Development of Fuzzy System Based Channel Equalisers

Sarat Kumar Patra

A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
August 1998

Abstract

Channel equalisers are used in digital communication receivers to mitigate the effects of inter symbol interference (ISI) and inter user interference in the form of co-channel interference (CCI) and adjacent channel interference (ACI) in the presence of additive white Gaussian noise (AWGN). An equaliser uses a large part of the computations involved in the receiver. Linear equalisers based on adaptive filtering techniques have long been used for this application. Recently, use of nonlinear signal processing techniques like artificial neural networks (ANN) and radial basis functions (RBF) have shown encouraging results in this application. This thesis presents the development of a nonlinear fuzzy system based equaliser for digital communication receivers.

The fuzzy equaliser proposed in this thesis provides a parametric implementation of symbol-by-symbol *maximum a-posteriori probability* (MAP) equaliser based on Bayes's theory. This MAP equaliser is also called Bayesian equaliser. Its decision function uses an estimate of the noise free received vectors, also called channel states or channel centres. The fuzzy equaliser developed here can be implemented with lower computational complexity than the RBF implementation of the MAP equaliser by using scalar channel states instead of channel states. It also provides schemes for performance tradeoff with complexity and schemes for subset centre selection. Simulation studies presented in this thesis suggests that the fuzzy equaliser by using only 10%-20% of the Bayesian equaliser channel states can provide near optimal performance.

Subsequently, this fuzzy equaliser is modified for CCI suppression and is termed fuzzy–CCI equaliser. The fuzzy–CCI equaliser provides a performance comparable to the MAP equaliser designed for channels corrupted with CCI. However the structure of this equaliser is similar to the MAP equaliser that treats CCI as AWGN. A decision feedback form of this equaliser which uses a subset of channel states based on the feedback state is derived. Simulation studies presented in this thesis demonstrate that the fuzzy–CCI equaliser can effectively remove CCI without much increase in computational complexity. This equaliser is also successful in removing interference from more than one CCI sources, where as the MAP equalisers treating CCI as AWGN fail. This fuzzy–CCI equaliser can be treated as a fuzzy equaliser with a preprocessor for CCI suppression, and the preprocessor can be removed under high signal to interference ratio condition.

Declaration of Originality

This thesis was composed entirely by myself. The work reported here in has been exclusively conducted by myself in the Department of Electronics and Electrical Engineering at the University of Edinburgh, Edinburgh, U.K. except where stated otherwise.

Sarat Kumar Patra

October 5, 1998

to my parents

Acknowledgements

My three years of study experience in the UK and particularly in Edinburgh, was a rewarding one and I enjoyed every bit of it. I thank all who tangibly or intangibly contributed to my family's comfortable and enjoyable stay in Edinburgh. However, I offer my sincere thanks to,

- Dr.Bernard Mulgrew, my supervisor, for his advice, guidance, encouragement and his willingness to share his deep insight during the various stages of this work and also for proof reading the thesis;
- Prof. Peter Grant for his support and guidance;
- Association of Commonwealth Universities and British Council for providing me funding to pursue this study in U.K. in the form of Commonwealth scholarship;
- the members of Signals and Systems Group, Electronics and Electrical Engineering Department, University of Edinburgh for all the help during this period; I would like to thank in particular (not in any order) Justine F., Paul S, Yoo Sok S., Jon A., John T., Rudy T., George T., Dave L. for numerous valuable discussions on different topics and for sharing their know-how of computer hardware and software; I would also like to thank John T. and Apolostos G. for their help in proof reading chapters of the thesis;
- Dr. P. K. Ahluwalia, Dr. J.K.Satpathy for reading the thesis in the final stages of preparation for errors and omissions in the draft manuscripts;
- my parents for inspiring and brothers for lending me a wonderful family;
- Sasmita for giving me cheerful company through out the course of this work and bearing with all the difficulties of bringing up our little son and running the home smilingly;
- Sanchit, for filling my evenings with happy moments after a days work; during final stages of my work, I will always remember his insistence over phone *papa jaldi a jao* (papa come back soon);
- all those who made our stay in Edinburgh an unforgettable and rewarding experience.

Contents

List of figures

List of tables

Acronyms and abbreviations

ACI	adjacent channel interference
ANN	artificial neural network
ANFF	adaptive neuro-fuzzy filter
ASK	amplitude shift keying
AWGN	additive white Gaussian noise
BER	bit error ratio
BLMS	block least mean square
BP	back propagation
CCI	co-channel interference
CDMA	code division multiple access
COG	centre of gravity
DCR	digital cellular radio
DCS	digital communication system
DFE	decision feedback equaliser
DSP	digital signal processing
FAF	fuzzy adaptive filter
FBF	fuzzy basis function
FDMA	frequency division multiple access
FEXT	far end cross talk
FIR	finite impulse response
FSE	fractionally spaced equaliser
FSK	frequency shift keying
FT	Fourier transform
GSM	global system for mobile communication
HDTV	high definition television
i.i.d.	independent identically distributed
IIR	infinite impulse response
ISI	inter symbol interference
LAN	local area network

LMS	least mean square
LS	least square
MAP	maximum *a-posteriori* probability
MLP	multi layer perceptron
MLSE	maximum likelihood sequence estimator
MMSE	minimum mean square error
MSE	mean square error
NBESS	normalised Bayesian equaliser with scalar states
NBSS–CCI	normalised Bayesian equaliser with scalar states for a CCI channel
NEXT	near end cross talk
OLS	orthogonal least squares
PAM	pulse amplitude modulation
pdf	probability density function
PSK	phase shift keying
QPSK	quadrature phase shift keying
RBF	radial basis function
RLS	recursive least squares
SIR	signal to interference ratio
SINR	signal to interference noise ratio
SNR	signal to noise ratio
TDL	tapped delay line
TDMA	time division multiple access
VLSI	very large scale integration
w.r.t.	with respect to
ZF	zero forcing

Nomenclature

a_i	channel impulse response tap weight i
$a_{0,j}$	channel impulse response tap weight j in presence of co-channel
$a_{i,j}$	co-channel impulse response tap weight j $(1 \leq i \leq L)$
B	number of interfering adjacent channels
c_{il}	$(l+1)$ component of channel state $\mathbf{c}_i$
$c_{co,il}$	$(l+1)$ component of co-channel state $\mathbf{c}_{co,i}$
$C_{co,\alpha}$	scalar co-channel state α
C_i	scalar channel state i
$\mathbf{c}_i$	channel state i
$\mathbf{c}_{co,i}$	co-channel state i
$\mathbf{C}_d$	channel state matrix
$\mathbf{C}_d^+$	positive channel state matrix
$\mathbf{C}_d^-$	negative channel state matrix
$C(z)$	z-transform for impulse response of a linear equaliser
d	equaliser decision delay
$e(k)$	error signal at time index k
$\mathcal{E}$	expected value
$f(x)$	Arbitrary function f with variable x
$\mathfrak{F}\{\mathbf{r}(k)\}$	equaliser decision function with input vector $\mathbf{r}(k)$
F_i^l	fuzzy set F corresponding to rule l and input x_i
G^l	fuzzy inference corresponding to rule l
g_i^-	minimum value of fuzzy filter input x_i
g_i^+	maximum value of fuzzy filter input x_i
$h(t)$	impulse response of a channel
$h_{aci\_i}(t)$	impulse response of an adjacent channel i
$h_{co\_i}(t)$	impulse response of a co-channel i
$h_C(t)$	inpulse response of a physical channel
$h_R(t)$	impulse response of a receiver matched filter
$h_T(t)$	impulse response of a transmitter modulation filter

$h_{TR}(t)$	combined impulse response of transmitter and receiver filters
$\mathbf{H}$	channel matrix
$H(z)$	z-transform for channel impulse response $h(t)$
$H_{aci\_i}(z)$	z-transform for ith adjacent channel impulse response $h_{aci\_i}(t)$
$H_{co\_i}(z)$	z-transform for ith co-channel impulse response $h_{co\_i}(t)$
$\mathcal{H}(f)$	Fourier transform of $h(t)$
$\mathcal{H}^{\star}(f)$	complex conjugate of $\mathcal{H}(f)$
$\mathcal{H}_C(f)$	Fourier transform of $h_C(t)$
$\mathcal{H}_R(t)$	Fourier transform of $h_R(t)$
$\mathcal{H}_T(t)$	Fourier transform of $h_T(t)$
$\mathcal{H}_{TR}(t)$	Fourier transform of $h_{TR}(t)$
i	arbitrary variable
$i1, i2, \ldots, im$	arbitrary variables
j	arbitrary variable
$j1, j2, \ldots, jm$	arbitrary variables
k	time index
l	arbitrary variable
L	number of interfering co-channels
m	equaliser order
M	number of scalar channel states
M_i	number of membership functions for input i in a fuzzy filter
M_{co}	number of scalar co-channel states used by the fuzzy equaliser
n	number of layers in a neural network(MLP)
n_c	number of channel taps
N_f	number of feedback states
n_i	number of neurons in layer i of MLP
N_c	number of fuzzy IF $\ldots$ THEN $\ldots$ rules in a fuzzy filter
N_r	number of centres in the RBF
N_s	number of channel states
$N_s^{'}$	a subset of channel states N_s
N_{sf}	number of channel states corresponding to each of feedback states
$N_{s,co}$	number of co-channel states
N_s^{+}	number of channel states corresponding $+1$ being transmitted

N_s^-	number of channel states corresponding -1 being transmitted
q	equaliser feedback order
$p(a\|b)$	state conditional pdf of a given b
$P(a)$	*a-priori* probability of a
$P(a\|b)$	*a-posteriori* probability of a having observed b
$r(k)$	received scalar sample at time index k
$\mathbf{r}(k)$	received vector at time index k
$\widehat{r}(k)$	noise free received scalar at time index k
$r_{res}(k)$	channel residual at time index k
$\widehat{r}_{co}(k)$	co-channel component of received signal $r(k)$
$\mathbb{R}$	1-dimensional space in $\mathbb{R}$
$\mathbb{R}^i$	i-dimensional space in $\mathbb{R}$
$R^{(l)}$	a fuzzy rule l
$s(k)$	transmitted sample at time index k
$s_0(k)$	transmitted sample at time index k of the desired channel
$s_{aci\_i}(k)$	transmitted sample at time index k of adjacent channel $i,\ 1 \leq i \leq B$
$s_i(k)$	transmitted sample at time index k of interfering channel $i,\ 1 \leq i \leq L$
$\mathbf{s}(k)$	transmitted signal vector at time index k
$\widehat{s}(k-d)$	estimated sample at time index k with decision delay d
T	time period of transmitted signal
w_i	weight i of a filter/ RBF centre/ neuron / equaliser
$x(k)$	scalar input at time index k
$\mathbf{x}(k)$	vector input at time index k
$y(k)$	scalar output at time index k
$\widehat{y}(k)$	desired scalar output at time index k
$\alpha,\ \alpha 1$	arbitrary constant
β	excess bandwidth factor
δ_i^j	centre j corresponding to input i for a fuzzy filter
$\eta(k)$	AWGN at time index k
$\kappa,\ \kappa_1,\ \kappa_2$	arbitrary constant
$\lambda,\ \lambda_1$	power scaling parameter for co-channel-1
μ	learning rate/ adaptation gain in clustring algorithm
ν	learning rate for cluster varience in enhanced κ-means algorithm

φ	activation function associated with a neuron
$\boldsymbol{\rho}$	RBF centres
ϱ	learning rate for weights in a fuzzy filter
σ	spread parameter
σ_c^2	channel output signal power
σ_{co}^2	co-channel signal power
σ_s^2	signal power
σ_η^2	channel noise varience
σ_r^2	RBF centre spread parameter
σ_i^j	spread parameter for fuzzy centre δ_i^j
ψ_i^j	membership function j for input i
$\boldsymbol{\Psi}$	fuzzy basis function
$\tau(f)$	channel group delay characteristics
θ	phase response of the communication channel
$\vartheta_l(k)$	weight corresponding to rule l of the fuzzy adaptive filter
ξ	threshold weight of a neuron
ω_c	channel bandwidth
ω_s	signal bandwidth
$\zeta_\alpha(k)$	square distance between $r_{res}(k)$ and $C_{co,\alpha}(k-1)$
$\bar{\zeta}_\alpha(k)$ $\left(\bar{\zeta}_{\alpha^*}(k)\right)$	cluster varience (smallest cluster varience) weighted square distance in enhanced κ-means clustring algorithm
$\lvert . \rvert$	absolute distance
$\lVert . \rVert$	Euclidean distance

Chapter 1
Introduction

1.1 Introduction

The field of digital data communications has experienced an explosive growth in recent years. The demand for this form of communication is also on the rise as additional services are being added to the existing infrastructure. The telephone networks were originally designed for voice communication but, in recent times, the advances in digital communications using ISDN, data communications with computers, fax, video conferencing etc. have pushed the use of these facilities far beyond the scope of their original intended use. Similarly, introduction of digital cellular radio (DCR) and wireless local area networks (LAN's) have stretched the limited available radio spectrum capacity to the limits it can offer. These advances in digital communications have been made possible by the effective use of the existing communication channels with aid of signal processing techniques. Nevertheless these advances on the existing infrastructure have introduced a host of new unanticipated problems.

Bandwidth efficient data communication requires the use of adaptive equalisers. This thesis deals with the development of fuzzy system based adaptive equalisers to overcome some of the channel impairments encountered in present day digital communication systems (DCS).

The chapter begins with an exposition of the principal motivation behind the work undertaken in this thesis. Following this, section 1.3 provides a brief literature survey on equalisation in general and nonlinear equalisers in particular. Section 1.4 outlines the contributions made in this thesis. At the end, section 1.5 presents the thesis layout.

1.2 Motivation for work

The revolution in digital communication techniques can be attributed to the invention of the automatic linear adaptive equaliser in the late 1960's [1]. From this modest start, adaptive equalisers have gone through many stages of development and refinement in the last 30 years.

Early equalisers were based on linear adaptive filter algorithms[2] with or without a decision feedback. Alternatively maximum likelihood sequence estimator (MLSE)[3] were implemented using the Viterbi[4, 5] algorithm. One may ask, if scientists and engineers were satisfied with these forms of equalisers for nearly two decades, what was the necessity of investigating new equaliser structures? The reason for this can be attributed to the following two main factors.

Firstly, both forms of the equalisers provided two extremities in-terms of performance achieved and the computational cost involved. The linear adaptive equalisers are simple in structure and easy to train but they suffer from poor performance in severe conditions. On the other hand, the infinite memory MLSE provide good performance but at the cost of large computational complexity.

Secondly, rapid advances in digital signal processing (DSP) techniques have provided scope for very large scale integration (VLSI) implementation. These can also be implemented with software algorithms for testing. The programming capability of DSP processors make them very attractive for complex signal processing applications. These features of DSP techniques have been successfully used in a variety of applications like signal processing, speech processing, image processing and digital communication to name a few.

Owing to the aforementioned reasons nonlinear equalisers have been investigated in the last decade resulting in a rich variety of techniques using artificial neural networks (ANN) [6, 7], radial basis function (RBF) [8, 9] and recurrent networks [10] etc. But the study of new techniques can provide adaptive equalisers which have the advantages of both good performance and low computational cost. Based on these reasons, this thesis undertakes the development of fuzzy system based equalisers. Some of the expected advantages of using fuzzy equalisers stem from the success of fuzzy systems in a variety of signal processing applications including equalisation [11] and pattern classification [12–15].

1.3 Background literature survey

Nyquist laid the foundation for digital communication over band limited analogue channels in 1928 [16], with the enunciation of telegraph transmission theory. The research in channel equalisation started much later in 1960's and was centred around the basic theory and structure of zero forcing equalisers. The LMS algorithm by Widrow and Hoff in 1960 [2] paved the

way for the development of adaptive filters used for equalisation. But it was Lucky [1] who used this algorithm in 1965 to design adaptive channel equalisers. With the popularisation of adaptive linear filters in the field of equalisation their limitations were also soon revealed. It was seen that the linear equaliser, in-spite of best training, could not provide acceptable performance for highly dispersive channels. This led to the investigation of other equalisation techniques beginning with the MLSE equaliser [3] and its Viterbi implementation[4] in 1970's. Another form of the nonlinear equaliser which appeared around the same time was the infinite impulse response (IIR) form of the linear adaptive equaliser, where the equaliser employs feedback [17] and was termed decision feedback equaliser (DFE). The adaptive equalisers for pulse amplitude modulation (PAM) systems were extended to other complex signalling systems as well [18]. Other works carried out in this field in 1970's and 1980's were the development of fast convergence and/or computational efficient algorithms like the recursive least square (RLS) algorithm, Kalman filters[19] and RLS lattice algorithm [20]. Other forms of equalisers like fractionally spaced equalisers (FSE) [21] were also developed during this period. A review of the development of equalisers till 1985 is available in [22].

The late 1980's saw the beginning of development in the field of ANN [10]. The multi layer perceptron (MLP) based symbol-by-symbol equalisers was developed in 1990 [23, 24]. This brought new forms of equalisers that were computationally more efficient than MLSE and could provide superior performance compared to the conventional equalisers with adaptive filters. Another form of nonlinear processor called the RBFs, which were first used for multidimensional functional interpolation [25], were also used for equalisation applications subsequently [26, 27]. Subsequent years saw the development of new training algorithms and equaliser structures using ANN [28, 29] and RBF [30] networks. During this time the application of these networks to the equalisation of communication systems with complex signal constellation [31, 32] was also considered. A comprehensive review of some of these works can be found in [30].

The recent advances in nonlinear equalisers are centred around the application of different signal processing techniques to equalisation. Some of these are recurrent neural networks [33, 34], recurrent RBF [35] and Mahalonobis classifiers [36]. The development of new training algorithms [37] for selecting the equaliser structures and, for setting of the equaliser parameters [38], is an active field of research. Designing low complexity networks [39] is also an area of interest. Currently use of these signal processing techniques in other digital communication applications like code division multiple access (CDMA) [40, 41] and spread spectrum [42] is

also being actively pursued.

1.4 Thesis contributions

This section outlines some of the major contributions of the study presented in this thesis. This thesis develops fuzzy system based equalisers for DCS. The fuzzy equalisers developed here can be generally classified as nonlinear equalisers suitable for radio communication applications where channel dispersions spread over a few symbols. The digital communication problem is discussed first and the need for an equaliser is established in this context. With this existing equalisation techniques are reviewed which places the work undertaken here in context.

The thesis presents a fuzzy implementation of *maximum a-posteriori probability* (MAP) equalisers based on Bayes's theory. At first a fuzzy equaliser is developed for inter symbol interference (ISI) channels. Here ISI channels are the channels where, during transmission, the symbols are affected by preceding and succeeding symbols due to the effect of ISI and are additionally corrupted with additive white Gaussian noise (AWGN). This fuzzy equaliser provides a parametric implementation of the Bayesian equaliser with advantages in terms of computational complexity. The Bayesian equaliser can also be implemented using a RBF network with scalar centres, but the use of fuzzy systems provides a flexibility of designing a wider variety of equalisers with varying computational complexities. One of the major drawbacks of the Bayesian equaliser and its RBF implementation is the computational complexity due to the large number of centres needed to implement the decision function. The fuzzy equaliser proposed here addresses this issue by providing efficient schemes for subset centre selection to provide the equaliser decision function.

