an SNR of 6dB) and 35% (at an SNR of 4dB) for dispersive Gaussian and Rayleigh channels, respectively.

6.2 Suggestions for Future Research

In most of our work initially we have made some idealistic assumptions for our AQAM scheme, such as those in Section 3.3.3. Further work has to explore and quantify the performance of our proposed systems in practical scenarios in the presence of co-channel interference (CCI), CIR estimation errors, channel quality estimation latency, etc. Research has also been conducted in employing neural network based equalisers in code-division multiple-access (CDMA) environments [112, 113, 114, 115, 116, 117, 118, 119] and it appears promising to explore further this work in conjunction with our coded AQAM schemes.

Space-time processing techniques [120] are powerful in enhancing the capability of mobile communication services and are currently regarded by many within the wireless communications community as a core system component in future generation of mobile networks. Space-time processing techniques optimize the cellular spectral efficiency of the network by implementing more than one antenna element in order to optimally transmit and receive signals to or from users using both temporal and spatial signal processing techniques in the transceiver. Implementing coding in conjunction with space-time techniques, Tarokh *et. al.*

[121, 122, 123, 124, 125] presented a range of new advanced modem schemes suitable for high-data-rate wireless communications based on space-time coded modulation (STCM). Simulation results for the proposed STCM-based modem show great promise as a powerful channel coding method for high-data-rate wireless applications. Hence it appears promising to implement this technique in conjunction with our proposed RBF equalisers and to investigate its performance over wideband channels subjected to CCI.

Chen *et. al.* proposed a strategy for designing DFE-based support vector machines (SVM) [73]. The SVM design in conjunction with low-complexity conventional DFE structures achieves asymptotically the minimum BER (MBER) solution [70], which provides a performance close to the optimal Bayesian DFE. Unlike the exact MBER solution, the SVM solution can be computed significantly more efficiently. The low-complexity structure of the SVM DFE provides an attractive alternative to the optimal Bayesian-based RBF DFE and to its implementation in conjunction with AQAM schemes and turbo coding.

Appendix A

Least Mean Square Algorithm

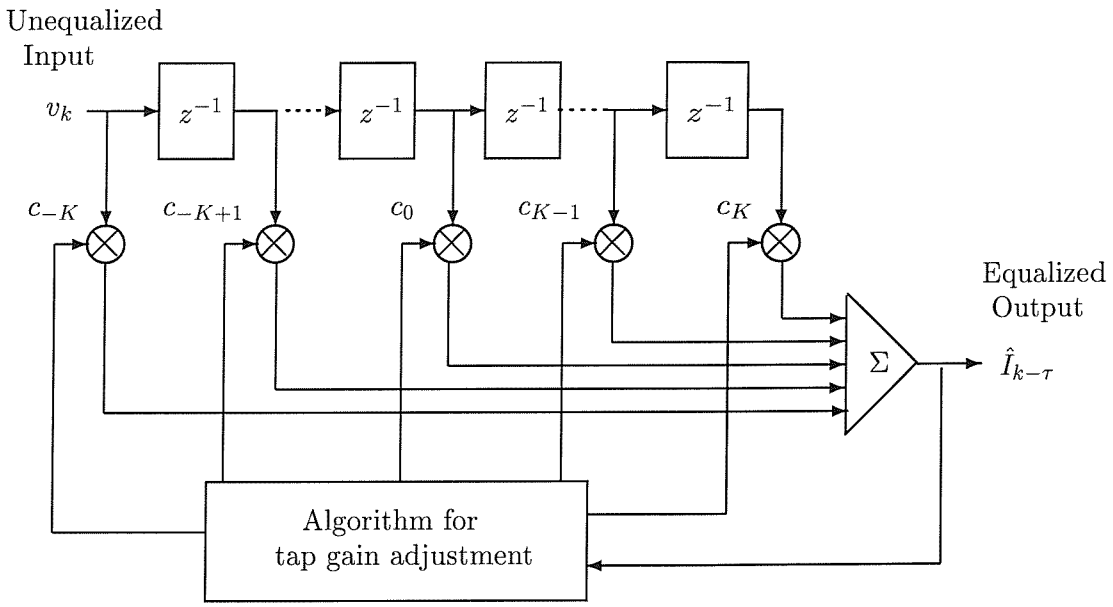


Figure A.1: Linear equaliser schematic

For a linear transversal filter shown in Figure A.1, the optimum coefficients according to the mean square error (MSE) criterion [59], are determined from the solution of a set of linear equations, which can be expressed in matrix form as:

$$\mathbf{\Gamma} \boldsymbol{C} = \boldsymbol{\xi}, \tag{A.1}$$

where $\mathbf{\Gamma}$ is the $(2K+1) \times (2K+1)$ covariance matrix of the input signal samples $\{v_k\}$, $\boldsymbol{C}$ is the column vector of $(2K+1)$ equaliser tap weights $\{c_k\}$ and $\boldsymbol{\xi}$ is a set of $(2K+1)$ cross-correlations between the unequalised input samples v_k and the equalised desired response $\{I_k\}$.

In order to avoid the direct matrix inversion in obtaining $\boldsymbol{C}_{\text{opt}}$, we can minimize the MSE J by iteratively descending on the associated MSE versus the equaliser coefficient surface

via gradient methods [126, 59]. Each equaliser tap weight is changed in the direction opposite to its corresponding gradient component $\delta J/\delta c_k, k = -K, \ldots, -1, 0, -1, \ldots, K$ at the currently encountered point of the MSE surface. The iteratively updated values of the coefficient vector $\boldsymbol{C}$ are given by [59]:

$$\boldsymbol{C}_{k+1} = \boldsymbol{\Gamma}\boldsymbol{C}_k - \boldsymbol{\xi} = \boldsymbol{C}_k + \mu E(\varepsilon_k \boldsymbol{V}_k^*), \tag{A.2}$$

where the vector $\boldsymbol{C}_k$ is the set of equaliser coefficients at the kth iteration, $\varepsilon_k = I_k - \hat{I}_k$ is the equalisation error at kth iteration, $\boldsymbol{V}_k$ is the vector of the equaliser input signal samples that generate the equaliser output $\hat{I}_k$, i.e., $\boldsymbol{V}_k = [v_{k+K} \ldots v_k \ldots v_{k-K}]^T$ and μ is the associated step-size. The difficulty with the gradient descent method is in determining the covariance matrix $\boldsymbol{\Gamma}$ and the vector $\boldsymbol{\xi}$ of cross correlations, which will need a collection of unequalised data $\{v_k\}$. An alternative is to estimate the MSE surface gradient and adjust the tap weights according to the relation [126, 59]:

$$\boldsymbol{C}_{k+1} = \boldsymbol{C}_k + \mu\varepsilon\boldsymbol{V}_k^*. \tag{A.3}$$

This is the LMS algorithm, which is implementationally simple. In order to guarantee convergence of the recursive relation in Equation A.3, the step-size μ must satisfy the inequality [59]:

$$0 < \mu < 2/\lambda_{\max}, \tag{A.4}$$

where $\lambda_{\max}$ is the largest so-called eigenvalue of $\boldsymbol{\Gamma}$ [59]. Note that $\lambda_{\max}$ cannot be greater than the trace of $\boldsymbol{\Gamma}$, tr[$\boldsymbol{\Gamma}$], which can be expressed as [59] $(2K+1)E(v_k^2)$ for a linear transversal filter. Thus convergence of the coefficient vector is assured by [126]

$$\begin{aligned} \text{In general:} \quad & 0 < \mu < \frac{2}{tr[\boldsymbol{\Gamma}]} \\ \text{Transversal filter:} \quad & 0 < \mu < \frac{2}{(2K+1)(\text{received signal power})}. \end{aligned} \tag{A.5}$$

In practical applications, the LMS algorithm employs noisy estimates of the MSE surface gradient. The noise in these estimates causes the coefficients to fluctuate randomly around the optimal values. The final MSE in steady state is $J_{\min} + J_{\text{excess}}$. The *excess mean square error* term J_{excess} is defined in a simplified form by Proakis [59] and Widrow[126] as:

$$J_{\text{excess}} \approx \frac{1}{2}\mu J_{\min} tr[\boldsymbol{\Gamma}]. \tag{A.6}$$

From Equation A.6, we can see that the value of μ has to be as small as possible, in order to reduce the excess MSE. However, at the same time the step size μ is proportional to the speed of convergence. A fast convergence is important, if the statistical time variations of the signal occur rapidly. Therefore a compromise is necessary for ensuring good tracking of

the time-variant signal statistics without undue degradation of the associated performance. To overcome this problem, in the LMS algorithm the step-size is often made time-varying. A few of the time-varying forms found in the literature are:

$$\text{Stochastic approximation schedule [127]}: \quad \mu(k) = \frac{a}{k} \tag{A.7}$$

$$\text{Search-then-converge schedule [128]}: \quad \mu(k) = \frac{\mu_0}{1 + (k/\kappa)} \tag{A.8}$$

where a, μ_0 and κ are constant.

Appendix B

Minimal Feedforward Order of RBF Equaliser with Decision Feedback [Proof] [26]

The RBF DFE has a feedforward order of m, feedback order of n and a decision delay of τ. We denote the $(m+L)$-symbol length channel input sequence that determines the values of the noiseless channel state $\mathbf{r}_j, j = 1, \ldots, n_s$ by $\tilde{\mathbf{I}}_{k-\tau}$, where the CIR length is $L+1$. Let $\tilde{\mathbf{I}}_{k-\tau} = \mathbf{s}_j, j = 1, \ldots, n_s$, where $\mathbf{s}_j$ represents the n_s possible states of $\tilde{\mathbf{I}}_{k-\tau}$. Referring to Equation 2.106, we consider $\mathbf{r}^i_{j,l} \in V^i_{m,\tau,j}, j = 1, \ldots, n_f$ for $I_{k-\tau} = \mathcal{I}_i, i = 1, \ldots, \mathcal{M}$, where

$$\mathbf{r}^i_{j,l} = \begin{bmatrix} r^i_{j,l,0} & \ldots & r^i_{j,l,m-1} \end{bmatrix}^T, \qquad 1 \le i \le \mathcal{M}, \qquad 1 \le j \le n_s, \qquad 1 \le l \le n_f. \tag{B.1}$$

Assuming $m > \tau + 1$, the squared distance between the channel output vector $\mathbf{v}_k$ of Equation 2.2 and $\mathbf{r}^i_{j,l}$ is

$$\begin{aligned} \omega^i_{m,j,l}(k) &= \|\mathbf{v}_k - \mathbf{r}^i_{j,l}\|^2 \\ &= \sum_{u=0}^{\tau} (v_{k-u} - r^i_{j,l,u})^2 + \sum_{u=\tau+1}^{m-1} (v_{k-u} - r^i_{j,l,u})^2 \\ &= \omega^i_{\tau+1,j,l}(k) + \sum_{u=\tau+1}^{m-1} (v_{k-u} - r^i_{j,l,u})^2, \\ &\qquad 1 \le i \le \mathcal{M}. \end{aligned} \tag{B.2}$$

The feedback symbols are assumed to be correct, that is,

$$\tilde{\mathbf{I}}_{k-\tau} = \begin{bmatrix} I_{k-\tau-1} & \ldots & I_{k-\tau-n} \end{bmatrix}^T, \tag{B.3}$$

where $n = L + m - 1 - \tau$. For any $\mathbf{r}^i_{j,l} \in V^i_{m,\tau,j}$ and $1 \le i \le \mathcal{M}$, we have:

$$r^i_{j,l,u} = \sum_{n=0}^{L} f_n I_{k-u-n} \qquad \tau + 1 \le u \le m - 1. \tag{B.4}$$

Upon introducing

$$\tilde{\omega}(k) = \sum_{u=\tau+1}^{m-1} (v_{k-u} - r^i_{j,l,u})^2. \tag{B.5}$$

We have:

$$\omega^i_{m,j,l}(k) = \omega^i_{\tau+1,j,l}(k) + \tilde{\omega}(k). \tag{B.6}$$