Subsequently the problem of co-channel interference (CCI) is discussed. In radio communication systems the problem of CCI limits the equaliser performance when the signal to noise ratio (SNR) is larger than the signal to interference ratio (SIR). When the SNR is larger than the SIR an equaliser treating CCI as noise suffers from severe performance degradation and the performance of the equaliser is limited by the CCI. A Bayesian equaliser designed for a CCI channel has large computational complexity. Here a CCI channel is defined as a communication channel where the signal is affected by CCI due from the signal transmitted by other users using the same carrier frequency. In addition to this the signal is also corrupted due to the effects of

ISI and AWGN. The fuzzy equaliser developed for ISI channels is modified for CCI mitigation. This fuzzy–CCI equaliser is shown to provide efficient equalisation where the Bayesian equaliser treating CCI as AWGN may fail completely. The computational complexity of both these equalisers is comparable. It is also shown that the fuzzy–CCI equaliser can provide considerable performance gain when a communication channel is corrupted with interference from more than one co-channel interferers.

The advantage provided by fuzzy equalisers in terms of computational complexity and performance gain can provide efficient equaliser design for DCR applications.

1.5 Thesis outline

The rest of the thesis is organised as follows.

Chapter 2 provides the fundamental concepts of channel equalisation and discusses linear and nonlinear equalisation techniques. This chapter analyses the channel characteristics that bring out the need for an equaliser in a communication system. Subsequently an equaliser classification is presented which puts in context the work undertaken in this thesis. A short review of linear and nonlinear equalisation techniques is also undertaken.

Chapter 3 is devoted to fuzzy implementations of Bayesian equalisers. This chapter derives the normalised Bayesian equaliser with scalar channel states (NBESS) and provides a fuzzy implementation for it. The computational issues relating to the developed fuzzy equaliser are also addressed and presented. The results of Monte Carlo simulations for bit error ratio (BER) performance have been presented to demonstrate the performance of the fuzzy equalisers developed here.

Chapter 4 analyses the problem of CCI in a DCS. The optimal equaliser for CCI channels is presented and a suboptimal fuzzy–CCI equaliser for this is derived. The results of Monte Carlo simulation for BER performance have been presented to demonstrate the performance of the fuzzy–CCI equaliser developed here in relation to some of the other equalisation techniques.

Chapter 5 summarises the work undertaken in this thesis and points to possible directions for future research.

Chapter 2
Background

2.1 Introduction

This thesis discusses the development of fuzzy system based channel equalisers for a variety of channel impairments. In order to establish the context and need for the work undertaken clearly and coherently, it is necessary to discuss the fundamental concepts involved in various aspects of this study. This chapter brings out the need for an adaptive equaliser in a DCS and describes the classification of adaptive equalisers.

This chapter is organised as follows. Following this introduction, section 2.2 discusses the communication system in general and section 2.3 discusses the propagation channel model in a DCS, providing the general finite impulse response (FIR) filter model for ISI channels and CCI channels. Section 2.4 presents a classification of equalisers with emphasis on symbol-by-symbol equalisers. Section 2.5 derives the decision function for the optimal Bayesian symbol-by-symbol equaliser for ISI channels. Sections 2.6 and 2.7 provide a short overview of developments of linear and nonlinear equalisers respectively. Finally, section 2.8 provides the concluding remarks.

2.2 Digital communication system

The block diagram of a general DCS is presented in Figure 2.1. A DCS, in general, may not have some of the blocks shown here. The *data source* constitutes the signal generation system that generates the information to be transmitted. Some of the typical examples of this are telephone, television and computer systems. The work of the *encoder* in the transmitter is to encode the information bits before transmission so as to provide redundancy in the system. This in turn helps in error correction at the receiver end. Some of the typical coding schemes used are *convolutional codes*, *block codes* and *grey codes*. The encoder does not form an essential part of the communication system but is being increasingly used. The digital data transmission requires very large bandwidth. The efficient use of the available bandwidth is achieved

through the *transmitter filter*, also called the modulating filter. The *modulator* on the other hand places the signal over an high frequency carrier for efficient transmission. Some of the typical modulation schemes used in digital communication systems are amplitude shift keying (ASK), frequency shift keying (FSK), pulse amplitude modulation (PAM) and phase shift keying (PSK) modulation. The *channel* is the medium through which information propagates from the transmitter to the receiver. At the receiver the signal is first *demodulated* to recover the

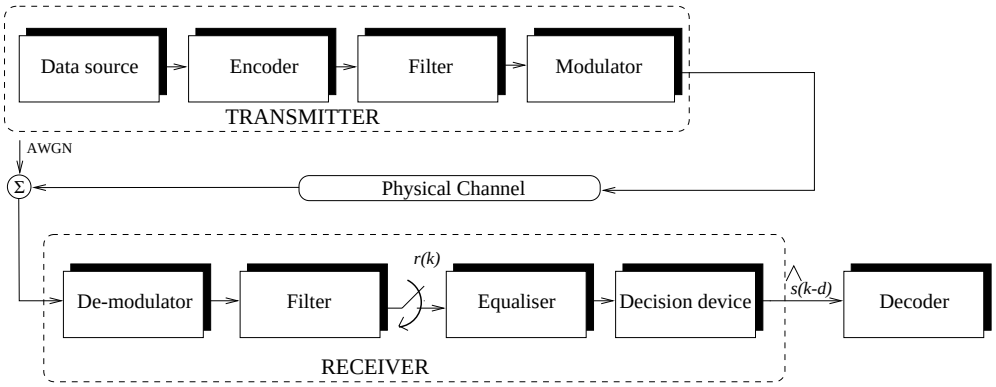


Figure 2.1: *Block diagram of a digital communication system*

baseband transmitted signal. This demodulated signal is processed by the *receiver filter*, also called receiver demodulating filter, which should be ideally matched to the transmitter filter and channel[1]. The *equaliser* in the receiver removes the distortion introduced due to the channel impairments. The *decision device* provides the estimate of the encoded transmitted signal. The *decoder* reverses the work of the encoder and removes the encoding effect revealing the transmitted information symbols.

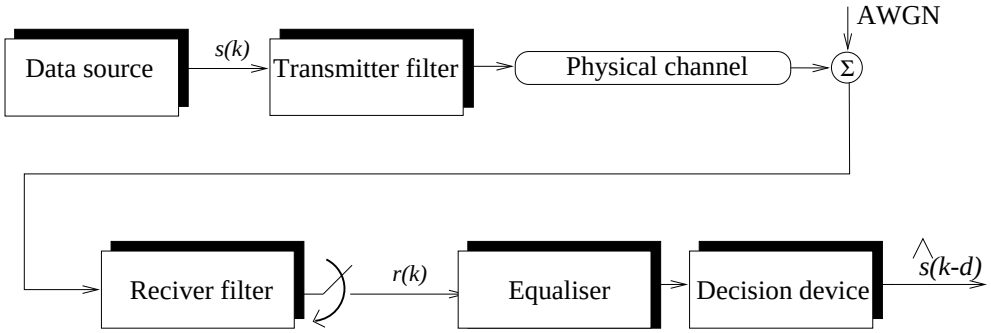


Figure 2.2: *Baseband model of digital communication system*

This DCS system in Figure 2.1 has all the necessary blocks. But, the analysis of this system

[1]Normally the channel transfer function is not known to the receiver and may be non-stationary. For this reason the receiver is matched to the transmitter filter only.

is very difficult due to the complexity associated with all the subsystems. For this reason communication systems are studied in the baseband frequency. Figure 2.2 presents the equivalent baseband model of the DCS presented in Figure 2.1. Here the encoder, decoder, modulator and the demodulators have been removed. This simplified communication system model, while maintaining the basic principles involved, is easy to analyse.

2.3 Propagation channel

This section discusses the channel impairments that limit the performance of a DCS. The DCS considered here is shown in Figure 2.2. The transmission of digital pulses over analogue communication channel would require infinite bandwidth[2]. An ideal physical propagation channel should behave like an ideal low pass filter represented by its frequency response,

$$\mathcal{H}_C(f) = |\mathcal{H}_C(f)| \exp(j\theta f) \tag{2.1}$$

where, $\mathcal{H}_C(f)$ represents the Fourier transform (FT) of the channel and θ is the phase response of the channel . The amplitude response of the channel $|\mathcal{H}_C(f)|$ can be defined as,

$$|\mathcal{H}_C(f)| = \begin{cases} \kappa_1 & |f| \leq \omega_c \\ 0 & |f| > \omega_c \end{cases} \tag{2.2}$$

where, κ_1 is a constant and ω_c is the upper cutoff frequency. The channel group delay characteristic is given by,

$$\tau(f) = -\frac{1}{2\pi}\frac{d\theta(f)}{df} = \kappa_2 \tag{2.3}$$

where κ_2 is an arbitrary constant. The conditions described in (2.2) and (2.3) constitute fixed amplitude and linear phase characteristics of a channel. This channel can provide distortion free transmission of analogue signal band limited to ω_c . Transmission of the infinite bandwidth digital signal over a band limited channel of ω_c will obviously cause distortion. This demands for the infinite bandwidth digital signal be band limited to at least ω_c, to guarantee distortion free transmission. This work is done with the aid of transmitter and receiver filters shown in

[2]The essential bandwidth of the signal is finite but some portion of signal may extend over infinite bandwidth

Figure 2.2. The combined frequency response of the physical channel, transmitter filter and the receiver filter can be represented as,

$$\mathcal{H}(f) = \mathcal{H}_T(f)\mathcal{H}_C(f)\mathcal{H}_R(f) \tag{2.4}$$

where, $\mathcal{H}_T(f),\ \mathcal{H}_C(f)$ and $\mathcal{H}_R(f)$ represent the FT of transmitter filter, propagation channel and the receiver filter respectively. When the receiver filter is matched to the combined response of the propagation channel and the transmitter filter, the system provides optimum signal to noise ratio (SNR) [43] at the sampling instant. The channel response is generally not known to the receiver beforehand. For this reason the receiver filter impulse response $h_R(t)$ is generally matched to the transmitter filter impulse response $h_T(t)$. This condition can be represented as

$$\mathcal{H}_R(f) = \mathcal{H}_T^{\star}(f) \tag{2.5}$$

$$h_R(t) = h_T^{\star}(-t) \tag{2.6}$$

where, $\mathcal{H}_T^{\star}(f)$ and $h_T^{\star}(t)$ are complex conjugates of $\mathcal{H}_T(f)$ and $h_T(t)$ respectively. It is desired to select $\mathcal{H}(f)$ so as to minimise the distortion at the output of the receiver filter at sampling instants. For the ideal channel presented in (2.1), the design of transmitter and receiver filters is critical for achieving distortion free transmission. One such filter capable of satisfying this criterion is the raised cosine filter given by,

$$\mathcal{H}_{TR}(f) = \begin{cases} T & 0 \leq f \leq \frac{1-\beta}{2T} \\ \frac{T}{2}\left\{1 + \cos\left[\frac{\pi T}{\beta}(|f| - \frac{1-\beta}{2T})\right]\right\} & \frac{1-\beta}{2T} \leq |f| \leq \frac{1+\beta}{2T} \\ 0 & |f| > \frac{1+\beta}{2T} \end{cases} \tag{2.7}$$

$$\mathcal{H}_{TR}(f) = \mathcal{H}_T(f)\mathcal{H}_R(f) \tag{2.8}$$

where, T is the source symbol period and β, $0 \leq \beta \leq 1$, is the excess bandwidth and $\mathcal{H}_{TR}$ is the FT of the combined response of transmitter and receiver filter. The plot of this combined filter response is presented in Figure 2.3. Figure 2.3(a) and Figure 2.3(b) represents the impulse response and frequency response of the combined filter respectively. From the Figures 2.3(a) and 2.3(b), it can be observed that any value of β can provide distortion free transmission if the receiver output is sampled at the correct time. A sampling timing error causes ISI, which reduces with an increase in β. The special case of $\beta = 0$ provides a pulse satisfying the

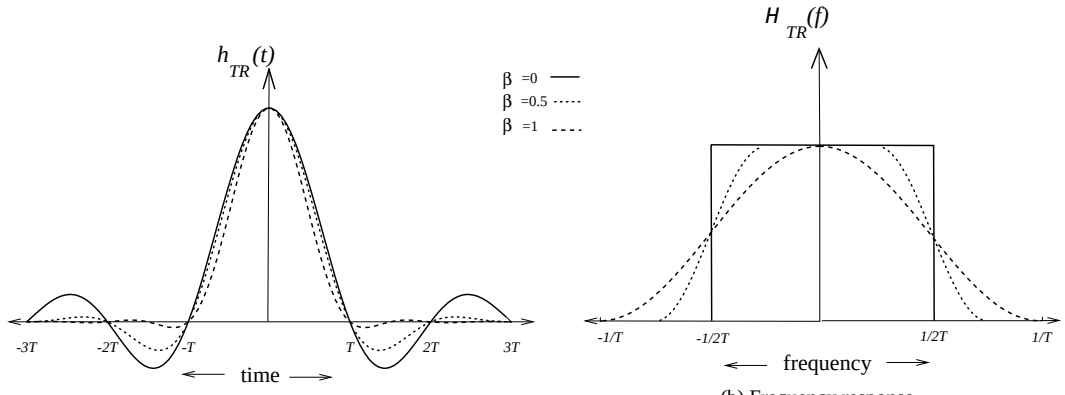


Figure 2.3: *Raised cosine pulse and its spectrum*

condition,

$$h_{TR}(t) = \frac{\sin\left(\frac{\pi t}{T}\right)}{\left(\frac{\pi t}{T}\right)} \tag{2.9}$$

Under this condition the channel can provide highest signalling rate[3], $T = \frac{1}{2\omega_c}$. At the other extreme, $\beta = 1$ provides a signalling rate equal to reciprocal of the bandwidth, $T = \frac{1}{\omega_c}$. In this process selection of β provides a compromise between quality and signalling speed.

Here it has been assumed that the physical channel is an ideal low pass filter (2.1). However, in reality all physical channels deviate from this behaviour. This introduces ISI even though the receiver is sampled at the correct time. The presence of this ISI requires an equaliser to provide proper detection.

In general all types of DCS's are affected by ISI. Communication systems are also affected by other forms of distortion. Multiple access techniques give rise to CCI and adjacent channel interference (ACI) in addition to ISI. The presence of amplifiers in the transmitter and the receiver front end causes nonlinear distortion. Fibre optic communication systems are also affected by nonlinear distortion [44]. On the other hand the mobile radio channels are affected by multi-path fading due to relative motion between the transmitter and receiver [45].

In the following subsections these channel impairments are discussed and the channel models are presented. These models are used in the later chapters for evaluating equalisation algorithms that have been presented in this thesis. The discussions in these subsections are limited only to

[3]This is critical Nyquist criteria

the channel effects that have been analysed in this thesis.

2.3.1 Inter symbol interference (ISI)

The cascade of the transmitter filter $h_T(t)$, the channel $h_C(t)$, the receiver matched filter $h_R(t)$ and the T-spaced sampler in the communication system shown in Figure 2.2 can be modelled by a digital FIR filter. The noise at the equaliser input is correlated due to the presence of the matched filter. To take care of this, and since it is easier to deal with a white noise sequence in the equaliser, the equaliser is generally preceded by a noise whitening filter. This combined channel due to the transmitter filter, propagation channel, receiver filter, noise whitening filter and the T-spaced sampler can be modelled by the digital FIR filter represented in Figure 2.4. Here the channel observed output $r(k)$ is given by the sum of the noise free channel output $\widehat{r}(k)$,

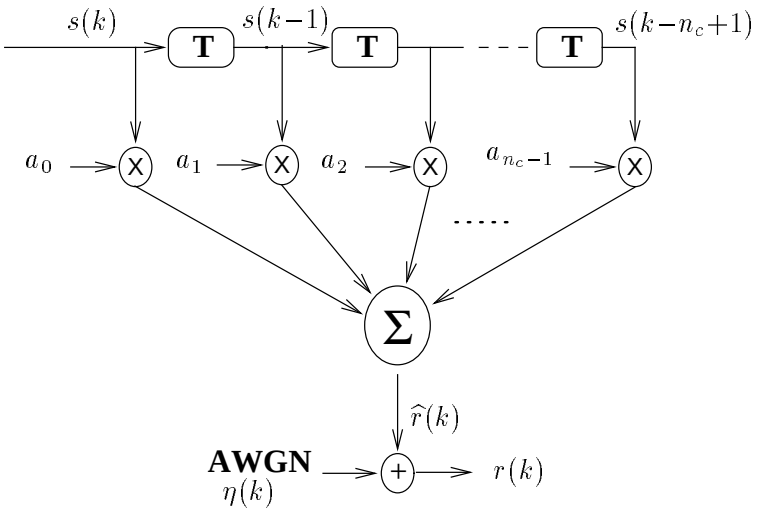


Figure 2.4: *Finite impulse response filter channel model*

which in turn is formed by the convolution of the transmitted sequence $s(k)$ with the channel taps a_i, $0 \leq i \leq n_c - 1$ and AWGN $\eta(k)$. The channel impulse response in the z-domain can be represented by the equation

$$H(z) = \sum_{i=0}^{n_c-1} a_i z^{-i} \tag{2.10}$$

where, the channel provides a dispersion up to n_c samples. This *discrete time white noise linear filter model* of the continuous channel will be used in the remaining part of the thesis for evaluation of equaliser algorithms. Here the AWGN, $\eta(k)$, is characterised by its variance σ_η^2.

2.3.2 Co-channel interference (CCI) and adjacent channel interference (ACI)

CCI and ACI occur in communication systems due to multiple access techniques using space, frequency or time. When the signal of interest in a communication system is corrupted by another signal occupying the same frequency band, CCI occurs. However, the source of ACI can be attributed to inadequate inter carrier spacing and non ideal receiver filter characteristics. In twisted pair cables CCI occurs due to interference of signals between different twisted pairs and is termed near end cross talk (NEXT), and far end cross talk (FEXT) [46, 47]. In DCR the CCI can be attributed to interference from cells of neighbouring clusters using the same carrier frequency [48] and ACI is due to inter carrier spacing between different cells in time division multiple access (TDMA) [49] and inter carrier spacing among carriers in the same cell in FDMA [48, 50, 51] systems. The frequency spectrum of the signals that carry the desired signal, the CCI and ACI signals is presented in Figure 2.5.

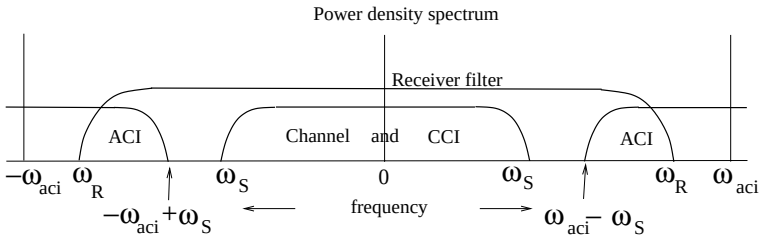


Figure 2.5: *Spectrum of desired signal, CCI and ACI in DCS*

Here the signal of interest occupies a double sided bandwidth of ω_s. The CCI signal also occupies the same frequency band [4]. The ACI signal centre frequency is spaced at ω_{aci} w.r.t. the desired carrier. The receiver filter rejects signal beyond ω_R. The guard band provided in the system is $\omega_{aci} - 2\omega_s$. From the figure it can be seen that a portion of the signal spectrum in the neighbouring carrier w.r.t. the signal of interest is received by the receiver filter and this signal is the main cause of ACI. The main reasons for this ACI can be attributed to non ideal cutoff characteristics of the receiver filter and close spacing of the carrier frequencies. Discrete time representation of the channel, the co-channel and the adjacent channel interferers using digital filters is presented in Figure 2.6. This system consists of a channel $H(z)$ corrupted with L, CCI sources $H_{co\_j}(z)$, $1 \leq j \leq L$ and B, ACI sources $H_{aci\_j}(z)$, $1 \leq j \leq B$ each of which can be represented in the form of a FIR filter of the type presented in Figure 2.4. The channel is also additionally corrupted with AWGN, $\eta(k)$. The total CCI and ACI are presented as $\widehat{r}_{co}(k)$ and

[4] The CCI generally has a different spectrum

$\widehat{r}_{aci}(k)$ respectively. Here $s_0(k)$ are the transmitted symbols from the desired channel, $s_i(k)$, $1 \leq i \leq L$ represent the transmitted symbols from the co-channel i and $s_{aci\_j}(k)$ represent the transmitted symbols from adjacent channel j.

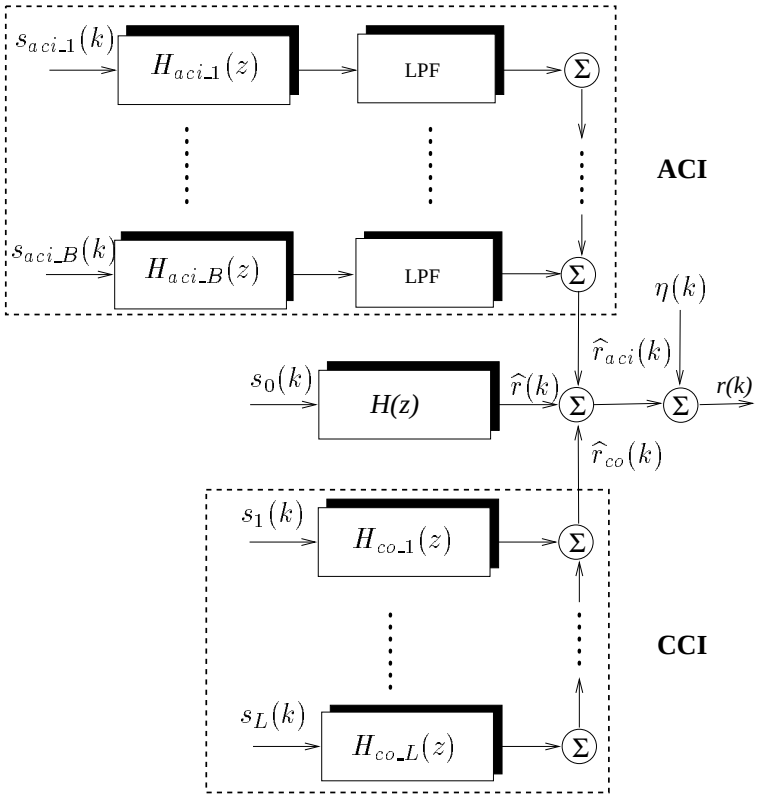


Figure 2.6: *FIR filter implementation of channel, CCI and ACI in digital communication system*

In Chapter 4, a modified form of this channel model will be used for investigating the performance of fuzzy equalisers in a CCI environment.

2.4 Equaliser classification

This section provides adaptive equaliser classification and specifies the domain of the investigation undertaken in this thesis. The general equaliser classification is presented in Figure 2.7. In general the family of adaptive equalisers can be classified as *supervised equalisers* and *unsupervised equalisers*. The channel distortions introduced into the transmitted signal in the process of transmission can be conveniently removed by transmitting a *training signal* or *pilot signal* periodically during the transmission of information. A replica of this pilot signal is available at the receiver and the receiver uses this to update its parameters during the

training period. These kinds of equalisers are known as supervised equalisers. However, the constraints associated with communication systems like digital television and digital radio do not provide the scope for the use of a training signal. In this situation the equaliser needs some form of unsupervised or *self recovery* method to update its parameters so as to provide near optimal performance. These equaliser are called *blind equalisers*. After training, the equaliser is switched to *decision directed* mode, where the equaliser can update its parameters based on the past detected samples. This thesis investigates supervised equalisers in general.

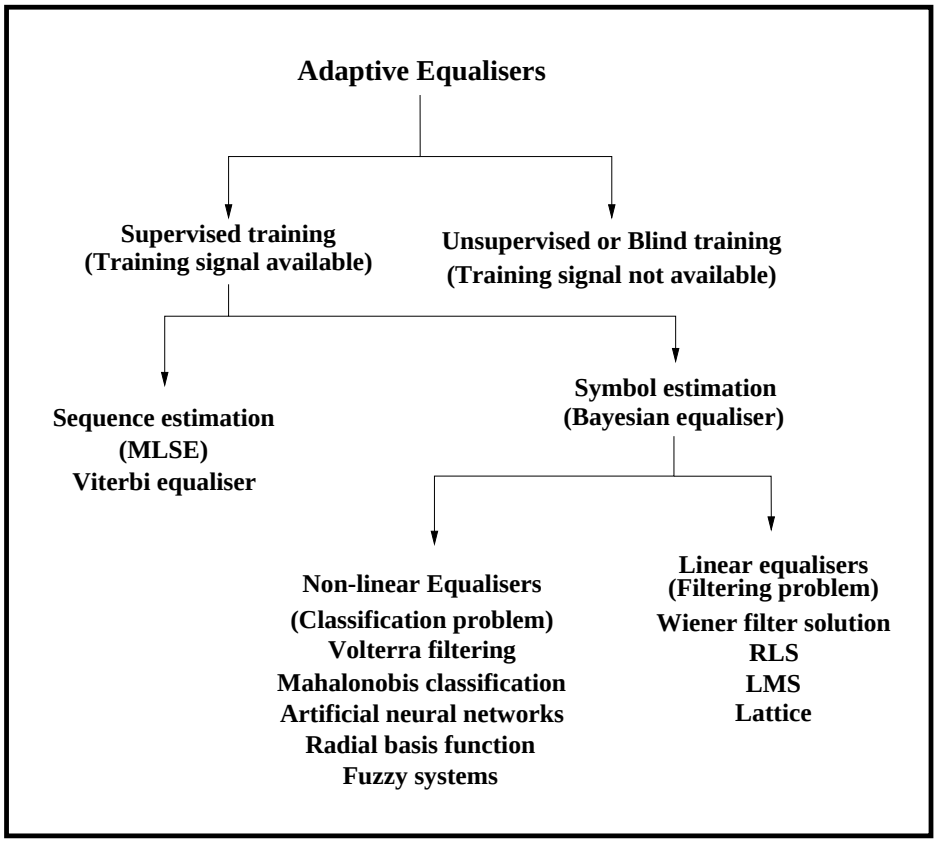


Figure 2.7: *Adaptive equaliser classification*

The process of supervised equalisation can be achieved in two forms. These are *sequence estimation* and *symbol-by-symbol estimation*. The sequence estimator uses the sequence of past received samples to estimate the transmitted symbol. For this reason this forms of equaliser is considered as an infinite memory equaliser and is termed MLSE [3]. The MLSE can be implemented with the Viterbi Algorithm [4]. An infinite memory sequence estimator provides the best bit error ratio (BER) performance for equalisation of time invariant channels. The symbol-by-symbol equaliser on the other hand works as a finite memory equaliser and uses a fixed number of input samples to detect the transmitted symbol. The optimum decision function

for this type of equaliser is given by MAP criterion and can be derived by Bayes's theory [52]. Hence this optimum finite memory equaliser is also called the Bayesian equaliser [53]. An infinite memory Bayesian equaliser can provide a performance better than the MLSE, but its computational complexity is very large. A finite memory Bayesian equaliser can provide performance comparable to the MLSE but with a reduced computational complexity [54].

The Bayesian equaliser provides the lower performance bound for symbol-by-symbol equalisers in terms of probability of error or BER and can be implemented with *linear* or *nonlinear* systems. The linear adaptive equaliser is a linear FIR adaptive filter [55] trained with an adaptive algorithm like the LMS, RLS or lattice algorithm. These linear equalisers treat equalisation as inverse filtering and during the process of training optimise a certain performance criteria like minimum mean square error (MMSE) or amplitude distortion. Linear equalisers trained with MMSE criterion provide the Wiener filter[56] solution. Recent advances in nonlinear signal processing techniques have provided a rich variety of nonlinear equalisers. Some of the equalisers developed with these processing techniques are based on Volterra filters, ANN, perceptrons, MLP, RBF networks, fuzzy filters and fuzzy basis functions. A review of some of these equalisation techniques can be seen in [28–30]. All of these nonlinear equalisers, during their training period, optimise some form of a cost function like the MSE or probability of error and have the capability of providing the optimum Bayesian equaliser performance in terms of BER. The nonlinear equalisers treat equalisation as a pattern classification process where the equaliser attempts to classify the input vector into a number of transmitted symbols. The fuzzy equalisers investigated in this thesis fall into this category.

Another form of nonlinear equaliser that can be constructed with any of the symbol-by-symbol based equalisers is the DFE, where previously made decisions are used for estimating the present and the future decisions. This equaliser is also considered as a infinite memory equaliser. The conventional DFE using a linear filter is designated as a nonlinear equaliser in a wide varities of communication literature since the decision function used here forms a nonlinear combination of the received samples which is, in fact the linear combination of the received samples and previously detected samples. In this thesis the term nonlinear equalisers is used exclusively for those equalisers that provide a nonlinear decision function based on received samples or the received samples along with previously detected samples. The following two sections analyse some of the linear and nonlinear equalisers in greater detail.

2.5 Optimal symbol-by-symbol equaliser : Bayesian equaliser

In this section the optimum symbol-by-symbol equaliser decision function is derived. This equaliser is termed as Bayesian equaliser. To derive the equaliser decision function the discrete time model of the baseband digital communication system presented in Figure 2.8 is considered. The channel is modelled as an FIR filter as in Figure 2.4. The equaliser uses an input vector $\mathbf{r}(k) \in \mathbb{R}^m$, the m dimensional space. The term m is the equaliser length and the equaliser order can be considered as $m-1$. The equaliser provides a decision function $\mathfrak{F}\{(\mathbf{r}(k)\}$ based on the input vector and this is passed through a decision device to provide the estimate of transmitted signal $\widehat{s}(k-d)$ where d is a delay associated with equaliser decision. The communication system is assumed to be a two level PAM system where the transmitted sequence $s(k)$ is drawn from a independent identically distributed (i.i.d.) sequence comprising of $\{\pm 1\}$ symbols. The noise source $\eta(k)$ is assumed to be zero mean white Gaussian with a variance of σ_η^2. The received signal $r(k)$ at the sampling instant k can be represented as,

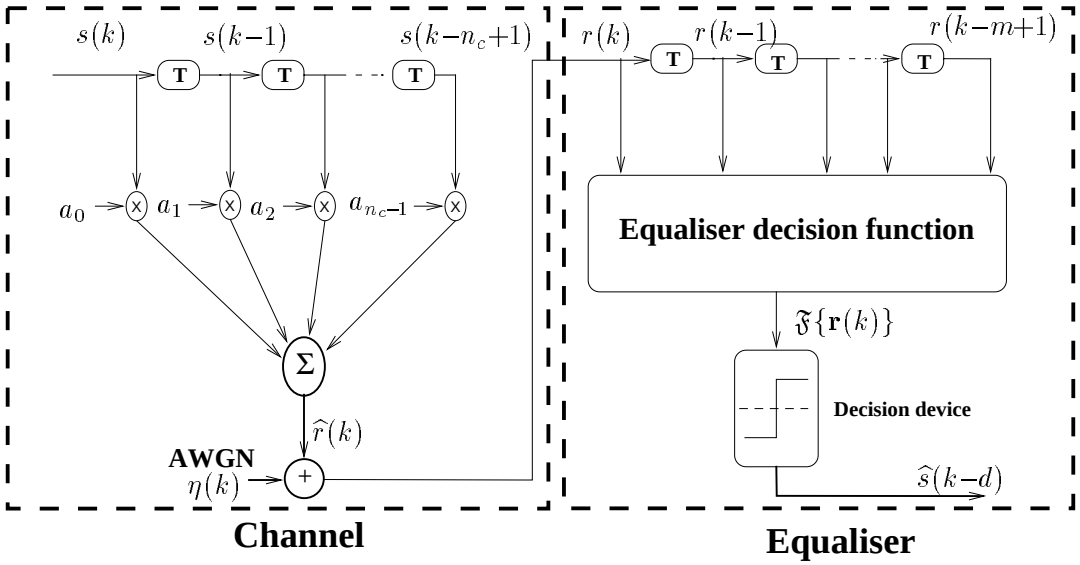


Figure 2.8: *Discret time model of a digital communication system*

$$\begin{aligned} r(k) &= \widehat{r}(k) + \eta(k) \\ &= \sum_{i=0}^{n_c-1} a_i s(k-i) + \eta(k) \end{aligned} \tag{2.11}$$

The equaliser performance is described by the probability of misclassification w.r.t. SNR. The

SNR is defined as,

$$\begin{aligned} \text{SNR} &= \frac{\mathcal{E}[\widehat{r}(k)^2]}{\mathcal{E}[\eta(k)^2]} \\ &= \frac{\sigma_s^2 \sum_{i=0}^{n_c-1} a_i^2}{\sigma_\eta^2} \end{aligned} \tag{2.12}$$

where, $\mathcal{E}$ is the expectation operator, σ_s^2 represent the signal power and $\sum_{i=0}^{n_c-1} a_i^2$ is the channel power. With the assumption that the signal is drawn from an i.i.d. sequence of $\{\pm 1\}$, the signal power $\sigma_s^2 = 1$. With this the system SNR can be represented as,

$$\text{SNR} = \frac{\sum_{i=0}^{n_c-1} a_i^2}{\sigma_\eta^2} \tag{2.13}$$

The equaliser uses the received signal vector $\mathbf{r}(k) = [r(k),\, r(k-1),\, \ldots,\, r(k-m+1)]^T \in \mathbb{R}^m$ to estimate the delayed transmitted symbol $s(k-d)$. The equaliser with its decision function and a memoryless detector to quantise the real valued output from decision function $\mathfrak{F}\{\mathbf{r}(k)\}$, provides an estimate of the transmitted signal. The memoryless detector is implemented using a $\text{sgn}(x)$ function given by,

$$\text{sgn}\,(x) = \begin{cases} +1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{2.14}$$

The process of equalisation discussed here can be viewed as a classification process in which the equaliser partitions the input space $\mathbf{r}(k) \in \mathbb{R}^m$ into two regions corresponding to each of the transmitted sequences $+1/-1$ [24, 53, 57]. The locus of points which separate these two regions is termed as the decision boundary. The partition which provides the minimum probability of misclassification is the Bayesian decision boundary derived with the MAP criterion.

2.5.1 Channel states

To derive the Bayesian equaliser decision function the concept of channel states is introduced first. The equaliser input vector has been defined as $\mathbf{r}(k) = [r(k),\, r(k-1),\, \ldots,\, r(k-m+$

$1)]^T \in \mathbb{R}^m$ and $r(k) = \widehat{r}(k) + \eta(k)$. The vector $\widehat{\mathbf{r}}(k)$ is the noise free received signal vector and $\widehat{\mathbf{r}}(k) = [\widehat{r}(k), \widehat{r}(k-1), \ldots, \widehat{r}(k-m+1)]^T \in \mathbb{R}^m$. Each of these possible noise free received signal vectors constitutes a channel state. The channel states are determined by the transmitted symbol vector $\mathbf{s}(k) = [s(k), s(k-1), \ldots, s(k-m-n_c+2)]^T \in \mathbb{R}^{m+n_c-1}$. Here $\widehat{\mathbf{r}}(k)$ can be represented as $\widehat{\mathbf{r}}(k) = \mathbf{H}[\mathbf{s}(k)]$, where matrix $\mathbf{H} \in \mathbb{R}^{m \times (m+n_c-1)}$ is the channel matrix.

$$\mathbf{H} = \begin{bmatrix} a_0 & a_1 & \cdots & a_{n_c-1} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & a_0 & \cdots & a_{n_c-2} & a_{n_c-1} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & a_0 & \cdots & a_{n_c-1} \end{bmatrix} \tag{2.15}$$

Since $\mathbf{s}(k)$ has $N_s = 2^{m+n_c-1}$ combinations, $\widehat{\mathbf{r}}(k)$ has N_s states. These channel states are constructed with N_s sequences of $\mathbf{s}(k)$, which can be denoted as,

$$\mathbf{s}_j(k) = [s_j(k), s_j(k-1), ..., s_j(k-m-n_c+2)]^T, \quad 1 \leq j \leq N_s \tag{2.16}$$

The corresponding channel states are denoted as $\mathbf{c}_j$ and are given by

$$\mathbf{c}_j = \widehat{\mathbf{r}}(k) = \mathbf{H}[\mathbf{s}_j(k)], \quad 1 \leq j \leq N_s \tag{2.17}$$

The channels state matrix $\mathbf{C}_d = \{\mathbf{c}_j\}, 1 \leq j \leq N_s$, can be partitioned into two subsets depending on the transmitted symbol $s(k-d)$, i.e.,

$$\mathbf{C}_d = \mathbf{C}_d^+ \cup \mathbf{C}_d^- \tag{2.18}$$

where,

$$\begin{aligned} \mathbf{C}_d^+ &= \{\widehat{\mathbf{r}}(k) \mid s(k-d) = +1\} \\ \mathbf{C}_d^- &= \{\widehat{\mathbf{r}}(k) \mid s(k-d) = -1\} \end{aligned} \tag{2.19}$$

No.	$\mathbf{c}_j$	$s(k)$	$s(k-1)$	$s(k-2)$	$\widehat{\mathbf{r}}(k)$	
					$\widehat{r}(k)$	$\widehat{r}(k-1)$
1	$\mathbf{c}_1$	1	1	1	1.5	1.5
2	$\mathbf{c}_2$	1	1	−1	1.5	−0.5
3	$\mathbf{c}_3$	1	−1	1	−0.5	0.5
4	$\mathbf{c}_4$	1	−1	−1	−0.5	−1.5
5	$\mathbf{c}_5$	−1	1	1	0.5	1.5
6	$\mathbf{c}_6$	−1	1	−1	0.5	−0.5
7	$\mathbf{c}_7$	−1	−1	1	−1.5	0.5
8	$\mathbf{c}_8$	−1	−1	−1	−1.5	−1.5

Table 2.1: *Channel states calculation for channel* $H(z) = 0.5 + 1.0z^{-1}$ *with* $m = 2$, $d = 0$ *and* $N_s = 8$

Each of the sets of the channel state matrix $\mathbf{C}_d^+$ and $\mathbf{C}_d^-$ contain $\frac{N_s}{2}$ channel states. Here the channel states $\mathbf{c}_j \in \mathbf{C}_d^+$ are termed the positive channel states and $\mathbf{c}_j \in \mathbf{C}_d^-$ are termed the negative channel states.