The conditional Bayesian decision variables, given that $\tilde{\mathbf{I}}_{k-\tau} = \mathbf{s}_{f,j}$, are as follows:

$$\zeta_i(k|\tilde{\mathbf{I}}_{k-\tau} = \mathbf{s}_{f,j}) = \sum_{l=1}^{n^i_{s,j}} \alpha \cdot exp(-\omega^i_{m,j,l}/\rho), \qquad 1 \leq i \leq \mathcal{M}, \tag{B.7}$$

where α is an arbitrary positive scalar, $\rho = 2\sigma^2_\eta$ and $n^i_{s,j}$ is the number of states in $V^i_{m,\tau,j}$. Substituting Equation B.6 into Equation B.7 yields:

$$\begin{aligned} \zeta_i(k|\tilde{\mathbf{I}}_{k-\tau} = \mathbf{s}_{f,j}) &= \sum_{l=1}^{n^i_{s,j}} \alpha \cdot exp\left(-\tilde{\omega}(k)/\rho\right) exp\left(-\omega^i_{\tau+1,j,l}(k)/\rho\right) \\ &= \sum_{l=1}^{\tilde{n}^i_{s,j}} \tilde{\alpha} \cdot exp(-\omega^i_{\tau+1,j,l}(k)/\rho) \qquad 1 \leq i \leq \mathcal{M}, \end{aligned} \tag{B.8}$$

where $\tilde{n}^i_{s,j}$ is the number of states in $V^i_{\tau+1,\tau,j}$ and $\tilde{\alpha}$ is a positive scalar, since α and $\tilde{\omega}(k)$ are positive scalars. This proves that the RBF DFE based on the Bayesian solution [26] having a feedforward order of $m = \tau + 1$ has the same conditional decision variables, as those of arbitrary higher feedforward orders of $m > \tau + 1$.

In the above proof, the number of states in $V^i_{\tau+1,\tau,j}$, $\tilde{n}^i_{s,j}$ has first implicitly been multiplied by a factor of $\mathcal{M}^{m-\tau-1}$ so as to match the number of states in $V^i_{m,\tau,j}$, n_s, and then reduced to the original $\tilde{n}^i_{s,j}$. This is allowed, since $\tilde{\alpha}$ is an arbitrary positive scalar.

List of Symbols

b_j	bias of the jth neuron (Figure 2.4(b)).
c_k	channel encoded data bit at signalling instant k (Figure 5.2).
$\mathbf{c}_i$	the centres of the ith RBF at the hidden layer of the RBF network (Equation 2.29).
$\mathbf{c}_{i,k}$	the ith RBF centres at signalling interval k (Equation 2.94).
d_i	the ith training data or desired response.
d_n	data bit at signalling instant n (Figure 5.2).
$E(.)$	expectation operator.
$\mathcal{E}(F)$	cost function of the neural network output function F defined by Equation 2.51.
E_b/N_0	SNR per bit.
$F(\mathbf{x})$	output function of the neural network with input $\mathbf{x}$ (Figure 2.12).
$f(.)$	activation function of neuron (Figure 2.4(b)).
f_n	the nth CIR taps in Equation 2.1 and Figure 2.1.
$f_{Bayes}(\cdot)$	optimal Bayesian decision function (defined by Equation 2.17).
$f_c(\cdot)$	the correction function of the Jacobian logarithmic relationship in Equation 4.1.
$f_p(\cdot)$	polynomial function of the PP network in Equation 2.25.
$f_{PP}(\cdot)$	overall response of the PP network equaliser (defined by Equation 2.26).

$f_{RBF}(\cdot)$	overall response of the RBF network equaliser (defined by Equation 2.81).
$G(\mathbf{x};\mathbf{x}_i)$	the Green function centred at $\mathbf{x}_i$ with input $\mathbf{x}$ (2.54).
I_k	discrete transmitted symbol at signalling instance k (Figure 2.1, Equation 2.2).
$\tilde{I}_k$	detected symbol at signalling instance k (Figure 2.2).
$\mathcal{I}_i$	the ith symbol constellation point of $\mathcal{M}$-ary modulation scheme where $i = 1, 2, \ldots, \mathcal{M}$.
$J(\lambda_1, \lambda_2)$	the Jacobian logarithmic relationship between the values λ_1 and λ_2), as defined by Equation 4.1.
l	polynomial degree of the PP network (Equation 2.25).
L	memory length of the CIR defined in Figure 2.1 and Equation 2.1.
$\mathcal{L}^D$	the LLR values output by the SISO decoder (Figure 5.1).
$\mathcal{L}_e^D(c_n)$	the channel-deinterleaved extrinsic LLR of the coded bit c_n obtained from the decoder (Figure 5.1).
$\mathcal{L}_p^D(c_n)$	the channel-deinterleaved *a posteriori* LLR values of the coded bits c_k obtained from the SISO decoder (Figure 5.1).
$\mathcal{L}^E$	the LLR values output by the SISO equaliser (Figure 5.1).
$\mathcal{L}_a^E(c_k)$	the channel-interleaved *a priori* LLR information of the coded bits c_k provided to the SISO equaliser (Figure 5.1).
$\mathcal{L}_i^E(c_k)$	the combined channel and extrinsic LLR information obtained form the SISO equaliser (Figure 5.1).
$\mathcal{L}_p^E(c_k)$	the channel-interleaved *a posteriori* LLR information of the coded bits c_k obtained from the SISO equaliser (Figure 5.1).
$\mathcal{L}_{\text{average}}$	average burst LLR magnitude defined by Equation 4.15.
$\mathcal{L}_i^{\mathcal{M}}$	the ith switching LLR magnitude threshold corresponding to the $\mathcal{M}$-QAM mode (Equation 4.16).
L_b	number of data bits per transmitted burst.

L_D	number of data symbols per transmitted burst.
L_T	number of training symbols per transmitted burst.
$L(u_k = \pm 1)$	log-likelihood of the data bit u_k having the value ± 1 (Equation 4.8).
$\mathcal{L}(u_k)$	LLR value of data bit u_k defined by Equation 4.8.
m	number of equaliser feedforward taps or equaliser feedforward order (Figure 2.2).
M	number of linearly independent basis functions in the RBF network described by Figure 2.12 and Equation 2.31.
$\mathcal{M}$	number of symbol constellation points in a multilevel modulation scheme.
$\text{MSE}(\mathbf{P}, \mathbf{C})$	MSE cost function of for the K-means clustering algorithm, defined in Equation 2.78, that partitions the input pattern $\mathbf{x}$ into partition $\mathbf{P}$ and finds a reference vector $\mathbf{C}$ for the partitioned input pattern.
$\text{MSE}(\mathbf{c}, k)$	MSE of RBF vector centres at signalling interval k defined by Equation 2.125.
$M_i(\mathbf{x})$	membership indicator that specifies, whether the input pattern $\mathbf{x}$ belongs to region $\mathbb{P}_i$ and also whether the ith neuron is active (Equation 2.79).
n	feedback order of decision feedback equaliser.
$n_f = \mathcal{M}^n$	number of feeback states of the RBF network equaliser with decision feedback.
$n_s = \mathcal{M}^{L+m}$	number of desired channel states or noise-free channel output values.
$n_s^{\pm}$	number of desired channel states corresponding to the transmitted symbol $I_k = \pm 1$.
$n_{s,j} = \mathcal{M}^{L+m-n}$	number of desired channel states corresponding to the feedback state $\mathbf{s}_{f,j}$.
$n_{s,f} = \mathcal{M}^{L+1}$	number of noise-free scalar channel states.