EXAMPLE 2.1

An example is considered to show the channel states. The channel considered here is represented by its z-transform,

$$H(z) = H_1(z) = 0.5 + 1.0z^{-1} \tag{2.20}$$

This channel is a non-minimum phase channel with its zero outside the unit circle (located at $z = -2.0$). The equaliser length considered here is $m = 2$. This equaliser has $N_s = 8$ channel states. The channel states for this equaliser are presented in Table 2.1 and are located at $\widehat{\mathbf{r}}(k)$ with its components taken from scalars $[\widehat{r}(k), \widehat{r}(k-1)]^T$.

2.5.2 Bayesian equaliser decision function

The presence of AWGN makes the channel observation vector $\mathbf{r}(k)$ a random process having a conditional Gaussian density function centred at each noise free received vector $\widehat{\mathbf{r}}(k)$. Given this to be the channel state $\widehat{\mathbf{r}}(k) = \mathbf{c}_j$, $1 \leq j \leq N_s$, the conditional probability density distribution of the observed vector is,

$$p(\mathbf{r}(k) \mid \mathbf{c}_j) = (2\pi\sigma_\eta^2)^{-m/2} \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_j\|^2}{2\sigma_\eta^2}\right) \tag{2.21}$$

where $\|.\|$ constitute the Euclidean distance. If the received signal vector is perturbed sufficiently to cross the decision boundary due to the presence of AWGN, mis-classification results. To minimise the probability of mis-classification for a given received signal vector $\mathbf{r}(k)$ [52, 58], the transmitted symbol should be estimated based on $s(k) \in \{\pm 1\}$ having maximum *a-posteriori* probability $P(s(k-d) = s \mid \mathbf{r}(k))$. The decision device at the equaliser output provides a decision

$$\widehat{s}(k-d) = \text{sgn}\left(\mathfrak{F}\{\mathbf{r}(k)\}\right) = \begin{cases} +1 & \text{if } \mathfrak{F}\{\mathbf{r}(k)\} \geq 0 \\ -1 & \text{if } \mathfrak{F}\{\mathbf{r}(k)\} < 0 \end{cases} \tag{2.22}$$

where $\mathfrak{F}\{\mathbf{r}(k)\}$ is the Bayesian equaliser decision function that compares the *a-posteriori* probabilities of the binary transmitted symbol, i.e.,

$$\mathfrak{F}\{\mathbf{r}(k)\} = P\Big(s(k-d) = +1 \mid \mathbf{r}(k)\Big) - P\Big(s(k-d) = -1 \mid \mathbf{r}(k)\Big) \tag{2.23}$$

where $P\Big(s(k-d) = +1 \mid \mathbf{r}(k)\Big)$ and $P\Big(s(k-d) = -1 \mid \mathbf{r}(k)\Big)$ are the *a-posteriori* probabilities that the transmitted signal is $+1$ or -1 respectively, having observed the received signal vector $\mathbf{r}(k)$. This function is the Bayesian decision function where Bayes's rule [52] is applied to express the *a-posteriori* probability into the product of the *a-priori* probability $P(s(k-d) = s)$ and the state conditional probability distribution function (pdf) $p\left(\mathbf{r}(k) \mid s(k-d) = s\right)$ over the pdf of $\mathbf{r}(k)$,

$$P\Big(s(k-d) \mid \mathbf{r}(k)\Big) = \frac{p\Big(\mathbf{r}(k) \mid s(k-d) = s\Big) P\Big(s(k-d) = s\Big)}{p\Big(\mathbf{r}(k)\Big)} \tag{2.24}$$

The *a-priori* and the state conditional probabilities can be calculated in terms of the channel and the noise statistics. If the transmitted symbol is i.i.d., the *a-priori* probabilities of the transmitted signal $s(k-d)$, $P(s(k-d)) = +1$ and $P(s(k-d)) = -1$ have equal value of $\frac{1}{2}$. The state conditional pdf $p(\mathbf{r}(k) \mid s(k-d) = +1)$, is the sum of pdf for each of channel states

$\mathbf{c}_j \in \mathbf{C}_d^+$ and is described as,

$$\begin{aligned} p\Big(\mathbf{r}(k) \mid s(k-d) = +1\Big) &= \frac{1}{N_s} \sum_{\mathbf{c}_j \in \mathbf{C}_d^+} p(\mathbf{r}(k) \mid \mathbf{c}_j) \\ &= \frac{1}{N_s} \sum_{\mathbf{c}_j \in \mathbf{C}_d^+} (2\pi\sigma_\eta^2)^{-m/2} \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_j\|^2}{2\sigma_\eta^2}\right) \end{aligned} \tag{2.25}$$

where $\frac{1}{N_s}$ is the *a-priori* probability of $\mathbf{c}_j$. Similarly the conditional p.d.f of $p\Big(\mathbf{r}(k) \mid s(k-d) = -1\Big)$ can be expressed as,

$$\begin{aligned} p\Big(\mathbf{r}(k) \mid s(k-d) = -1\Big) &= \frac{1}{N_s} \sum_{\mathbf{c}_j \in \mathbf{C}_d^-} p(\mathbf{r}(k) \mid \mathbf{c}_j) \\ &= \frac{1}{N_s} \sum_{\mathbf{c}_j \in \mathbf{C}_d^-} (2\pi\sigma_\eta^2)^{-m/2} \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_j\|^2}{2\sigma_\eta^2}\right) \end{aligned} \tag{2.26}$$

With this the Bayesian decision function can be derived by substituting (2.24) into (2.23) leading to,

$$\begin{aligned} \mathfrak{F}\{\mathbf{r}(k)\} = &\frac{p\Big(\mathbf{r}(k) \mid s(k-d) = +1\Big) P\Big(s(k-d) = +1\Big)}{p\Big(\mathbf{r}(k)\Big)} \\ - &\frac{p\Big(\mathbf{r}(k) \mid s(k-d) = -1\Big) P\Big(s(k-d) = -1\Big)}{p\Big(\mathbf{r}(k)\Big)} \end{aligned} \tag{2.27}$$

The *a-priori* probabilities of both the transmitted symbols is same and hence the denominator of both the parts on the right hand side of (2.27) have the same value. Moreover in the process of equalisation the sign of the decision function is of interest since it is passed through the $\mathrm{sgn}(x)$ function. With these assumptions, the decision function can be represented as,

$$\mathfrak{F}\{\mathbf{r}(k)\} = p\Big(\mathbf{r}(k) \mid s(k-d) = +1\Big) - p\Big(\mathbf{r}(k) \mid s(k-d) = -1\Big) \tag{2.28}$$

Substituting the values from (2.25) and (2.26) the decision function can be represented as,

$$\begin{aligned}
\mathfrak{F}\{\mathbf{r}(k)\} &= \frac{1}{N_s}\sum_{\mathbf{c}_j\in\mathbf{C}_d^+} p(\mathbf{r}(k)\mid\mathbf{c}_j) - \frac{1}{N_s}\sum_{\mathbf{c}_i\in\mathbf{C}_d^-} p(\mathbf{r}(k)\mid\mathbf{c}_i) \\
&= \frac{1}{N_s}\sum_{\mathbf{c}_j\in\mathbf{C}_d^+} (2\pi\sigma_\eta^2)^{-m/2}\exp\left(\frac{-\|\mathbf{r}(k)-\mathbf{c}_j\|^2}{2\sigma_\eta^2}\right) \\
&\quad - \frac{1}{N_s}\sum_{\mathbf{c}_i\in\mathbf{C}_d^-} (2\pi\sigma_\eta^2)^{-m/2}\exp\left(\frac{-\|\mathbf{r}(k)-\mathbf{c}_i\|^2}{2\sigma_\eta^2}\right) \\
&= \frac{1}{N_s}(2\pi\sigma_\eta^2)^{-m/2}\left\{\sum_{\mathbf{c}_j\in\mathbf{C}_d^+}\exp\left(\frac{-\|\mathbf{r}(k)-\mathbf{c}_j\|^2}{2\sigma_\eta^2}\right)\right. \\
&\quad \left. - \sum_{\mathbf{c}_i\in\mathbf{C}_d^-}\exp\left(\frac{-\|\mathbf{r}(k)-\mathbf{c}_i\|^2}{2\sigma_\eta^2}\right)\right\}
\end{aligned} \tag{2.29}$$

Removing the scaling term $\frac{1}{N_s}(2\pi\sigma_\eta^2)^{-m/2}$ from the right hand side, since the sign of the decision function is sufficient to provide the decision, yields,

$$\begin{aligned}
\mathfrak{F}\{\mathbf{r}(k)\} &= \sum_{\mathbf{c}_j\in\mathbf{C}_d^+}\exp\left(\frac{-\|\mathbf{r}(k)-\mathbf{c}_j\|^2}{2\sigma_\eta^2}\right) - \sum_{\mathbf{c}_i\in\mathbf{C}_d^-}\exp\left(\frac{-\|\mathbf{r}(k)-\mathbf{c}_i\|^2}{2\sigma_\eta^2}\right) \\
&= \sum_{i=1}^{N_s} w_i\exp\left(\frac{-\|\mathbf{r}(k)-\mathbf{c}_i\|^2}{2\sigma_\eta^2}\right)
\end{aligned} \tag{2.30}$$

where $w_i = +1$, if $\mathbf{c}_i \in \mathbf{C}_d^+$ and $w_i = -1$, if $\mathbf{c}_i \in \mathbf{C}_d^-$. The decision function in (2.30) represents the Bayesian equaliser decision function. From the decision function it is obvious that the decision function is nonlinear and is completely specified in terms of the channel states and the noise characteristics. So, with the knowledge of the channel and the channel noise statistics, the Bayesian equaliser decision function can be found.

Below an example is considered to demonstrate the calculation of the Bayesian equaliser decision function.

EXAMPLE 2.2

As seen from the decision function of the Bayesian equaliser in (2.30), the optimal symbol-by-symbol equaliser decision function is dependent on the location of channel states, the noise statistics and the decision delay. The noise affects the spread associated

with channel state functions and this controls how fast the decision function approaches zero. When the noise is Gaussian, its effect on the decision function is not significant. This feature is presented here first. The system considered here is same as in Example 2.1 where

$$H(z) = 0.5 + 1.0z^{-1} \text{ with } m = 2 \text{ and } d = 0$$

This equaliser has $N_s = 8$ channel states. The channel states for this equaliser are presented in Table 2.1. The channel states $\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\} \in \mathbf{C}_d^+$ and $\{\mathbf{c}_5, \mathbf{c}_6, \mathbf{c}_7, \mathbf{c}_8\} \in \mathbf{C}_d^-$.

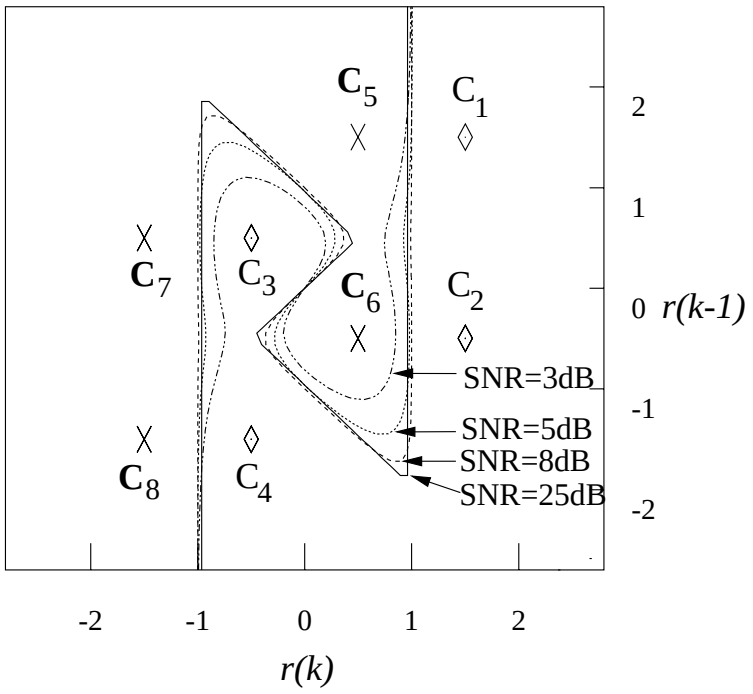


Figure 2.9: *Decision boundary of the Bayesian equaliser for channel* $H(z) = 0.5 + 1.0z^{-1}$, $m = 2$ $d = 0$, *with different SNR conditions,* $\Diamond$ *positive channel states and* $\times$ *negative channel states*

The decision boundary of the Bayesian equaliser for SNR =3 dB, 5 dB, 8 dB and 25 dB are presented in Figure 2.9 where the positive and negative channel states are presented with $\Diamond$ and $\times$ symbols respectively. From the decision boundary curves it is seen that, 8 dB to 25 dB change in SNR does not affect the decision boundary appreciably. From the decision boundary curves it can be inferred that as SNR $\rightarrow \infty$ the decision boundary can be asymptotically approximated with straight lines.

The Bayesian equaliser for a given SNR condition provides a set of decision boundaries for different decision delays. This effect of decision delay on the equaliser decision boundary is presented next. Figure 2.10 presents the decision boundary for the equaliser considered here for delay $d = 0, 1$ and 2. Here the SNR=8 dB. For $d = 0$, $\{\mathbf{c}_1,\ \mathbf{c}_2,$

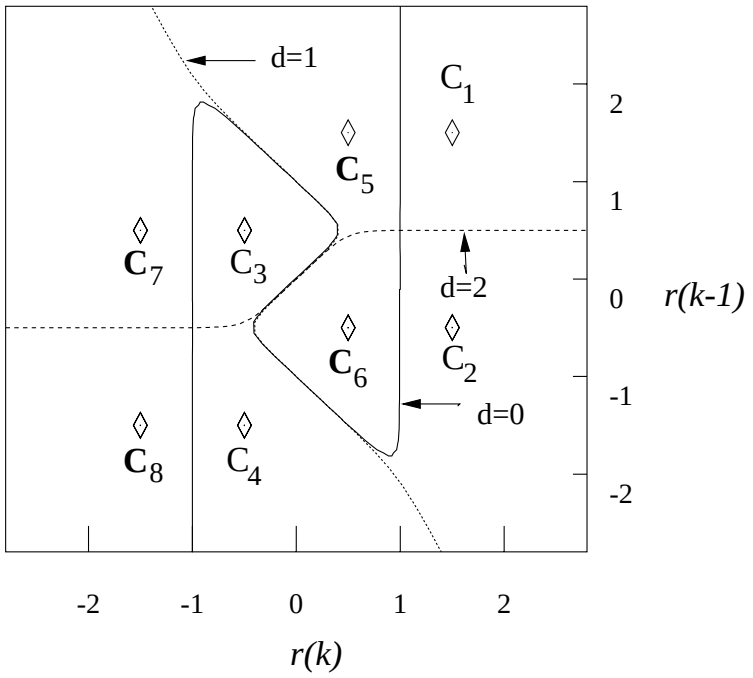


Figure 2.10: *Effect of decision delay on decision boundary for the Bayesian equaliser for for channel $H(z) = 0.5 + 1.0z^{-1}$, $m = 2$ and $\Diamond$represents the channel states*

$\mathbf{c}_3,\ \mathbf{c}_4\} \in \mathbf{C}_d^+$ and $\{\mathbf{c}_5,\ \mathbf{c}_6,\ \mathbf{c}_7,\ \mathbf{c}_8\} \in \mathbf{C}_d^-$. However, when $d = 1$, $\{\mathbf{c}_1,\ \mathbf{c}_2,\ \mathbf{c}_5,\ \mathbf{c}_6\} \in \mathbf{C}_d^+$ and $\{\mathbf{c}_3,\ \mathbf{c}_4,\ \mathbf{c}_7,\ \mathbf{c}_8\} \in \mathbf{C}_d^-$ and for $d = 2$, $\{\mathbf{c}_1,\ \mathbf{c}_3,\ \mathbf{c}_5,\ \mathbf{c}_7\} \in \mathbf{C}_d^+$ and $\{\mathbf{c}_2,\ \mathbf{c}_4,\ \mathbf{c}_6,\ \mathbf{c}_8\} \in \mathbf{C}_d^-$. From the decision boundary curves it is seen that each set of combinations of channel states corresponding to $\mathbf{C}_d^+$ and $\mathbf{C}_d^-$ provide different decision boundaries. It is interesting to note that the decision boundary for d =1 and 2, the groups of positive and negative channel states are linearly separable. But for $d = 0$ these states are nonlinearly separable. From the figure it is also observed that increasing the delay for this non-minimum phase channel[5] makes the decision boundary more linear. This accounts for better performance of the linear equalisers for these types of channel with maximum permissible delay [59], since the linear equalisers can only provide a linear decision boundary.

[5]Non-minimum phase channel has all its zeros outside the unit circle in z-plane

2.6 Symbol-by-symbol linear equalisers

This section introduces the concept of the linear equaliser. As discussed in section 2.4, the linear equalisers in this thesis refer to equalisers that provide a decision based on the linear combination of the input to the equaliser. If decision feedback is employed, the linear equaliser provides a decision function based on the linear combination of received samples and previously detected samples. The structure of a linear equaliser is presented in Figure 2.11. The equaliser consists of a T-spaced tapped delay line (TDL) which receives the receiver sampled input vector $\mathbf{r}(k) = [r(k), r(k-1), \ldots, r(k-m+1)]^T$ and provides an output $y(k)$ by weighted sum computation of input vector $\mathbf{r}(k)$ with weight vector $\mathbf{w}$. The output is computed once per symbol and can be represented as

$$y(k) = \sum_{i=0}^{m-1} w_i \, r(k-i) \tag{2.31}$$

The weight vector $\mathbf{w}$ optimises one of the performance criteria like zero forcing (ZF) or MMSE criteria. The decision device presented at the output of the filter provides the transmitted signal constellation.

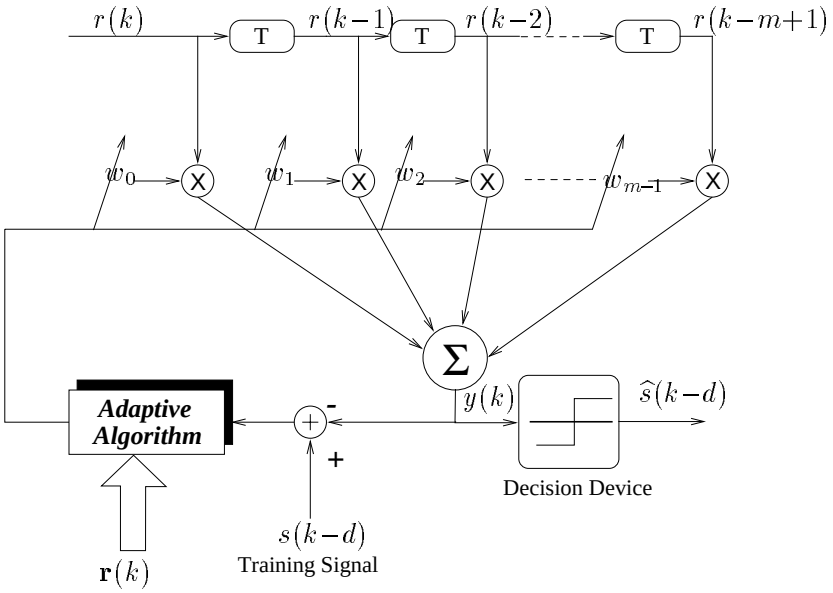


Figure 2.11: *Structure of a linear equaliser*

The ZF criteria is defined as the worst case ISI at the output of the equaliser. The condition for

minimisation of peak distortion can be presented as

$$C(z) = \frac{1}{H(z)} \tag{2.32}$$

Here $C(z)$ is the equaliser impulse response. With this, the combined equaliser and the channel response is zero for all but one coefficient. From the equaliser condition presented in (2.32) it can be seen that, for FIR channels, the equaliser is realisable when the zeros of the channel are inside the unit circle in the z-plane. When the zeros are outside the unit circle, the equaliser becomes unstable and hence unrealisable. Equalisation of this type of channel can be overcome by the introduction of a nonzero decision delay d [59].

The MMSE criteria provides equaliser tap coefficients $\mathbf{w}(k)$ to minimise the mean square error at the equaliser output before the decision device. This condition can be represented as

$$J = \mathcal{E}|e(k)|^2 \tag{2.33}$$

$$e(k) = s(k-d) - y(k) \tag{2.34}$$

where $e(k)$ is the error associated with filter output $y(k)$. The equaliser designed using ZF criteria neglects the effect of noise. However, the MMSE criteria optimises the equaliser weights for minimising the MMSE under noise and ISI. Minimisation of MMSE criteria provides equalisers that satisfy the Wiener criterion [56]. The evaluation of the equaliser weights with this criteria requires computation of matrix inversion and the knowledge of the channel, which in most cases is not available. However, adaptive algorithms like LMS [2] and RLS[55] can be used to recursively update the equaliser weights during the training period. the convergence properties and the performance of linear equalisation techniques have been well documented in the literature [22, 43, 60].

A DFE [61] using a linear filter is presented in Figure 2.12. This equaliser is characterised by its feed forward length m and the feedback order q. The equaliser uses m feed forward samples and q feedback samples from the previously detected samples. The feedback signal vector $\widehat{\mathbf{s}}(k) = [\widehat{s}(k-d-1),\ \widehat{s}(k-d-2),\ \ldots,\ \widehat{s}(k-d-q)]^T$ is associated with feedback weight vector $\mathbf{w}_f = [w_0^f,\ w_1^f,\ \ldots,\ w_{q-1}^f]^T$. The feedback section in the equaliser helps to remove the ISI contribution from the estimated symbols. This equaliser provides better performance than the conventional feed forward linear equaliser. When there is an error in the decision the error is fed back and this results in more errors due to error propagation. It has been observed

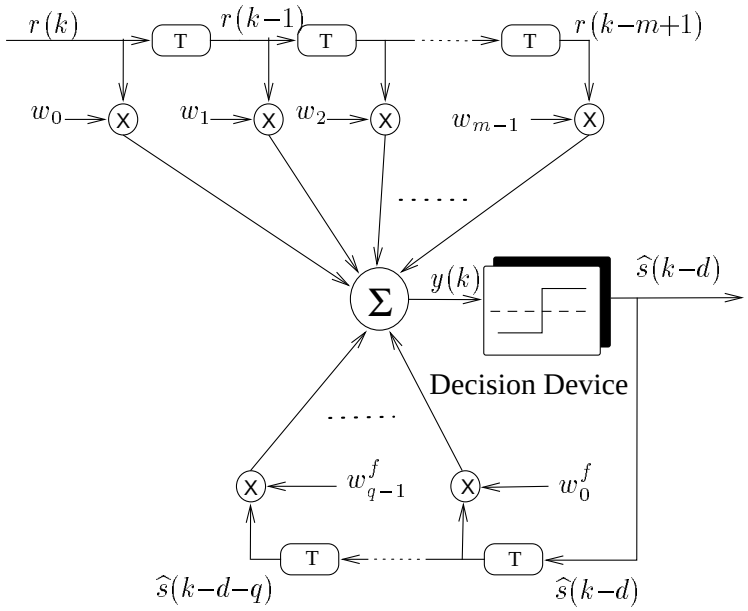


Figure 2.12: *Structure of a linear decision feedback equaliser*

that the equalisers can recover from this condition automatically and error propagation does not pose a considerable problem.

2.7 Symbol-by-symbol adaptive nonlinear equalisers

Some of the popular forms of nonlinear equalisers are introduced in this section. Nonlinear equalisers treat equalisation as a nonlinear pattern classification problem and provide a decision function that partitions the input space $\mathbb{R}^m$ to the number of transmitted symbols. As a result the equaliser assigns the input vector to one of the signal constellations. The nonlinear equalisers introduced in this section are based on the RBF networks and the ANN. Some of the other forms of nonlinear equalisers based on the recurrent RBF [35], the recurrent ANN [34], the Volterra filters [62], the functional link networks [63] and Mahalobonis classifiers [36] have not been discussed. This section also presents an introduction to fuzzy systems and adaptive fuzzy filters and their use as equalisers. Other fuzzy schemes like neuro fuzzy filter [12], ANN trained with fuzzy reasoning [64] have not been analysed.

2.7.1 Radial basis function equaliser

The RBF network was originally developed for interpolation in multidimensional space [9, 25, 65]. The schematic of this RBF network with m inputs and a scalar output is presented in Figure 2.13. This network can implement a mapping $f_{rbf} : \mathbb{R}^m \rightarrow \mathbb{R}$ by the function,

$$f_{rbf}\{\mathbf{x}(k))\} = \sum_{i=1}^{N_r} w_i \phi(\|\mathbf{x}(k) - \boldsymbol{\rho}_i\|) \tag{2.35}$$

Where $\mathbf{x}(k) \in \mathbb{R}^m$ is the input vector, $\phi(.)$ is the given function from $\mathbb{R}^+$ to $\mathbb{R}$, w_i, $1 \leq i \leq N_r$ are weights and $\boldsymbol{\rho}_i \in \mathbb{R}^m$ are known as RBF centres. This RBF structure can be extended for

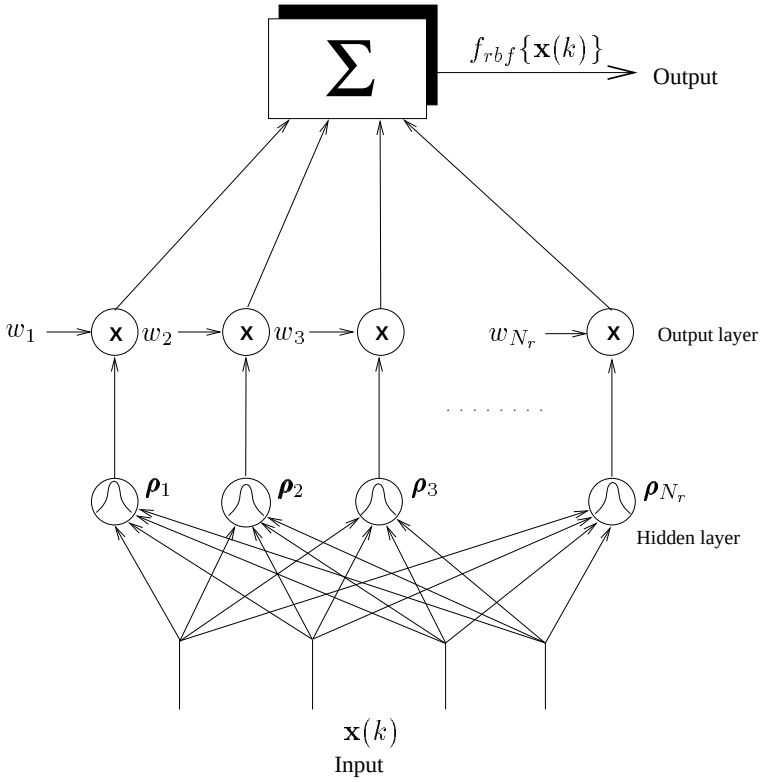


Figure 2.13: *A radial basis function network for signal processing applications*

multidimensional output as well. Possible choices for the radial basis function $\phi(\gamma)$ include a thin plate spline,

$$\phi(\gamma) = \frac{\gamma}{\sigma_r^2} \log\left(\frac{\gamma}{\sigma_r}\right) \tag{2.36}$$

a multi quadratic,

$$\phi(\gamma) = \sqrt{\gamma^2 + \sigma_r^2} \tag{2.37}$$

an inverse multi-quadratic,

$$\phi(\gamma) = \frac{1}{\sqrt{\gamma^2 + \sigma_r^2}} \tag{2.38}$$

and Gaussian kernel,

$$\phi(\gamma) = \exp\left(-\frac{\gamma^2}{2\sigma_r^2}\right) \tag{2.39}$$

Here, the parameter σ_r^2 controls the radius of influence of each basis functions and determines how rapidly the function approaches 0 with γ. The Gaussian and the inverse multi-quadratic kernel provide bounded and localised properties such that $\phi(\gamma) \rightarrow 0$ as $\gamma \rightarrow \infty$. Broomhead and Lowe [8] reinterpreted the RBF network as a least square estimator which led to its wide spread use in signal processing applications such as time series prediction [26, 66], system identification[67, 68], interference cancellation[69], radar signal processing[70], pattern classification[71] and channel equalisation[27, 72]. In signal processing applications the RBF inputs are presented through a TDL. Training of the RBF networks involves setting the parameters for the centres $\boldsymbol{\rho}_i$, spread σ_r and the linear weights w_i. The RBF networks are easy to train since the training of centres, spread parameter and the weights can be done sequentially and the network offers a nonlinear mapping, maintaining its linearity in parameter structure at the output layer. One of the most popular schemes employed for training the RBF in a supervised manner is to estimate the centres using a clustering algorithm like the κ-means clustering and setting σ_r^2 to an estimate of input noise variance calculated from the centre estimation error. The output layer weights can be trained using popular stochastic gradient LMS algorithm. Other schemes for RBF training involve selecting a large number of candidate centres initially and use the orthogonal least squares (OLS) [26] algorithm to pick a subset of the centres that provides near optimal performance. The MLP back propagation algorithm can also be used[72] to train the RBF centres.

In early RBF equalisers [27] the RBF centres were selected at random, picked from a few of the

initial input vectors. The weights were updated using supervised training by the LMS algorithm or its momentum version [73]. This resulted in equalisers with large number of centres making the network computationally complex. Chen proposed the OLS algorithm [26, 74] for selecting an optimum number of centres from a large number of candidate centres, resulting in near optimal performance. Subsequently, the close relationship between the RBF network and the Bayesian equaliser was found [57] and this provided the parametric implementation of the Bayesian equalisers with the RBF. In these equalisers supervised κ-means clustering [68, 75] provides the estimate of the centres while linear weights are estimated using the LMS algorithm. With the development of RBFs that could handle complex signals [31], they were used for equalisation in communication systems with complex signal constellation [32]. Cha proposed the stochastic gradient algorithm [76] to adapt all the RBF parameters and used this technique to equalise 4-QAM digital communication systems.

A deeper examination of the RBF decision function in (2.35), in conjunction with a Gaussian kernel (2.39), and the Bayesian equaliser decision function in (2.30) shows that both of these functions are similar. The RBF network can provide a Bayesian decision function by setting the RBF centres, $\boldsymbol{\rho}_i$, to channel states, $\mathbf{c}_i$, RBF spread parameter, σ_r^2, to channel noise variance, σ_η^2, and the linear weights $w_i = +1$ if $\mathbf{c}_i \in \mathbf{C}_d^+$ and $w_i = -1$ if $\mathbf{c}_i \in \mathbf{C}_d^-$. This provides the optimum RBF network as an equaliser. In this implementation the channel state vectors $\mathbf{c}_i$ can be estimated using supervised κ-means clustering or alternatively they can be calculated from an estimate of the channel.

The RBF equaliser can provide optimal performance with small training sequences but they suffer from computational complexity. The number of RBF centres required in the equaliser increases exponentially with equaliser order and the channel delay dispersion order. This increases all the computations exponentially. Some of these issues have been discussed in [36, 77]. In a varied implementation [78] the RBF with scalar centres results in a reduction of computational complexity. The issues relating to the RBF equaliser design have been discussed extensively in [30].

2.7.2 Neural network equalisers

Neural networks are nonlinear processing elements like biological neurons and possess universal approximation capabilities [79]. One of the popular forms of neural networks used in signal processing applications is the MLP. The basic building block of a MLP is a neuron presented

in Figure 2.14(a). The neuron receives an m-dimensional real input vector $\mathbf{x}(k) \in \mathbb{R}^m$ and computes a weighted sum with its weight vector $\mathbf{w} = [w_0, w_1, \ldots, w_{m-1}]^T$, and adds a threshold weight ξ. The resulting output is passed through a node activation function φ. The most popular form of this activation function is the sigmoid nonlinearity given by

$$\varphi(y) = \frac{1 - e^{-y}}{1 + e^{-y}} \tag{2.40}$$

The transfer characteristic of this sigmoidal nonlinearity is presented in Figure 2.14(b).

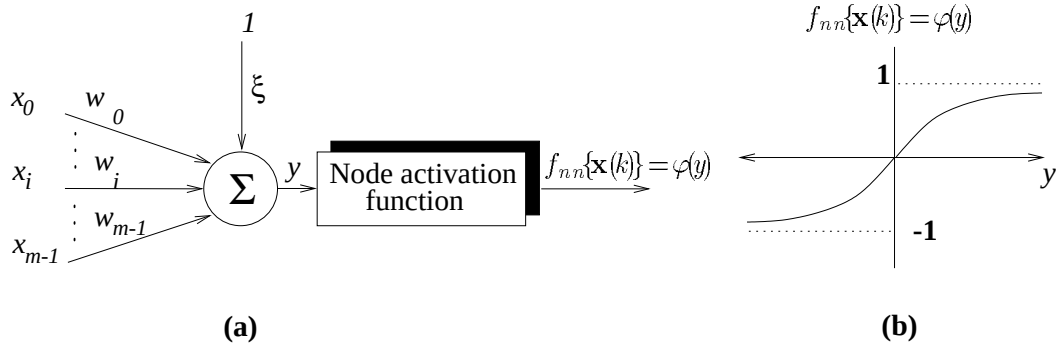


Figure 2.14: *Structure of a neuron*

An MLP constitute a number of processing neurons organised in layers. All the neurons in a layer are fully connected to the neurons in the preceding and succeeding layers. There is no interconnection among the neurons in the same layer. There is also no interconnection between the neurons in layers beyond the preceding and the succeeding layers in an MLP. In equalisation applications input to the MLP is presented through a set of tapped delay lines and the output layer has a single neuron. The structure of a MLP for this is presented in Figure 2.15. The m-dimensional received signal vector $\mathbf{r}(k) = [r(k), r(k-1), \ldots, r(k-m+1)]^T$ forms the input to the MLP. The equaliser consists of n layers of neurons with N_1 to N_n neurons in each layer and $N_n = 1$. The network output is passed through a hard limiter to determine the estimated signal $\widehat{s}(k-d)$. A two layer neural network is sufficient to model any nonlinear system but the number of elements needed for this two layer network may be large [79]. For this reason a three layers MLP should provide reasonable performance with relatively smaller number of elements.

Training an MLP equaliser involves estimating proper weights and thresholds. The MLP equaliser can be trained in a supervised manner using the back propagation (BP) [80] algorithm. Siu et. al.[23] developed MLP equalisers with decision feedback and showed that this equaliser

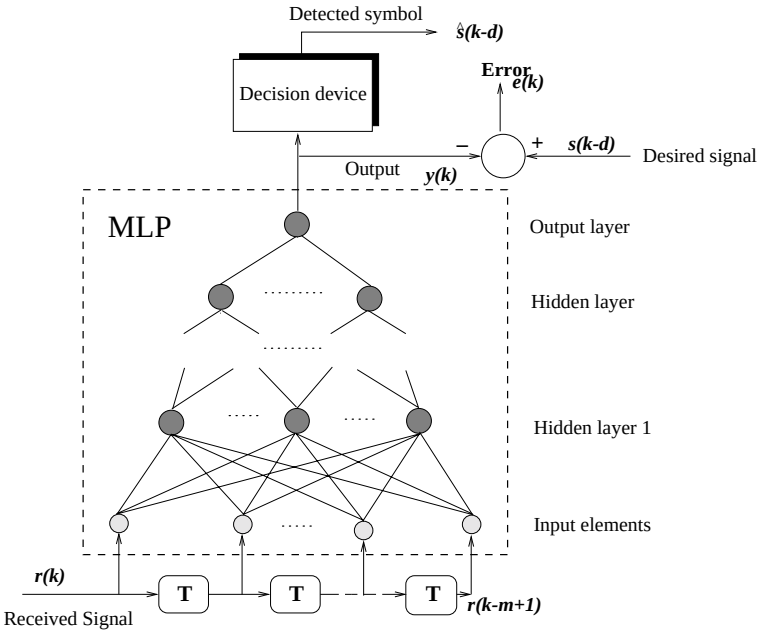


Figure 2.15: *An MLP equaliser*

could provide better performance than linear equalisers with decision feedback. On similar lines Gibson et. al. also proposed an MLP equaliser [24]. Subsequently MLP equalisers have been developed for equalisation of a number of systems. Some examples are channels with nonlinear distortion [81], quadrature phase shift keying (QPSK) [82] communication systems, satellite channels with nonlinear distortion with MQAM [83] signal constellation, satellite radio channels [84], indoor radio channels [85], combined equalisation and decoding [86], fibre optic communication systems [44] and data storage equalisation [87]. Chen et. al. [53] showed that MLP equalisers can provide the nonlinear decision boundary associated with the MAP equaliser. MLP has also been used for co-channel interference suppression[88]. In spite of its good performance, MLPs have raised many controversial issues that need to be addressed. Some of these are as under.

- There has been very little understanding on the relationship between the network architecture and the communication problem. Hence the networks turn out to be very bulky.
- The high degree of nonlinearity of MLPs makes their theoretical analysis of the performance with respect to adaptation parameters difficult, and hence training parameters are generally selected by trial and error.
- No relationship has been derived between the MLP and the optimal Bayesian equalisers.

- The equaliser training starts with random weight initialisation and there is no method guaranteeing proper weight convergence.
- The BP algorithm optimises the weights with the MMSE criterion and also require long training time. The optimum equalisation criteria is based on minimum error probability which is different from MMSE criteria.
- The computational complexity of the MLP is large.