$\mathbb{P}$	domain containing input patterns $\mathbf{x}$ (Equation 2.78).
$\mathbf{P}$	partition of input patterns (Equation 2.78).
$\mathcal{P}$	linear (pseudo) differential operator (Equation 2.53).
$P_{\text{bit, short-term}}$	the estimated short-term BER defined by Equation 3.15.
$P_i^{\mathcal{M}}$	the ith switching BER threshold corresponding to the $\mathcal{M}$-QAM mode (Equation 3.7).
$P_{error}(u_k)$	probability of error of the detected bit u_k defined by Equation 4.11.
p_i	*a priori* probability of appearance of the ith desired channel state (Equation 2.83).
$p(\cdot)$	probability density function.
$P(x)$	probability of x occurring.
$p(x\|\mathbf{s})$	conditional probability density function of x, given that the state of nature is $\mathbf{s}$.
$r_{i,k}$	the ith scalar channel states at signalling interval k (Equation 2.100).
$\mathbf{r}_i$	noise-free channel output vector defined in Figure 2.3 where $\mathbf{r}_i = \tilde{\mathbf{v}}_k$.
$\mathbf{r}_i^{\pm}$	the ith noise-free channel output state corresponding to the transmitted symbol $I_k = \pm 1$ (Equation 2.10).
$\mathbf{r}_{j,i}^{\pm}$	the ith noise-free channel output state corresponding to the transmitted symbol $I_k = \pm 1$ and feedback sequence $\mathbf{s}_{f,j}$ (Equation 2.104).
$\mathbf{s}_i$	the ith channel input sequence, $i = 1, \ldots, n_s = \mathcal{M}^{L+m}$ (Equation 2.7).
$\mathbf{s}_{f,i}$	the ith channel feedback sequence, $i = 1, \ldots, n_{s,j} = \mathcal{M}^n$ (Equation 2.103).
T	symbol duration.
$\text{tr}\{\cdot\}$	trace of a matrix.

$v_j^{(m)}$	output of the jth neuron at the mth layer of the MLP (Figure 2.9, Equation 2.24).
v_k	noise-contaminated received symbol at the output of the channel, which is input to the receiver (defined by Figure 2.1 and Equation 2.2).
$\tilde{v}_k$	noise-free channel output defined in Equation 2.3. (We have $v_k = \tilde{v}_k$, when v_k is uncontaminated by noise.)
$\mathbf{v}_k$	equaliser input vector constituted by m v_k samples, as defined by Figure 2.1 and Equation 2.2.
$\tilde{\mathbf{v}}_k$	equaliser noise-free input vector constituted by m $\tilde{v}_k$ sample (Equation 2.11).
$V_{m-\tau}^{\pm}$	subset of channel states that corresponds to the transmitted symbol $I_k = \pm 1$ for the RBF equaliser having an equaliser order of m and delay τ (defined by Equation 2.8).
$V_{m,\tau,j}^{\pm}$	subset of channel states that corresponds to the transmitted symbol $I_k = \pm 1$ and feedback state $\mathbf{s}_{f,j}$ for the RBF equaliser having an equaliser order of m and delay τ (defined by Equation 2.103).
w_{ij}	the ith weight of the jth neuron (Figure 2.4(b)).
w_i	the ith weight of the RBF network.
x_i	the ith input of the neural network at the input layer (Figure 2.6).
y_j	the output of the jth neuron (Figure 2.4(b)).
$y_{i,k}$	the monomials corresponding to the inputs v_{k-i_1} to $v_{k-i_1} \ldots v_{k-i_l}$ of the PP network having a polynomial degree of l at signalling interval k (Equation 2.25).
$\alpha_k(s)$	the forward oriented transition probability (Figure 5.3, Equation 5.11).
$\beta_k(s)$	the backward oriented transition probability (Figure 5.3, Equation 5.12).

$\gamma_k(s', s)$	the trellis transitions probability between the trellis stages $(k-1)$ and k (Figure 5.3, Equation 5.13).
$A_k(s)$	the forward oriented transition probability in logarithmic form (defined by Equation 5.18).
$B_k(s)$	the backward oriented transition probability in logarithmic form (defined by Equation 5.19).
$\Gamma_k(s', s)$	the trellis transitions probability between the trellis stages $(k-1)$ and k in logarithmic form (defined by Equation 5.20).
ε_k	error between the actual channel output v_k and the estimated channel output at signalling interval k (defined by Equation 2.89).
τ	equaliser delay (Figure 2.2).
η_k	white Gaussian noise sequence with zero mean and variance σ_η^2 (Figure 2.1, Equation 2.2).
γ_{DFE}	pseudo SNR output of the conventional DFE (defined by Equation 3.5).
ρ	width parameter of the RBF equaliser (Equation 2.81).
σ^2	variance of the Gaussian function.
σ_η^2	variance of the additive white Gausian noise.
σ_N^2	variance of the Gaussian noise at the output of the equaliser after equalisation.
σ_I^2	variance of the information symbols.
$\sigma_{\tilde{v}}^2$	variance of the noise-free received signal.
θ_j	firing threshold of the jth neuron, where $\theta_j = -b_j$ (Figure 2.4(b)).
μ	step-size or learning rate of the adaptive K-means clustering algorithm defined by Equation 2.79.
μ_c	learning rate for the vector centres (Equation 2.94).
μ_f	step-size for the CIR estimator (Equation 2.90).
μ_r	learning rate for the scalar centres (Equation 2.100).

μ_w	learning rate for the RBF weights (Equation 2.98).
λ	regularization parameter in Equation 2.51.
$\varphi_i(\cdot)$	the ith radial basis function at the hidden layer of the RBF network (Figure 2.12, Equation 2.29).
$\mathbf{\Phi}$	interpolation matrix (Equation 2.47).