Attempts have been made to address some of these issues in recent years[29]. The development of fast training schemes based on Kalman filters [89] and other least squares (LS) training schemes [90] provides better convergence at the cost of computational complexity. Training schemes to optimise minimum BER of neural network equalisers using fuzzy decision learning have also been developed [64]. Algorithms for training ANN equalisers to achieve MLSE performance with minimum BER criterion involving conditional distributed learning [37], Hopfield networks with mean field annealing [91], cellular neural networks with hardware annealing [92–94] have shown better equaliser performance. A single layer neural network can provide nonlinear mapping if sufficient order of nonlinearity is incorporated in the input [95]. With this a number of neural equalisers using single layer architecture with polynomial perceptron [96, 97], functional link perceptron [63, 88, 98–100], polynomial lattice equalisers [101] and perceptron equalisers with multilevel sigmoidal perceptron [102] have been developed. Some of the issues relating to the design of MLP structure for equalisation applications have been addressed in [38]. A review of neural network techniques for equalisation problem is presented in [28, 29].

2.7.3 Fuzzy and neuro fuzzy equalisers

Fuzzy systems or fuzzy logic[6] system is the name for systems which have a direct relationship with fuzzy concepts(like fuzzy sets, linguistic variables) and fuzzy logic [103, 104]. The basic building block of a fuzzy logic system is presented in Figure 2.16. Here the fuzzifier converts the real world crisp input sample $x_i(k)$ to a fuzzy output F_i^l described by the membership function ψ_i^l. This provides the degree to which the the input scalar $x_i(k)$ belongs to the fuzzy set F_i^l. The inference engine provides the relationship between the fuzzy input in terms of

[6] In the literature it is also commonly referred to as fuzzy logic controller

membership functions and the fuzzy output of the controller using a set of IF ... THEN ... rules derived from the rule base. The rule l in the fuzzy rule base can be defined as

$$R^{(l)} : \text{IF} \quad x_1 \text{ is } F_1^l \quad \text{and} \quad ... \quad \text{and} \quad x_n \text{ is } F_n^l \quad \text{THEN} \quad y \text{ is } G^l \tag{2.41}$$

The defuzzifier converts the inferences G^l to provide the crisp output $y(k)$. Generally in a fuzzy system the rule base is generated in advance with expert knowledge of the system under consideration. However, recently [105] online learning properties have been introduced which provide scope for training. This feature in fuzzy systems is achieved with the adaptation and learning block that uses the available information in the system. The available linguistic rules can also be applied in the adaptation algorithm. These types of systems are also called adaptive neuro fuzzy filters(ANFF)[12] and they possesses the ability to incorporate training like neural networks and can also use rule bases from human experts as in fuzzy systems. The adaptive fuzzy systems have been applied to a variety of engineering applications[106] such as medical diagnostics, image processing, pattern classification [107, 108], clustering [109] control applications [110–112] and time series forecasting [113] etc. Wang et. al. [114] presented fuzzy

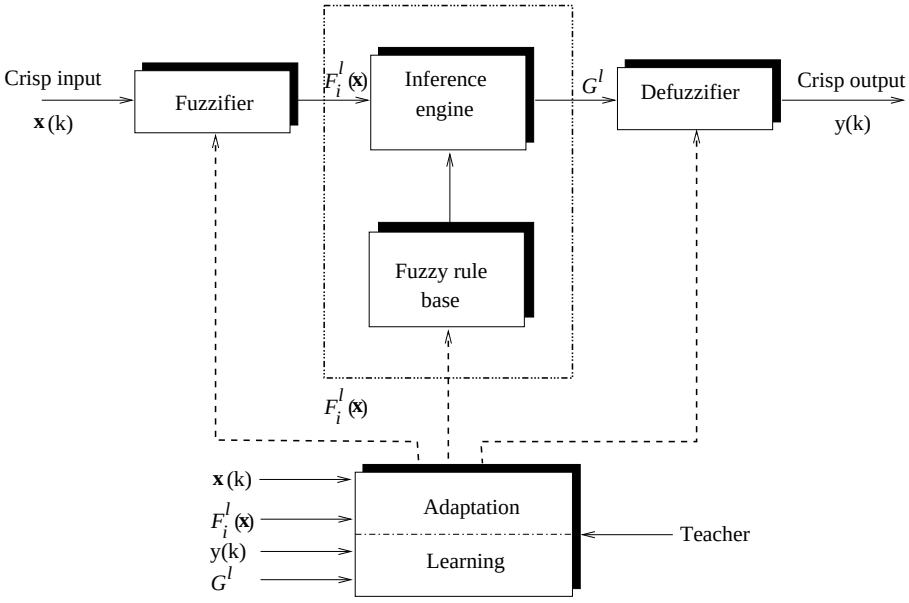


Figure 2.16: *A typical fuzzy logic system*

basis functions (FBF) and used a combination of these functions for universal approximation and later on used them as a fuzzy filter [11] for channel equalisation. Based on these concepts other fuzzy filter based equalisers were developed for different applications [115–119].

Other forms of fuzzy equalisers were presented in [120]. Nie[121] proposed a learning algorithm to reduce the number of rules used in the equaliser proposed in [11]. Gan [122, 123] proposed fuzzy techniques for the adjustment of the step size in the LMS algorithm and a similar technique was used [124] for step size adjustment of LMS algorithm for equalisation of high definition television (HDTV) systems. Lin and Juang [12, 125] developed the ANFFs and used it for equalisation and noise reduction. This ANFF constructs its rule base in a dynamic way with the training samples. These ANFF provide scope to design nonlinear filters that are computationally simple and can accept linguistic variables from expert systems.

Most of the fuzzy equalisers developed in the recent years have structures similar to the LMS or RLS fuzzy filters proposed in [11]. An equaliser based on fuzzy RLS filters is computationally complex and the rule base grows exponentially with the number of rules in each dimension. On the other hand the LMS filter, though computationally simpler than its counterpart, suffers from performance degradation if initial parameters are not selected properly. This thesis presents the development of similar forms of fuzzy equaliser that alleviates the problems associated with fuzzy equalisers in [11] and subsequently a modified form of this filter is designed for CCI mitigation.

2.8 Conclusion

In this chapter the optimum symbol-by-symbol equaliser decision function was derived and its implementation using the RBF was presented. Other forms of nonlinear equalisers using the ANN and fuzzy techniques have also been introduced. The fuzzy equalisers and ANFF introduced here are used in subsequent chapters for deriving the fuzzy implementation of the Bayesian equaliser. The concept of CCI was also introduced in this chapter. The equalisation of CCI channels using fuzzy filters is discussed in Chapter 4.

Chapter 3
Fuzzy Implementation of Bayesian Equalisers

3.1 Introduction

Channel equalisation is a nonlinear classification problem. Even when the channel is linear, the equalisation problem is still a nonlinear one. This was shown in Chapter 2. Under many circumstances the nonlinear decision boundary can be approximated by a linear boundary. This is the best performance a linear equaliser can provide and therefore it suffers from performance degradation. Owing to this suboptimal performance of linear equalisers, it is always desirable to explore new nonlinear equalisation algorithms that can provide a performance trade off with computational complexity against the optimal MAP Bayesian equaliser.

This chapter discusses the development of a new fuzzy nonlinear equaliser which can be considered as a fuzzy implementation of the Bayesian equaliser. The chapter addresses the issues described in following steps:

- The fuzzy implementation of the Bayesian equaliser is derived and its performance is evaluated and compared with the optimal Bayesian equaliser using BER as the performance criterion.
- Computational complexity issues of the fuzzy equaliser are presented.
- The concept of subset state selection in the fuzzy implemented Bayesian equaliser is presented.

The chapter organisation is as follows. Following this, section 3.2 introduces the design of a fuzzy adaptive filter. Section 3.3 develops the normalised form of Bayesian equaliser[1] with scalar channel states. Sections 3.4 develops the fuzzy equaliser design, while section 3.5 and

[1] The decision function for the Bayesian equaliser was presented in section 2.5

3.6 discuss its training and computational complexity issues respectively. Some simulation results are presented in section 3.7. The chapter ends with the concluding remarks.

3.2 Fuzzy adaptive filter and LMS algorithm

The fuzzy adaptive filter (FAF) was originally proposed by Wang and Mendel [11]. Fuzzy filters are nonlinear filters that can incorporate fuzzy IF ... THEN ... rules from a human expert system. Wang and Mendel had proposed two types of fuzzy filters [11], the RLS fuzzy filter and the LMS fuzzy filter. The fuzzy filter presented in this thesis has a structure similar to the RLS filter proposed in [11] and the equaliser is trained with the LMS algorithm.

The filter considered here maps a real input vector $\mathbb{R}^m \rightarrow \mathbb{R}$ with the function

$$f_{faf}\{\mathbf{x}(k)\} : U \subset \mathbb{R}^m \rightarrow \mathbb{R} \tag{3.1}$$

where $\mathbf{x}(k) = [x_1(k),\ x_2(k),\ \ldots,\ x_i(k),\ \ldots,\ x_m(k)]^T$, $x_i(k) \in U \equiv [g_i^-, g_i^+]$ is the input to the fuzzy filter and g_i^- , g_i^+ are the minimum and maximum limits for the input scalars $x_i(k)$. Here $f_{faf}\{\mathbf{x}(k)\}$ is the FAF output, corresponding to the filter input $\mathbf{x}(i)$. The filter minimises the sum squared error performance index such that

$$e(k) = \sum_{i=0}^{k} [y(i) - f_{faf}\{\mathbf{x}(i)\}]^2 \tag{3.2}$$

where $y(i)$ is the desired filter output corresponding to the filter input $\mathbf{x}(i)$ and $e(k)$ is the sum of the error squares that needs to be minimised.

3.2.1 Filter design

A filter with an input vector of length m having a scalar output is considered. Each element of the filter input is fuzzified with a Gaussian membership function. The membership function for the inputs can be represented as

$$\psi_i^j(k) = \exp\left\{-\frac{1}{2}\left(\frac{x_i(k) - \delta_i^j}{\sigma_i^j}\right)^2\right\} \tag{3.3}$$

where δ_i^j and σ_i^j are the jth centre and spread parameters respectively corresponding to input scalar x_i, $1 \leq i \leq m$ such that the input space $x_i \in U \equiv [g_i^-, g_i^+]$ is completely covered. These parameters once selected remain fixed and the input x_i is associated with the membership functions $\psi_i^1,\ \psi_i^2,\ \ldots,\ \psi_i^{M_i}$, so that the filter is characterised by a total of $\sum_{i=1}^{m} M_i$ membership functions. The filter consists of fuzzy IF $\ldots$ THEN $\ldots$ rules of the form

$R^{(1,1,\ldots,1)}$: IF x_1 is F_1^1 x_2 is F_2^1 $\ldots$ x_m is F_m^1 THEN y is $\psi_1^1 \psi_2^1 \ldots \psi_m^1$

$\ldots$

$R^{(1,1,\ldots,M_m)}$: IF x_1 is F_1^1 x_2 is F_2^1 $\ldots$ x_m is $F_m^{M_m}$ THEN y is $\psi_1^1 \psi_2^1 \ldots \psi_m^{M_m}$

$\ldots$

$R^{i1,i2,\ldots,im}$: IF x_1 is F_1^{i1} x_2 is F_2^{i2} $\ldots$ x_m is F_m^{im} THEN y is $\psi_1^{i1} \psi_2^{i2} \ldots \psi_m^{im}$

$\ldots$

$R^{M_1,M_2,\ldots,M_m}$: IF x_1 is $F_1^{M_1}$ x_2 is $F_2^{M_2}$ $\ldots$ x_m is $F_m^{M_m}$ THEN y is $\psi_1^{M_1} \psi_2^{M_2} \ldots \psi_m^{M_m}$

where each of the terms $i1,\ i2,\ \ldots,\ im$ are single indices each ranging from 1 to M_i respectively. The filter considered here finds the following nonlinear function of the membership functions ψ_i^j so that,

$$f_{faf}\{\mathbf{x}(k)\} = \frac{\sum_{i1=1}^{M_1} \sum_{i2=1}^{M_2} \ldots \sum_{im=1}^{M_m} \vartheta_l(k)^{(i1,i2,\ldots,im)} \left\{\psi_1^{i1}(k)\ \psi_2^{i2}(k)\ \ldots\ \psi_m^{im}(k)\right\}}{\sum_{i1=1}^{M_1} \sum_{i2=1}^{M_2} \ldots \sum_{im=1}^{M_m} \left\{\psi_1^{i1}(k)\ \psi_2^{i2}(k)\ \ldots\ \psi_m^{im}(k)\right\}} \tag{3.4}$$

where $\vartheta(k)^{(i1,\,i2,\,\ldots,\,im)}$ is the weight associated with the fuzzy IF $\ldots$ THEN $\ldots$ rule $R^{i1,\,i2,\,\ldots,\,im}$.

The weight parameter $\vartheta(k)^{(i1,i2,\ldots,im)}$ is updated during the adaptation procedure so as to minimise the desired cost function in (3.2). Using the LMS algorithm to update the filter parameter $\vartheta k^{(i1,i2,\ldots,im)}$,

$$\vartheta(k+1)^{(i1,i2,\ldots,im)} = \vartheta(k)^{(i1,i2,\ldots,im)} + \varrho\,[y(k) - f_{faf}\{\mathbf{x}(k)\}]\,\mathbf{\Psi}\{\mathbf{x}(k)\}^{(i1,i2,\ldots,im)} \tag{3.5}$$

where,

$$\mathbf{\Psi}\{\mathbf{x}(k)\}^{(i1,i2,\ldots,im)} = \frac{\psi_1^{i1}\,\psi_2^{i2}\,\ldots\,\psi_m^{im}}{\sum_{j1=1}^{M_1}\sum_{j2=1}^{M_2}\cdots\sum_{jm=1}^{M_m}\left[\psi_1^{j1}\,\psi_2^{j2}\,\ldots\,\psi_m^{jm}\right]} \tag{3.6}$$

Here, $\mathbf{\Psi}(\mathbf{x})^{(i1,i2,\ldots,im)}$ is the input to the filter weight $\vartheta^{(i1,i2,\ldots,im)}$, ϱ is the learning rate and $j1, j2, \ldots, jm$ constitute single indices. The filter function in (3.4) finds a weighted sum of all possible combinations of the products of the membership functions, taking one from each input, and this sum is scaled with the sum of all possible product combinations of the membership functions taking one from each input. Since the membership functions are Gaussian in nature the term in the denominator of the filter function will be non-zero, making the filter realisable. Here it can be seen that the term $\mathbf{\Psi}\{\mathbf{x}(k)\}^{(i1,i2,\ldots,im)}$ is a FBF [114] with singleton fuzzifier, Gaussian membership function, product inference and centre of gravity (COG) defuzzifier. A combination of these basis functions can be used for universal approximation [114]. With the use of different types of membership functions, inference rules and defuzzification processes a variety of fuzzy filters can be designed to optimise any arbitrary function. Each of the FBF's works as a fuzzy rule and the FAF consist of

$$N_c = \prod_{i=1}^{m} M_i \tag{3.7}$$

fuzzy rules.

The effect of normalisation in FBF provides characterisation of local and global properties. It is well established in neural literature [126, 127] that the Gaussian RBF is good at characterising local properties and that the neural networks with sigmoid nonlinearities are good at characterising global properties. The fuzzy filter designed in this section will have the capabilities to optimise both local and global properties. The relationship of the FBF with other form of basis functions like RBF and PNN have been discussed in [128, 129].

3.3 Normalised Bayesian equaliser with scalar states (NBESS)

The communication system discussed in this chapter was presented in Figure 2.8. This communication system is again presented here in Figure 3.1 for convenience. The equaliser is

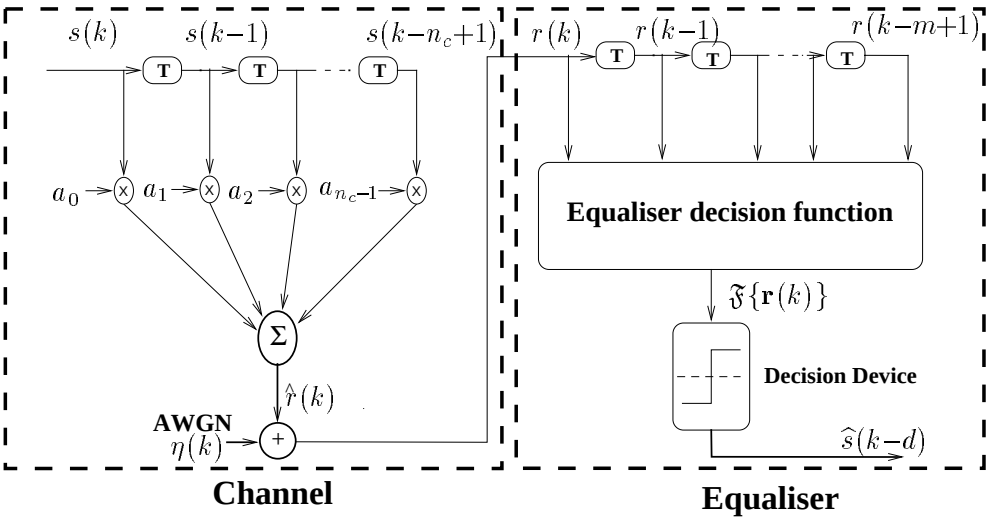


Figure 3.1: *Discrete time model of a digital communication system*

characterised by its feed forward length m and decision delay d and it does not use decision feedback. The decision function of the T-spaced symbol-by-symbol Bayesian equaliser can be represented as[2].

$$\mathfrak{F}\left\{\mathbf{r}(k)\right\} = \sum_{i=1}^{N_s} w_i \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_i\|^2}{2\sigma_\eta^2}\right) \tag{3.8}$$

where N_s is the number of channel states, equal to 2^{n_c+m-1} while w_i are the weights associated with each of the channel states and $w_i = +1$ if $\mathbf{c}_i \in \mathbf{C}_d^+$ and $w_i = -1$ if $\mathbf{c}_i \in \mathbf{C}_d^-$. It is also observed that each of the channel states has m components which can be represented as $\mathbf{c}_i = \left[c_{i0},\ c_{i1},\ c_{i2},\ \ldots,\ c_{i(m-1)}\right]^T \in \mathbb{R}^m$. This Bayesian equaliser presented in (3.8) can be implemented with RBF networks [57]. In line with the normalised RBF proposed by Cha et.al. [69], a normalised Bayesian equaliser, which estimates the transmitted symbols themselves rather than the decision function can be formed. This equaliser is represented as a normalised Bayesian equaliser,

$$\mathfrak{F}_{NBAY}\left\{\mathbf{r}(k)\right\} = \frac{\displaystyle\sum_{i=1}^{N_s} w_i \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_i\|^2}{2\sigma_\eta^2}\right)}{\displaystyle\sum_{i=1}^{N_s} \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_i\|^2}{2\sigma_\eta^2}\right)} \tag{3.9}$$

[2]This equaliser decision function was derived in Section 2.5

where (3.9) is the decision function of the Bayesian equaliser in (3.8) which has been normalised with the sum of the output of all the basis functions given by $\exp\left(\frac{-\|\mathbf{r}(k)-\mathbf{c}_i\|^2}{2\sigma_\eta^2}\right)$.