Glossary

3G	Third Generation
ANN	Artifial Neural Network
AQAM	Adaptive Quadrature Amplitude Modulation
ARIB	Association of Radio Industries and Businesses
AWGN	Additive White Gaussian Noise
BbB	Burst-by-Burst
BER	Bit Error Rate, the proportion of the bits received incorrectly
BPS	Bits Per Symbol, the proportion of the bits per symbol
BPSK	Binary Phase Shift Keying
CCI	Co-Channel Interference
CDMA	Code Division Multiple Access
CIR	Channel Impulse Response
DFE	Decision Feedback Equalizer
ETSI	European Telecommunications Standards Institute
FDD	Frequency Division Duplex
FEC	Forward Error Correction
FIR	Finite Impulse Response
FL	Functional Link
FPLMTS	Future Public Land Mobile Telecommunications Systems

GSM	Global System for Mobile Communications, A Pan-European digital mobile radio standard operating at 900MHz.
HT	Hilly Terrain, channel impulse response of a hilly terrain environment
I	In-phase component of QAM mode
I/Q EQ	In-Phase/Quadrature-phase Equaliser
IMT-2000	International Mobile Telecommunications in the year 2000
ISI	Intersymbol Interference
ITU	International Telecommunications Union, formerly the CCITT, standardisation group
LLR	Log Likelihood Ratio
LMS	Least Mean Square, a stochastic gradient algorithm
MAP	Maximum A-Posteriori algorithm
MBER	Minimum Bit Error Rate criterion
MLP	Multilayer Perceptron
MSE	Mean Square Error
NN	Neural Networks
NO TX	No Data Transmission
OLS	Orthogonal Least Square
PDC	Personal Digital Cellular, 2G system in Japan
PHS	Personal Handyphone System, 2G system in Japan
PP	Polynomial Perceptron, a perceptron structure based on polynomial of its input
Q	Quadrature component of QAM mode
QAM	Quadrature Amplitude Modulation
RBF	Radial Basis Function

RLS	Recursive Least Square
RSC	Recursive Systematic Convolutional code
RTT	Radio Transmission Technology
SER	Symbol Error Rate, the proportion of the symbols received incorrectly
SLP	Single Layer Perceptron
SNR	Signal to Noise Ratio, noise energy compared to the signal energy
SOM	Self-Organising Map
SOVA	Soft Output Veterbi Algorithm
STCM	Space-Time Coded Modulation
SVM	Support Vector Machine
TDD	Time Division Duplex
TIA	Telecommunications Industry Association, standard organizations in United States
TU	Typical Urban, channel impulse response of an urban environment

Bibliography

[1] M. Rahnema, "Overview of the GSM system and protocol architecture," *IEEE Communications Magazine*, pp. 92–100, April 1993.

[2] J. D. Gibson, ed., *The Mobile Communications Handbook.* CRC Press, 1999.

[3] T. Ojanperä and R. Prasad, *Wideband CDMA for Third Generation Mobile Communications.* Artech House, 1998.

[4] H. Holma and A. Toskala, eds., *WCDMA for UMTS : Radio Access for Third Generation Mobile Communications.* John Wiley and Sons, Ltd., 2000.

[5] P. Chaudhury, W. Mohr, and S. Onoe, "The 3GPP proposal for IMT-2000," *IEEE Communications Magazine*, vol. 37, pp. 72–81, December 1999.

[6] S. Haykin, *Neural Networks : A Comprehensive Foundation.* Macmillan Publishing Company, 1994.

[7] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proceedings of IEEE International Conference on Communications*, pp. 1064–1070, IEEE, May 1993.

[8] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 4, pp. 570–579, July 1993.

[9] M. S. Yee, T. H. Liew, and L. Hanzo, "Block turbo coded burst-by-burst adaptive radial basis function decision feedback equaliser assisted modems," in *Proceedings of IEEE Vehicular Technology Conference*, vol. 3, (Amsterdam, Netherlands), pp. 1600–1604, September 1999.

[10] C. Douillard, M. Jézéquel, and C. Berrou, "Iterative correction of intersymbol interference: Turbo-equalization," *European Transactions on Telecommunication*, vol. 6, pp. 507–511, September/October 1995.

[11] M. S. Yee and L. Hanzo, "Upper bound performance of radial basis function decision feedback equalised burst-by-burst adaptive modulation," in *ECMCS'99*, (Krakow, Poland), 24-26 June 1999. CD-ROM.

[12] M. S. Yee and L. Hanzo, "Radial basis function decision feedback equaliser assisted burst-by-burst adaptive modulation," in *Proceedings of IEEE Globecom'99*, (Rio de Janeiro, Brazil), pp. 2183–2187, 5-9 December 1999.

[13] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operation in the log domain," in *IEEE International Conference on Communications*, vol. 2, pp. 1009–1013, IEEE, June 1995.

[14] J. Erfanian, S. Pasupathy, and G. Gulak, "Reduced complexity symbol detectors with parallel structures for ISI channels," *IEEE Transactions on Communications*, vol. 42, pp. 1661–1671, February/March/April 1994.

[15] M. S. Yee, B. L. Yeap, and L. Hanzo, "Radial basis function assisted turbo equalisation," in *Proceedings of IEEE Vehicular Technology Conference*, (Japan, Tokyo), pp. 640–644, IEEE, 15-18 May 2000.

[16] S. Siu and C. F. N. Cowan, "Performance analysis of the l_p norm back propagation algorithm for adaptive equalisation," *IEE Proceedings*, vol. 140, pp. 43–47, February 1993.

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

[18] G. J. Gibson, S. Siu, and C. F. N. Cowan, "Multi-layer perceptron structures applied to adaptive equalizers for data communications," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, (IEEE Service Center, Piscataway, NJ, USA), pp. 1183–1186, IEEE, May 1989.

[19] G. J. Gibson, S. Siu, S. Chen, P. M. Grant, and C. F. N. Cowan, "The application of nonlinear architectures to adaptive channel equalisation," in *IEEE International Conference on Communications*, vol. 2, (IEEE Service Center, Piscataway, NJ, USA), pp. 649–653, IEEE, April 1990.

[20] S. Siu, G. J. Gibson, and C. F. N. Cowan, "Decision feedback equalisation using neural network structures and performance comparison with standard architecture," *IEE Proceedings*, vol. 137, pp. 221–225, August 1990.

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

[22] C.-H. Chang, S. Siu, and C.-H. Wei, "A polynomial-perceptron based decision feedback equalizer with a robust learning algorithm," *EURASIP Signal Processing*, vol. 47, pp. 145–158, November 1995.

[23] Z.-J. Xiang and G.-G. Bi, "A new lattice polynomial perceptron and its applications to frequency-selective fading channel equalization and ACI suppression," *IEEE Transactions on Communications*, vol. 44, pp. 761–767, July 1996.

[24] Z.-J. Xiang, G.-G. Bi, and T. Le-Ngoc, "Polynomial perceptrons and their applications to fading channel equalization and co-channel interference suppression," *IEEE Transactions on Signal Processing*, vol. 42, pp. 2470–2480, September 1994.

[25] S. Chen, S. McLaughlin, and B. Mulgrew, "Complex-valued radial basis function network, Part II: Application to digital communications channel equalisation," *EURASIP Signal Processing*, vol. 36, pp. 175–188, March 1994.

[26] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive Bayesian equalizer with decision feedback," *IEEE Transactions on Signal Processing*, vol. 41, pp. 2918–2927, September 1993.

[27] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Reconstruction of binary signals using an adaptive radial basis function equaliser," *EURASIP Signal Processing*, vol. 22, pp. 77–93, January 1991.

[28] B. Mulgrew, "Applying radial basis functions," *IEEE Signal Processing Magazine*, vol. 13, pp. 50–65, March 1996.

[29] C. A. Micchelli, "Interpolation of scatter data: Distance matrices and conditionally positive definite functions," *Constructive Approximation*, vol. 2, pp. 11–22, 1986.

[30] M. J. D. Powell, *Algorithms for Approximation*, ch. Radial Basis Functions for Multivariable Interpolation: a review, pp. 143–167. Oxford: Clarendon Press, 1987.

[31] H. L. V. Trees, *Detection, Estimation and Modulation Theory, Part 1.* New York: John Wiley and Sons, 1968.

[32] E. R. Kandel, *Principles of Neural Science*, ch. Nerve cells and behavior, pp. 13–24. Elsevier, New York, 2nd. ed., 1985.

[33] C. M. Bishop, *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, ch. Learning internal representations by error propagation, pp. 318–362. Cambridge, Mass. : MIT Press, 1986.

[35] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature (London)*, vol. 323, pp. 533–536, 1986.

[36] J. B. Gomm and D. L. Yu, "Selecting radial basis function network centers with recursive orthogonal least squares training," *IEEE Transactions on Neural Networks*, vol. 11, pp. 306–314, March 2000.

[37] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, pp. 4–22, April 1987.

[38] J. C.-Sueiro, A. A.-Rodriguez, and A. R. F.-Vidal, "Recurrent radial basis function networks for optimal symbol-by-symbol equalization," *EURASIP Signal Processing*, vol. 40, pp. 53–63, October 1994.

[39] W. S. Gan, J. J. Soraghan, and T. S. Durrani, "New functional-link based equaliser," *Electronics Letter*, vol. 28, pp. 1643–1645, August 1992.

[40] A. Hussain, J. J. Soraghan, and T. S. Durrani, "A new artificial neural network based adaptive non-linear equalizer for overcoming co-channel interferenc," in *IEEE Global Telecommunications Conference*, pp. 1422–1426, November 1996.

[41] A. Hussain, J. J. Soraghan, and T. S. Durrani, "A new adaptive functional-link neural-network-based DFE for overcoming co-channel interference," *IEEE Transaction on Communications*, vol. 45, pp. 1358–1362, November 1997.

[42] T. Kohonen, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," *Proceedings of the IEEE*, vol. 84, pp. 1358–1384, October 1996.

[43] S. Bouchired, M. Ibnkahla, D. Roviras, and F. Castanié, "Neural network equalization of satellite mobile communication channels," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 1998*, vol. 6, pp. 3377–3379, May 1998.

[44] S. Bouchired, M. Ibnkahla, D. Roviras, and F. Castanié, "Equalization of satellite UMTS channels using neural network devices," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 1999*, vol. 5, (Phoenix, Arizona), pp. 2563–2566, 15-19 March 1999.

[45] S. Bouchired, D. Roviras, and F. Castanié, "Equalization of satellite mobile channels with neural network techniques," *Space Communications*, vol. 15, no. 4, pp. 209–220, 1998-1999.

[46] G. J. Gibson and C. F. N. Cowan, "On the decision regions of multilayer perceptrons," *Proceedings of the IEEE*, vol. 78, pp. 1590–1594, October 1990.

[47] N. E. Cotter, "The Stone-Weierstrass theorem and its application to neural networks," *IEEE Transactions on Neural Networks*, vol. 1, pp. 290–295, December 1990.

[48] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Transactions on Electronic Computers*, vol. 14, pp. 326–334, 1965.

[49] W. A. Light, "Some aspects of radial basis function approximation," *Approximation Theory, Spline Functions and Applications, NATO ASI Series*, vol. 256, pp. 163–190, 1992.

[50] V. A. Morozov, *Regularization Methods for Ill-Posed Problems.* Boca Raton, FL: CRC Press, 1993.

[51] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-posed Problems.* Washington, DC:W. H. Winston, 1977.

[52] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, pp. 1481–1497, 1990.

[53] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, pp. 302–309, March 1991.

[54] S. Chen, E.-S. Chng, and K. Alkadhimi, "Regularised orthogonal least squares algorithm for constructing radial basis function networks," *International Journal of Control*, vol. 64, pp. 829–837, July 1996.

[55] J. MacQueen, "Some methods for classsification and analysis of multivariate observation," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (L. M. LeCun and J. Neyman, eds.), vol. 1, (Berkeley), pp. 281–297, University of California Press, 1967.

[56] D. E. Rumelhart and D. Zipser, "Feature discovery by competitive learning," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, Cambridge, MA: Bradford Books, 1986.

[57] D. DeSieno, "Adding a conscience to competitive learning," in *Proceedings 2nd IEEE International Conference on Neural Networks*, vol. 1, pp. 117–124, July 1988.

[58] C. Chinrungrueng and C. H. Sëquin, "Optimal adaptive k-means algorithm with dynamic adjustment of learning rate," *IEEE Transactions on Neural Networks*, vol. 6, pp. 157–169, January 1995.

[59] J. G. Proakis, *Digital Communications.* McGraw-Hill, Inc., 1995.

[60] J. C.-Sueiro and A. R. F.-Vidal, "Recurrent radial basis function networks for optimal blind equalisation," in *Proceedings IEEE Workshop on Neural Networks for Signal Processing*, pp. 562–571, IEEE, June 1993.

[61] L. Hanzo, W. Webb, and T. Keller, *Single- and Multi-carrier Quadrature Amplitude Modulation: Principles and Applications for Personal Communications, WLANs and Broadcasting.* John Wiley and Sons, Ltd., 2000.