3.3.1 Effects of normalisation

The equaliser presented in (3.8) can be implemented with a RBF network [57] and the normalised form of this in (3.9) can be implemented with a normalised RBF [69]. The effect of normalisation in RBF networks has been analysed in [130]. The application of the normalised RBF to Channel equalisation application is similar to interference cancellation discussed in [69]. The following characteristics of this problem makes the decision function immune to the ill effects of normalisation [130]:

- In equalisation applications, the network decision function $\mathfrak{F}_{NBAY}\{\mathbf{r}(k)\}$ (3.9) is passed through a memoryless detector to recover the transmitted symbol $\widehat{s}(k-d)$ which has a discrete constellation. The sign of the decision function output is enough to provide the final decision.
- The decision boundary corresponds to the locus of points in the decision surface for which $\mathfrak{F}_{NBAY}\{\mathbf{r}(k)\}=0$, and this does not change with normalisation since the spread associated with each of the channel states or RBF centres is uniform and equal to σ_η^2.

An example is considered below to show the effects of normalisation in Bayesian equaliser decision function.

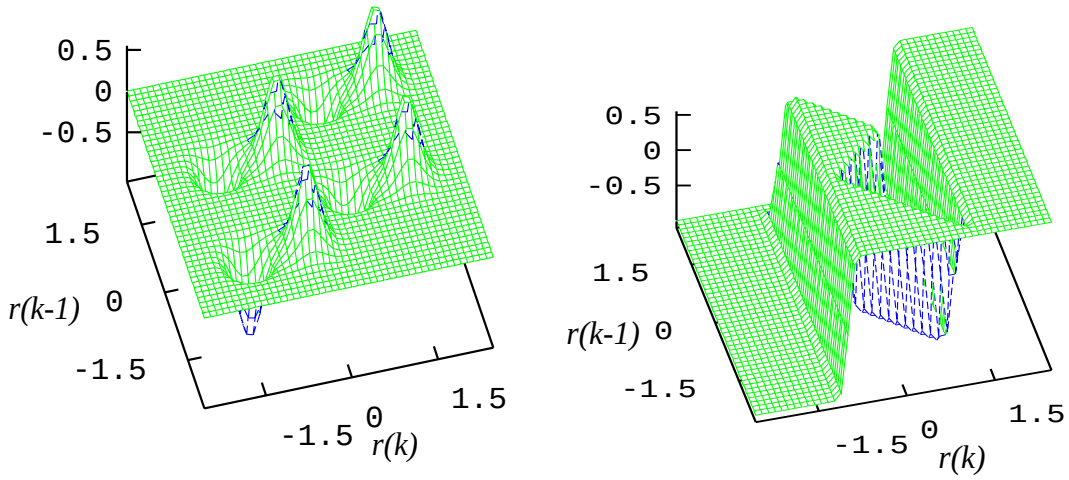
EXAMPLE 3.1

The channel and equaliser order considered in the example is same as considered in Example 2.1. Here

$$H(z)=H_1(z)=0.5+1.0z^{-1} \quad \text{with } m=2, d=0 \text{ and SNR} = 15\,\text{dB} \tag{3.10}$$

The channel states for the equalisers have been presented in Table 2.1.

The decision function provided by the Bayesian equaliser and the normalised Bayesian equaliser are presented in Figure 3.2(a) and 3.2(b) respectively. From Figure 3.2(a) it is seen that the Bayesian equaliser decision function has 4 peaks corresponding to channel states $\mathbf{c}_j \in \mathbf{C}_d^+$, $1 \le j \le 4$ and 4 valleys corresponding to $\mathbf{c}_j \in \mathbf{C}_d^-$, $5 \le j \le 8$.



(a) Bayesian Equaliser

(b) Normalised Bayesian Equaliser

Figure 3.2: *Effect of normalisation in Bayesian equaliser decision function with* $H(z) = 0.5 + 1.0z^{-1}$, $m = 2$, $d = 0$ *and SNR=15 dB*

When the equaliser input is far from all of the channel states, the decision function due to contributions from all the channel states is nearly equal and approaches 0. The decision function of the normalised Bayesian equaliser provides only 2 discrete outputs providing a decision of $+1/-1$ corresponding to the transmitted sample $s(k)$. It also provides a nonzero output of $+1/-1$ when the input is far from all the channel states.

The decision function of the Bayesian equaliser in (3.8) and (3.9) needs the channel states. The channel states can be estimated during the training period. The equaliser decision function in (3.9) reveals that the equaliser contains N_s channel states, each of m dimensions. The number of scalar channel states for any channel is $M = 2^{n_c}$. Each of the m components of the N_s channel states are taken from the set of M scalar channel states which form the estimate of noise free received scalars. Rewriting the squared norm of the $\exp(.)$ in (3.9) as a summation

and exploiting the properties of the $\exp(.)$ function yields:

$$\mathfrak{F}_{NBESS}\left\{\mathbf{r}(k)\right\} = \frac{\sum_{i=1}^{N_s} w_i \left\{ \prod_{l=0}^{m-1} \exp\left(-\frac{|r(k-l) - c_{il}|^2}{2\sigma_\eta^2} \right) \right\}}{\sum_{i=1}^{N_s} \left\{ \prod_{l=0}^{m-1} \exp\left(-\frac{|r(k-l) - c_{il}|^2}{2\sigma_\eta^2} \right) \right\}} \tag{3.11}$$

where c_{il} is the $(l+1)$ component of channel state $\mathbf{c}_i$, corresponding to the input scalar $r(k-l)$ and the Euclidean distance $\|.\|$ has been replaced by the absolute distance $|.|$ since the arguments used are scalars. This equaliser is termed as the NBESS.

Equations (3.9) and (3.11) provide alternative realisations of the Bayesian equaliser decision function. In (3.9) the Euclidean distance between the input vector $\mathbf{r}(k)$ and each of the channel states $\mathbf{c}_i$ is first calculated. The result is then scaled by $-1/(2\sigma_\eta^2)$ and the exponential function is evaluated. These are linearly combined to provide the decision function. Alternatively in (3.11), the square of scalar distances are first calculated, scaled by $-1/(2\sigma_\eta^2)$ and the exponential function evaluated. The exponential functions associated with particular channel states are linearly combined to form the channel states output. These are linearly combined with associated weights to provide the equaliser decision function. Both of these functions require the knowledge of channel states for estimating the decision function. It was noted in [131] that (3.11) may be preferable to (3.9) for implementation. This approach is adopted here.

Each of the components c_{il} of channel states $\mathbf{c}_i$ is taken from the scalar channel states C_j, $1 \leq i \leq M$. This relationship between the channel states and the scalar channel states can be represented as

$$c_{il} \in C_j \quad \text{with } 1 \leq i \leq N_s,\ 0 \leq l \leq m \text{ and } 1 \leq j \leq M \tag{3.12}$$

and is described in the following example.

EXAMPLE 3.2

The process of the generation of channel states from the scalar channel states is presented here in this example. The channel considered here is

$$H(z) = H_2(z) = 1.0 + 0.2z^{-1} \tag{3.13}$$

The equaliser with length $m = 3$ and decision delay $d = 0$ is considered. The optimal equaliser for this system has $N_s = 2^{n_c+m-1} = 16$ channel states and $M = 2^{n_c} = 4$ scalar channel states. These scalar channel states are presented in Table 3.1. The channel states for the equaliser along with the combination of scalar channel states that form the channel states are presented in Table 3.2.

j	C_j	$s(k)$	$s(k-1)$	$\widehat{r}(k)$
1	C_1	1	1	1.2
2	C_2	1	-1	0.8
3	C_3	-1	1	-0.8
4	C_4	-1	-1	-1.2

Table 3.1: *The scalar channel state calculation for channel* $H_2(z) = 1.0 + 0.2z^{-1}$, $M = 4$

Here each of the channel states is a vector of order 3. Each of the components of the $N_s = 16$ channel states is taken from the scalar channel states presented in Table 3.1. From Table 3.2 it can be seen that estimation of the scalar channel states only can provide the channel state for the equaliser in (3.11).

$$\mathbf{c}_i \in \begin{cases} \mathbf{C}_d^+ & \text{if} \quad 1 \leq i \leq \dfrac{N_s}{2} \\ \mathbf{C}_d^- & \text{if} \quad \dfrac{N_s}{2} + 1 \leq i \leq N_s \end{cases} \tag{3.14}$$

With the knowledge of the scalar channel states and the signal vector $\mathbf{s}(k)$ generating the scalar channel states, the channel states can be estimated and the equaliser in (3.11) can be constructed. This equaliser can be implemented using a normalised RBF with scalar centres[78].

With this understanding of process of the the formation of the channel states from scalar channel states, the NBESS equaliser decision function in (3.11)can be represented as:

$$\mathfrak{F}_{NBESS}\{\mathbf{r}(k)\} = \frac{\displaystyle\sum_{i=1}^{N_s} w_i \left\{ \prod_{l=0}^{m-1} \phi_{il} \right\}}{\displaystyle\sum_{i=1}^{N_s} \left\{ \prod_{l=0}^{m-1} \phi_{il} \right\}} \tag{3.15}$$

No.	$s(k)$	$s(k-1)$	$s(k-2)$	$s(k-3)$	$\mathbf{c}_i$	$\widehat{r}(k)$ c_{i0}	$\widehat{r}(k-1)$ c_{i1}	$\widehat{r}(k-2)$ c_{i2}	c_{i0}	c_{i1}	c_{i2}
1	1	1	1	1	$\mathbf{c}_1$	1.2	1.2	1.2	C_1	C_1	C_1
2	1	1	1	−1	$\mathbf{c}_2$	1.2	1.2	0.8	C_1	C_1	C_2
3	1	1	−1	1	$\mathbf{c}_3$	1.2	0.8	−0.8	C_1	C_2	C_3
4	1	1	−1	−1	$\mathbf{c}_4$	1.2	0.8	−1.2	C_1	C_2	C_4
5	1	−1	1	1	$\mathbf{c}_5$	0.8	−0.8	1.2	C_2	C_3	C_1
6	1	−1	1	−1	$\mathbf{c}_6$	0.8	−0.8	0.8	C_2	C_3	C_2
7	1	−1	−1	1	$\mathbf{c}_7$	0.8	−1.2	−0.8	C_2	C_4	C_3
8	1	−1	−1	−1	$\mathbf{c}_8$	0.8	−1.2	−1.2	C_2	C_4	C_4
9	1	1	1	1	$\mathbf{c}_9$	−0.8	1.2	1.2	C_3	C_1	C_1
10	−1	1	1	−1	$\mathbf{c}_{10}$	−0.8	1.2	0.8	C_3	C_1	C_2
11	−1	1	−1	1	$\mathbf{c}_{11}$	−0.8	0.8	−0.8	C_3	C_2	C_3
12	−1	1	−1	−1	$\mathbf{c}_{12}$	−0.8	0.8	−1.2	C_3	C_2	C_4
13	−1	−1	1	1	$\mathbf{c}_{13}$	−1.2	−0.8	1.2	C_4	C_3	C_1
14	−1	−1	1	−1	$\mathbf{c}_{14}$	−1.2	−0.8	0.8	C_4	C_3	C_2
15	−1	−1	−1	1	$\mathbf{c}_{15}$	−1.2	−1.2	−0.8	C_4	C_4	C_3
16	−1	−1	−1	−1	$\mathbf{c}_{16}$	−1.2	−1.2	−1.2	C_4	C_4	C_4

Table 3.2: *The channel states calculation for channel* $H_2(z) = 1.0 + 0.2z^{-1}$ *with* $m = 3$, $d = 0$, $N_s = 16$ *and* $M = 4$

where ϕ_{il} is a basis function of the form

$$\phi_{il} = \exp\left[-\frac{1}{2}\left\{\frac{|r(k-l) - c_{il}|^2}{\sigma_\eta^2}\right\}\right] \tag{3.16}$$

generated from the $(l+1)$ scalar components of the channel states $\mathbf{c}_i$, corresponding to the input scalar $r(k-l)$, $0 \leq l \leq (m-1)$. In (3.15) computation of $\prod_{l=0}^{m-1} \phi_{il}$ is the same as the computation of $\exp\left(\frac{-\|\mathbf{r}(k)-\mathbf{c}_i\|^2}{2\sigma_\eta^2}\right)$ in (3.9). Here the equaliser decision function presented in (3.15) can also be considered as a linear combination of nonlinear basis functions like the RBF and the FBF.

3.4 Fuzzy implementation of Bayesian equaliser

The FAF presented in Section 3.2, was proposed by Wang and Mendel [11]. This filter in conjunction with the RLS training algorithm, was used for equalisation. For equalisation the number of fuzzy sets M_i for each input are set equal so that $M_1 = M_2 = \cdots = M_m = M$. In [11] the membership function centres δ_i^j, $0 \leq j \leq M$, of the FAF were selected uniformly in the signal space $[g_i^+, g_i^-]$ and the spread parameter σ_i^j associated with each of the

membership functions were set to arbitrary uniform values < 1. The number of fuzzy sets M corresponding to each of the inputs were selected to provide good performance. This scheme of using the same set of membership function centres for each of the dimensions of the signal positions the FBF's in a regular grid in the multidimensional space $\mathbb{R}^m$. The use of a large number of basis functions made the equaliser complex and the RLS training scheme increased the complexity of the equaliser during training. Based on this idea Lee [115] proposed a fuzzy decision feedback equaliser where the fuzzy equaliser centres were positioned at scalar channel states, and the equaliser used a subset of the available M^m FBFs depending on the state of the feedback vector $\widehat{\mathbf{s}}_f(k) = [\widehat{s}(k-d-1),\ \widehat{s}(k-d-2),\ \ldots,\ \widehat{s}(k-d-q)]^T$, where q is the feedback order. This process of using a subset of the N_c basis functions reduces computational complexity. Later, complex fuzzy filters with a similar architecture were used in a variety of equalisation applications [116, 117]. All these equalisers used M^m FBFs working as fuzzy IF ... THEN ... rules. In this form the complexity of the equalisers is related exponentially to the number of scalar channel states. The scalar channel states are exponentially related to the signal constellation and channel length. This accounted for the high computational complexity of fuzzy equalisers making them unsuitable for high speed digital communication applications.

3.4.1 Fuzzy implementation

The FAF discussed earlier is used here to derive the fuzzy implementation of the Bayesian equaliser. The FAF presented in (3.4), along with its membership function in (3.3), is used to derive the fuzzy implementation of Bayesian equaliser. Setting the membership function centres in (3.3) to scalar channel states, spread parameter to channel noise variance and using $M_1 = M_2 = \cdots = M_m = M$ provides a fuzzy equaliser with decision function,

$$\mathfrak{F}_{faf}(\mathbf{x}(k)) = \frac{\displaystyle\sum_{i\,0=1}^{M}\sum_{i\,1=1}^{M}\cdots\sum_{i\,m-1=1}^{M} \vartheta_l(k)^{(i\,0,i\,1,\ldots,i\,m-1)}\left\{\psi_0^{i\,0}(k)\psi_1^{i\,1}(k)\ldots\psi_{m-1}^{i\,m-1}(k)\right\}}{\displaystyle\sum_{i\,0=1}^{M}\sum_{i\,1=1}^{M}\cdots\sum_{i\,m-1=1}^{M}\left\{\psi_0^{i\,0}(k)\psi_1^{i\,1}(k)\ldots\psi_{m-1}^{i\,m-1}(k)\right\}} \tag{3.17}$$

where, $\vartheta_i^l(k)^{(i\,0,i\,1,\ldots,i\,m-1)}$ are free design parameters of the filter which are adjusted during the training process. Here N_c corresponds to all possible combinations of the membership function

taking one from each input scalar and $N_c = M^m$. The membership functions are given by

$$\psi_i^j(k) = \exp\left\{-\frac{1}{2}\left(\frac{|r(k-i) - C_j|^2}{\sigma_\eta^2}\right)\right\} \quad \text{where } 1 \leq j \leq M \text{ and } 0 \leq l \leq m-1 \tag{3.18}$$

The equaliser receives its input from a TDL. Here the membership function centres for each of the inputs are placed in the same position and all the centres use a uniform spread parameter. Under this condition the membership function corresponding to $r(k-l-1)$ will be the membership functions corresponding to $r(k-l)$, delayed by one sample period. This can be represented as

$$\psi_i^j(k) = \psi_{i-1}^j(k-1) \text{ with } 1 \leq i \leq m-1 \tag{3.19}$$

The equaliser function (3.17) finds a weighted sum of the fuzzy basis functions (FBF) given by:

$$\mathbf{\Psi}_i\{\mathbf{r}(k)\} = \frac{\prod_{l=0}^{m-1} \psi_l^{ji}}{\sum_{i=1}^{N_c}\left\{\prod_{l=0}^{m-1} \psi_l^{ji}\right\}} \tag{3.20}$$

where j represents the $(l+1)$ th component of fuzzy IF $\ldots$ THEN $\ldots$ rule i. This corresponds to the IF part of the fuzzy rule $r(k-l)$ given as F_l^j. On observing the decision functions of the NBESS (3.15) and the fuzzy equaliser (3.17) it can be seen that the NBESS has $N_s = 2^{n_c+m-1}$ basis functions and the fuzzy equaliser has $N_c = M^m = (2^{n_c})^m$ basis functions. The number of basis functions in NBESS (3.15) is a subset of the basis functions in the fuzzy equaliser of (3.17), since the centres of the basis functions in (3.16), and the membership function centres in (3.18), are positioned at the same points and the centre spread parameters are uniformly set to σ_η^2. By comparing the equaliser functions in (3.15) and (3.17) it is seen that $(N_c - N_s)$ rules are trivial rules which can be neglected to provide optimal performance. These N_s rules can be extracted from the knowledge of the combination of scalar channel states forming the channel states. With this, the weights corresponding to N_s terms of the fuzzy filter can be assigned $+1/-1$ depending on the values of w_i in NBESS. Hence, the fuzzy equaliser in (3.17) can also be represented by (3.15) where only N_s FBFs out of the possible N_c functions are used. This reduces the computations involved with $(N_c - N_s)$ FBFs and provides the optimum decision function.

The membership function in (3.16) involves evaluation of mN_s membership functions for the equaliser. This involves calculation of the membership function for each component of m dimensional N_s channel states with respect to the input scalars. It can be seen from Table 3.1 that the scalar components of channel states are given by $c_{il} \in C_j$, $1 \leq j \leq M$. Thus calculation of mM membership functions w.r.t. scalar channel states C_j can provide the required mN_s membership functions. The combination of these can provide the channel states. This process has been presented in Table 3.2. The equaliser needs to evaluate only M membership functions. The membership function in (3.16) can be described as

$$\phi_{il} \in \psi_l^j \quad \text{where } 1 \leq i \leq N_s,\ 1 \leq j \leq M \text{ and } 0 \leq l \leq m-1$$

With this the fuzzy equaliser decision function can be described by the equations:

$$\mathfrak{F}\{\mathbf{r}(k)\} = \frac{\sum_{i=1}^{N_s} w_i \left\{ \prod_{l=0}^{m-1} \phi_{il} \right\}}{\sum_{i=1}^{N_s} \left\{ \prod_{l=0}^{m-1} \phi_{il} \right\}} \tag{3.21}$$

$$\psi_l^j = \exp\left\{ -\frac{1}{2} \left(\frac{|r(k-l) - C_j|^2}{\sigma_\eta^2} \right) \right\} \tag{3.22}$$

$$\phi_{il} \in \psi_l^j \tag{3.23}$$

3.4.2 Fuzzy equaliser structure

The structure of the fuzzy equaliser is presented in Figure 3.3. Here, the incoming signal sample is presented to the membership function generator. Each of the components of the membership function generator produces an output ψ_l^j, characterised by its centres C_l^j which are positioned at the scalar channel states. Here j represents the fuzzy centre at the scalar channel states. The membership functions from $r(k-i)$, $1 \leq i \leq m-1$ are generated by passing the membership function from $r(k)$ through a TDL.