[62] S. Chen, S. McLaughlin, B. Mulgrew, and P. M. Grant, "Adaptive Bayesian decision feedback equalizer for dispersive mobile radio channels," *IEEE Transactions on Communications*, vol. 43, pp. 1937–1945, May 1995.

[63] S. N. Crozier, D. D. Falconer, and S. A. Mahmoud, "Least sum of squared errors (LSSE) channel estimation," *IEE Proceedings Part F: Radar and Signal Processing*, vol. 138, pp. 371–378, August 1991.

[64] R. Steele and L. Hanzo, eds., *Mobile Radio Communications, Second Edition, Second and Third Generation Cellular and WATM Systems.* John Wiley and Sons, Ltd, 1999.

[65] A. Urie, M. Streeton, and C. Mourot, "An advanced TDMA mobile access system for UMTS," in *1994 International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC'94*, pp. 685–690, IEEE, September 1994.

[66] J. L. Valenzuela and F. Casadevall, "Performance of adaptive Bayesian equalizers in outdoor environment," in *Proceedings of IEEE Vehicular Technology Conference*, vol. 3, pp. 2143–2147, IEEE, May 1997.

[67] E.-S. Chng, H. Yang, and W. Skarbek, "Reduced complexity implementation of Bayesian equaliser using local RBF network for channel equalisation problem," *Electronics Letters*, vol. 32, pp. 17–19, January 1996.

[68] S. K. Patra and B. Mulgrew, "Computational aspects of adaptive radial basis function equalizer design," in *IEEE International Symposium on Circuits and Systems, ISCAS'97*, vol. 1, pp. 521–524, IEEE, Piscataway, NJ, USA, June 1997.

[69] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, pp. 1261–1271, October 1996.

[70] S. Chen, B. Mulgrew, E.-S. Chng, and G. J. Gibson, "Space translation properties and the minimum-BER linear-combiner DFE," *IEE Proceedings on Communications*, vol. 145, pp. 316–322, October 1998.

[71] S. Chen, "Importance sampling simulation for evaluation the lower-bound BER of the bayesian DFE." submitted to IEEE Transactions on Communications.

[72] S. Chen, E.-S. Chng, B. Mulgrew, and G. J. Gibson, "Minimum-BER linear-combiner DFE," in *Proceedings of ICC'96*, (Dallas, Texas), pp. 1173–1177, 1996.

[73] S. Chen, S. Gunn, and C. J. Harris, "Decision feedback equalizer design using support vector machines," *IEE Proceedings Vision, Image and Signal Processing*, 2000. Accepted for publication.

[74] S. Chen and C. J. Harris, "Design of the optimal separating hyperplane for the decision feedback equalizer using support vector machine," in *Proceedings of IEEE International Conference on Accoustic, Speech and Signal Processing*, pp. 2701–2704, IEEE, June 5-9 2000.

[75] A. P. Clark and R. Harun, "Assessment of Kalman-filter channel estimators for an HF radio link," *Proceedings of IEE*, vol. 133, pp. 513–521, October 1986.

[76] P. K. Shukla and L. F. Turner, "Channel estimation-based adaptive DFE for fading multipath radio channels," *IEE Proceedings - I*, vol. 138, pp. 525–543, October 1991.

[77] B. Sklar, "Rayleigh fading channels in mobile digital communication systems, part 1: Characterization," *IEEE Communications Magazine*, vol. 35, pp. 136–146, September 1997.

[78] M. Failli, *Digital land mobile radio communications COST 207.* European Commission, 1989.

[79] T. H. Liew, C. H. Wong, and L. Hanzo, "Block turbo coded burst-by-burst adaptive modems," in *Proceedings of Microcoll'99*, (Budapest, Hungary), pp. 59–62, March 1999.

[80] J. M. Torrance, *Adaptive Full Response Digital Modulation for Wireless Communications Systems.* PhD thesis, University of Southampton, United Kingdom, 1997.

[81] W. T. Webb and R. Steele, "Variable rate QAM for mobile radio," *IEEE Transactions on Communications*, vol. 43, pp. 2223–2230, July 1995.

[82] S. Sampei, N. Morinaga, and K. Hamaguchi, "Laboratory experimental results of an adaptive modulation/TDMA/TDD for wireless multimedia communication systems," in *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 467–471, IEEE, 1997.

[83] C. H. Wong, *Wideband Adaptive Full Response Multilevel Transceivers and Equalizers.* PhD thesis, University of Southampton, United Kingdom, November 1999.

[84] C. H. Wong and L. Hanzo, "Channel capacity upperbound of a wideband burst-by-burst adaptive modem," in *Proceedings of IEEE Vehicular Technology Conference*, (Houston, USA), pp. 1851–1855, IEEE, May 1999.

[85] J. C. S. Cheung, *Adaptive Equalisers for Wideband TDMA Mobile Radio.* PhD thesis, University of Southampton, United Kingdom, 1991.

[86] W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery, *Numerical Recipes in C : The Art of Scientific Computing.* Press Syndicate of the University of Cambridge, 1995.

[87] R. A. Iltis, J. J. Shynk, and K. Giridhar, "Bayesian algorithms for blind equalisation using parallel adaptive filtering," *IEEE Transactions on Communications*, vol. 42, pp. 1017–1032, February/March/April 1994.

[88] J. M. Torrance and L. Hanzo, "Optimization of switching levels for adaptive modulation in a slow Rayleigh fading channel," *Electronic Letters*, vol. 32, pp. 1167–1169, June 1996.

[89] S. Otsuki, S. Sampei, and N. Morinaga, "Square-QAM adaptive modulation/TDMA/TDD systems using modulation estimation level with Walsh function," *Electronic Letters*, vol. 31, pp. 169–171, February 1995.

[90] J. Rapeli, "UMTS: Targets, System Concept, and Standardization in a Global Framework," *IEEE Personal Communications*, vol. 2, pp. 20–28, February 1995.

[91] E. Berruto, M. Gudmundson, R. Menolascino, W. Mohr, and M. Pizarroso, "Research activites on UMTS radio interface, network architectures, and planning," *IEEE Communications Magazine*, vol. 36, pp. 82–95, February 1998.

[92] T. Ojanperä and R. Prasad, "An overview of air interface multiple access for IMT-2000/UMTS," *IEEE Communications Magazine*, vol. 36, pp. 82–95, September 1998.

[93] J. M. Torrance and L. Hanzo, "Statistical multiplexing for mitigating latency in adaptive modems," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC'97*, pp. 938–942, September 1997.

[94] J. M. Torrance and L. Hanzo, "Latency considerations for adaptive modulation in slow Rayleigh fading," in *Proceedings of IEEE Vehicular Technology Conference*, (Phoenix), pp. 1204–1209, IEEE, 1997.

[95] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *IEEE Globecom*, pp. 1680–1686, 1989.

[96] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimising symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, pp. 284–287, March 1974.

[97] M. Breiling and L. Hanzo, "Non-iterative optimum super-trellis decoding of turbo codes," *Electronics Letters*, vol. 33, pp. 848–849, May 1997.

[98] M. Breiling and L. Hanzo, "Optimum non-iterative turbo-decoding," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, 1997*, vol. 2, pp. 714–718, IEEE, 1997.

[99] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Transactions on Telecommunications*, vol. 8, 1997.

[100] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, pp. 429–445, March 1996.

[101] A. Klein, R. Pirhonen, J. Sköld, and R. Suoranta, "FRAMES multiple access mode 1 - wideband TDMA with and without spreading," in *Proceedings of PIMRC'97*, pp. 37–41, September 1997.

[102] M. J. Gertsman and J. L. Lodge, "Symbol-by-symbol MAP demodulation of CPM and PSK signals on Rayleigh flat-fading channels," *IEEE Transactions on Communications*, vol. 45, pp. 788–799, July 1997.

[103] D. Raphaeli and Y. Zarai, "Combined turbo equalisation and turbo decoding," *IEEE Communication Letter*, vol. 2, pp. 107–109, April 1998.

[104] A. Knickenberg, B. L. Yeap, J. Hamorsky, M. Breiling, and L. Hanzo, "Non-iterative joint channel equalisation and channel decoding," in *Proceedings of Globecom'99*, (Rio de Janeiro, Brazil), pp. 442–446, December 1999.

[105] G. Bauch, H. Khorram, and J. Hagenauer, "Iterative equalization and decoding in mobile communications systems," in *Proceedings of the European Personal Mobile Communications Conference*, (Bonn, Germany), pp. 301–312, 30 September - 2 October 1997.

[106] A. Glavieux, C. Laot, and J. Labat, "Turbo equalization over a frequency selective channel," in *International Symposium on Turbo Codes*, (Brest, France), pp. 96–102, 1997.

[107] M. S. Yee and L. Hanzo, "Multi-level radial basis function network based equalisers for Rayleigh channels," in *Proceedings of IEEE Vehicular Technology Conference*, (Houston, USA), pp. 707–711, IEEE, 16-19 May 1999.

[108] K. Abend, T. J. Harley Jr, B. D. Fritchman, and C. Gumacos, "On optimum receivers for channels having memory," *IEEE Transactions Information Theory*, vol. IT-14, pp. 818–819, November 1968.

[109] B. L. Yeap, C. H. Wong, and L. Hanzo, "Iterative channel estimation and reduced complexity in-phase/quadrature-phase turbo equalisation." submitted to IEEE Global Telecommunications Conference, 27 November - 1 December 2000.

[110] B. L. Yeap, C. H. Wong, and L. Hanzo, "Reduced complexity in-phase/quadrature-phase turbo equalisation." submitted to IEEE Journal on Selected Areas in Communication, 2000.

[111] B. L. Yeap, *Turbo Equalisation Algorithm for Full and Partial Response Modulation.* PhD thesis, University of Southampton, January 2000.

[112] B. Aazhang, B.-P. Paris, and G. C. Orsak, "Neural networks for multiuser detection in code-division multiple-access communications," *IEEE Transactions on Communications*, vol. 40, pp. 1212–1222, July 1992.

[113] R. Tanner, D. G. M. Cruickshank, S. Z. W. H. Sweatman, and B. Mulgrew, "Receivers for nonlinearly separable scenarios in DS-CDMA," *Electronics Letters*, vol. 33, pp. 2103–2105, December 1997.

[114] D. G. M. Cruickshank, "Radial basis function receivers for DS-CDMA," *Electronics Letters*, vol. 32, pp. 188–190, February 1996.

[115] T. Miyajima, T. Hasegawa, and M. Haneishi, "On the multiuser detection using a neural network in code-division multiple-access communications," *IEICE Transactions on Communications*, vol. E76-B, pp. 961–968, August 1993.

[116] T. Miyajima, "Adaptive multiuser receiver using a Hopfield network," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E79-A, pp. 652–654, May 1996.

[117] T. Nagaosa, T. Miyajima, and T. Hasegawa, "Multiuser detection using a Hopfield network in asynchronous M-ary/SSMA communications," in *Proceedings of the 1996 IEEE 4th International Symposium on Spread Spectrum Techniques and Applications*, vol. 2, pp. 837–841, IEE, September 1996.

[118] T. Miyajima and T. Hasegawa, "Multiuser detection using a Hopfield network for asynchronous code-division multiple-access systems," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E79-A, pp. 1963–1971, December 1996.

[119] G. I. Kechriotis and E. S. Manolakos, "Comparison of a neural network based receiver to the optimal and multistage CDMA multiuser detectors," in *Proceedings of the 5th IEEE Workshop on Neural Networks for Signal Processing (NNSP'95)*, (Piscataway, NJ, USA), pp. 613–622, IEEE, August 1995.

[120] P. Rooyen, M. P. Lötter, and D. Wyk, *Space-Time Processing for CDMA Mobile Communications*. Kluwer Academic Publishers, February 2000.

[121] A. F. Naguib, V. Tarokh, N. Seshadri, and A. R. Calderbank, "A space-time coding modem for high-data-rate wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 1459–1478, October 1998.

[122] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block coding for wireless communications: Performance results," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 451–460, March 1999.

[123] V. Tarokh, A. Naguib, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criteria in the presence of channel estimation errors, mobility, and multile paths," *IEEE Transactions on Communications*, vol. 47, pp. 199–207, February 1999.

[124] V. Tarokh, A. Naguib, N. Seshadri, and A. R. Calderbank, "Combined array processing and space-time coding," *IEEE Transactions on Information Theory*, vol. 45, pp. 1121–1128, May 1999.

[125] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 1451–1458, October 1998.

[126] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Prentice-Hall, Inc., 1985.

[127] H. Robbins and S. Monroe, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[128] C. Darken and J. Moody, "Towards faster stochastic gradient search," in *Advances in Neural Information Processing System 4* (J. E. Moody, S. J. Hanson, and R. P. Lippmann, eds.), pp. 1009–1016, San Mateo, CA: Morgan Kaufmann, 1992.

Index

C

N

O

P

Q

R

S

Author Index

Z