The inference block of the equaliser has N_s fuzzy IF ... THEN ... rules with product inference and the rule base is generated from the information of the combination of scalar channel states forming the channel states. Each of these rules uses only one of the ψ_l^j terms corresponding to each of the m inputs to the equaliser. The output of the inference units are suitably weighted and added to provide a and b which provide the function of the defuzzifier.

The output of the equaliser is computed by the equaliser function presented in (3.15) which is $(a-b)/(a+b)$. The output of the decision function passed through $\mathrm{sgn}(x)$ in (2.14) forms the detected sample. An example is considered to illustrate the working of this equaliser:

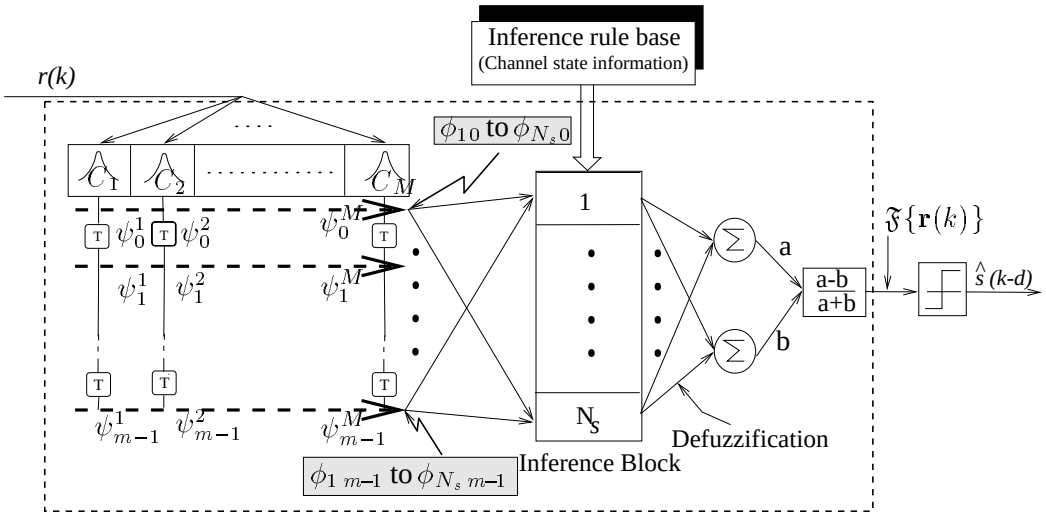


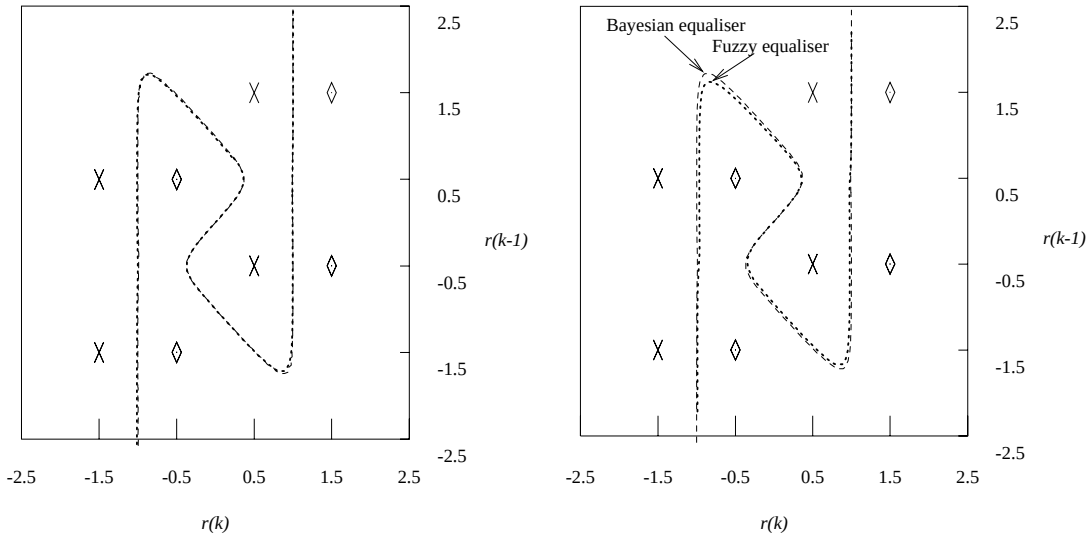
Figure 3.3: *Structure of fuzzy implemented Bayesian equaliser*

EXAMPLE 3.3

The channel considered here is $H(z) = 0.5 + 1.0z^{-1}$. The equaliser is characterised by $m = 2$, $d = 0$ and SNR= 8 dB. This provides $N_s = 8$ channel states and $M = 4$ scalar channel states. The channel states for this equaliser have been presented in Table 2.1. It is also seen that the m-dimensional N_s channel states take their components from the available M scalar channel states. The weights w_i of the equaliser decision function are $+1$ for $\mathbf{c}_1,\ \mathbf{c}_2,\ \mathbf{c}_3,\ \mathbf{c}_4$ and -1 for $\mathbf{c}_5,\ \mathbf{c}_6,\ \mathbf{c}_7,\ \mathbf{c}_8$.

For fuzzy implementation the centres for membership functions are positioned at scalar channel states $+1.5,\ -0.5,\ +0.5$ and -1.5. The membership functions $\psi_1^1,\ \psi_1^2,\ \psi_1^3$ and ψ_1^4 corresponding to $r(k-1)$, are delayed samples of $\psi_0^1,\ \psi_0^2,\ \psi_0^3$ and ψ_0^4 corresponding to $r(k)$. The inference block consist of $N_s = 8$ fuzzy IF $\ldots$ THEN $\ldots$ rules. Here $\phi_{10} = \phi_{20} = \psi_0^1,\ \phi_{30} = \phi_{40} = \psi_0^2,\ \phi_{50} = \phi_{60} = \psi_0^3,\ \phi_{70} = \phi_{80} = \psi_0^4,$ $\phi_{11} = \phi_{51} = \psi_1^1,\ \phi_{21} = \phi_{61} = \psi_1^2,\ \phi_{31} = \phi_{71} = \psi_1^3,$ and $\phi_{41} = \phi_{81} = \psi_1^4$. The products $\phi_{10}\phi_{11},\ \phi_{20}\phi_{21},\ \phi_{30}\phi_{31},\ \phi_{40}\phi_{41}$ constitute the rules for $\mathbf{C}_d^+$, are added to provide a and $\phi_{50}\phi_{51},\ \phi_{60}\phi_{61},\ \phi_{70}\phi_{71},\ \phi_{80}\phi_{81}$ constitute the rules for $\mathbf{C}_d^-$ and are added to provide b. The calculation of the decision function is straight-forward.

The decision boundary of this equaliser is presented in Figure 3.4. Figure 3.4(a) presents the decision boundary of the fuzzy equaliser and the Bayesian equaliser when the channel



(a) Fuzzy and Bayesian Equaliser with actual channel states

(b) Fuzzy equaliser with estimated channel states and Bayesian equaliser with actual channel states

Figure 3.4: *Fuzzy equaliser decision boundary for channel* $H(z) = 0.5 + 1.0z^{-1}$ *with* $m = 2$, $d = 0$ *and SNR= 8 dB.*$\Diamond$*positive channel states and* $\times$ *negative channel states*

states and noise statistics are known, whereas in Figure 3.4(b) the fuzzy equaliser uses the estimated channel states and noise statistics and the Bayesian equaliser uses the true channel parameters. The positive and negative channel states are shown with $\Diamond$ and $\times$ respectively. A study of the decision boundaries shows that, the fuzzy equaliser is able to provide a near optimal decision boundary even at a low SNR of 8 dB.

The fuzzy equaliser developed here, uses FBF with product inference and COG defuzzifier. Owing to the close relationship of this fuzzy equaliser with the Bayesian equaliser, the NBESS has been implemented using a RBF network with scalar centres [78]. However, the use of a fuzzy system to implement this equaliser provides the possibility of using other forms of inference rules and defuzzification processes. This can provide some of the alternate forms of fuzzy implementation of the Bayesian equaliser.

3.4.3 Alternate forms of fuzzy equalisers

Minimum inference

The fuzzy equaliser discussed above works with a product inference type of rule base where the output of each of the N_s inference rules is generated using the product rule. It is also seen from

the membership function generator (3.18) that the membership for any input is $0 < \psi_l^j \leq 1$. Hence the output of any of the inference rules will be in the range $(0, 1]$ and will always be less than the smallest membership function input to the rule. For this reason the product inference rule can be approximated by the minimum inference rule and the equaliser can be represented as

$$\mathfrak{F}\{\mathbf{r}(k)\} = \frac{\sum_{i=1}^{N_s} w_i \left\{ \min_{l=0}^{m-1} \phi_{il} \right\}}{\sum_{i=1}^{N_s} \left\{ \min_{l=0}^{m-1} \phi_{il} \right\}} \tag{3.24}$$

where $\min_{l=0}^{m-1}$ selects the minimum of the m inputs to each of the inference rules. With this the computation of the products can be replaced by a comparison operation which is easy to implement in hardware.

Maximum defuzzification

The output layer of the fuzzy equaliser (3.15) and (3.17) finds a weighted sum of the inference rules and normalises this with the sum of all inference outputs. The weights associated with the inference rules are $+1/-1$. It is seen that the rule nearest to the input vector would provide the maximum output, and the contribution from the remaining rules is minimal. These characteristics of the decision function can be utilised by replacing the COG defuzzifier with a maximum defuzzifier. This defuzzifier can be combined either with product inference or with minimum inference. The equaliser decision function for these two cases can be represented as,

$$\mathfrak{F}\{\mathbf{r}(k)\} = \frac{w_{max} \max_{i=1}^{N_s} \left\{ \prod_{l=0}^{m-1} \phi_{il} \right\}}{\max_{i=1}^{N_s} \left\{ \prod_{l=0}^{m-1} \phi_{il} \right\}} \tag{3.25}$$

with product inference and

$$\mathfrak{F}\{\mathbf{r}(k)\} = \frac{w_{max} \max_{i=1}^{N_s} \left\{ \min_{l=0}^{m-1} \phi_{il} \right\}}{\max_{i=1}^{N_s} \left\{ \min_{l=0}^{m-1} \phi_{il} \right\}} \tag{3.26}$$

with minimum inference. The notation $\max_{i=1}^{N_s}$ corresponds to the maximum of the available

N_s inferences and w_{max} is the weight associated with the maximum inference. With this the decision functions, (3.25) and (3.26) use maximum defuzzification, where the output of the equaliser is based on the maximum of the N_s inference rules and the weight associated with it. The equaliser (3.25) uses product inference whereas (3.26) uses the minimum inference rule. In both of these defuzzification processes the computation of the weighted sum of the inferences is replaced by comparison operation.

With the above analysis four types of fuzzy equalisers approximating the Bayesian decision function can be designed. These equalisers can provide alternative equaliser architectures with a reduction in computational complexity. All four forms of fuzzy equaliser are presented in Table 3.3.

No.	Fuzzy Type	Inference	Defuzzification
1	Fuzzy# 1	Product	Centroid
2	Fuzzy# 2	Product	Maximum
3	Fuzzy# 3	Minimum	Centroid
4	Fuzzy# 4	Minimum	Maximum

Table 3.3: *Different types of fuzzy equalisers with selection of inference rules and defuzzification process*

EXAMPLE 3.4

This example discusses the effects of different inference rules and defuzzification process on the fuzzy Bayesian equaliser decision surface and decision boundary. Here the channel used is

$$H_3(z) = 0.2682 + 0.9298z^{-1} + 0.2682z^{-2} \tag{3.27}$$

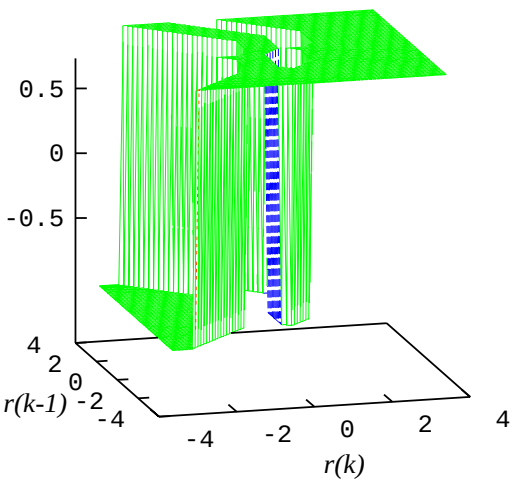
This channel is a mixed phase channel with its zeros located at $z_1 = -3.1492$ and $z_2 = -0.3175$. The equaliser is characterised by equaliser length $m = 2$ and delay $d = 0$. The system SNR is assumed to be 15 dB. The equaliser has $N_s = 16$ channel states which are generated from $M = 8$ scalar channel states. The fuzzy equaliser centres are positioned at $\pm 1.4662,\ \mp 0.9298,\ \pm 0.3934$ and ± 0.9298 which are the locations of the scalar channel states. The channel states were estimated with 200 training samples averaged over 50 experiments. All forms of implementation of fuzzy equalisers presented in Table 3.3 were investigated. The fuzzy equalisers used the estimated noise statistics.

The computational complexity of Fuzzy#1 equaliser is the largest and the complexity of Fuzzy#4 equaliser is smallest due to the types of inference rules and defuzzification

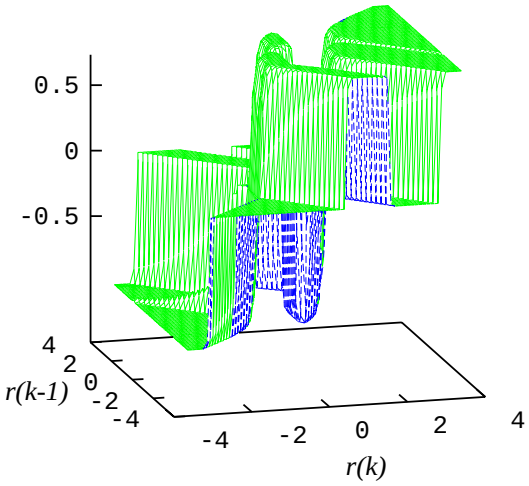
processes they implement. The decision surface of these equalisers is plotted in Figure 3.5. From the decision surfaces it can be seen that all the fuzzy equalisers provide near optimal decision surfaces. The optimal decision surface is the decision surface provided by the Fuzzy#1 equaliser.

The decision boundaries of the fuzzy equalisers along with the optimal Bayesian equaliser are presented in Figure 3.6. The positive and negative channel states are presented with $\Diamond$and $\times$ symbols respectively. From the decision boundary curves it is observed that the Fuzzy#1 and the Fuzzy#2 equalisers provide a near optimal performance. The fact that the optimal equaliser and the Fuzzy#1 equaliser decision boundaries are nearly the same confirms the fuzzy implementation of the Bayesian equaliser. The decision boundary provided by the Fuzzy#3 and the Fuzzy#4 equalisers deviates from the optimal equaliser decision boundary. These equalisers provide a different decision boundary when the input vector is far from the channel states. But, these equalisers do provide a nonlinear decision boundary separating the positive and negative channel states successfully. This shows that all form of the fuzzy equalisers presented in Table 3.3 are capable of providing nonlinear decision boundaries.

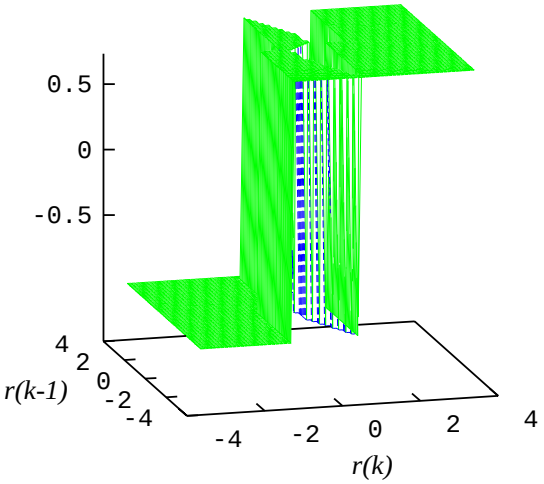
Thus the capability of fuzzy equalisers to provide near optimal decision boundary with a variety of network architectures has been demonstrated.



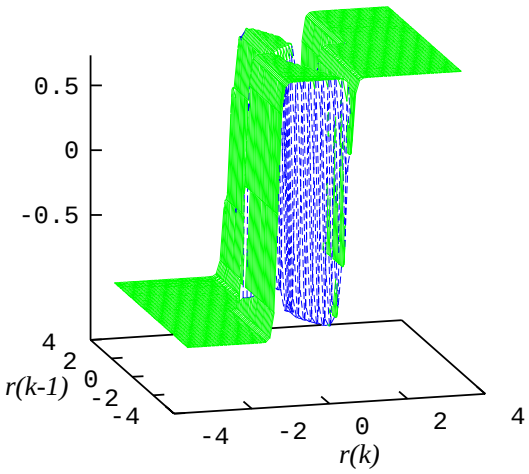
(a) Fuzzy#4(Minimum Inference, Maximum Defuzzification)



(b) Fuzzy#3(Minimum Inference, Centroid Defuzzification)

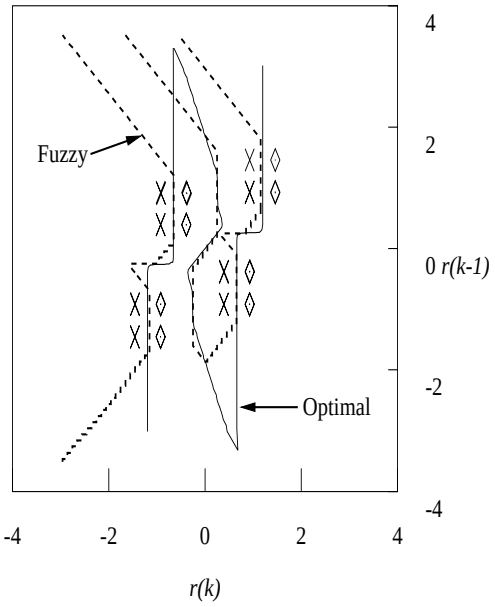


(c) Fuzzy#2(Product Inference, Maximum Defuzzification

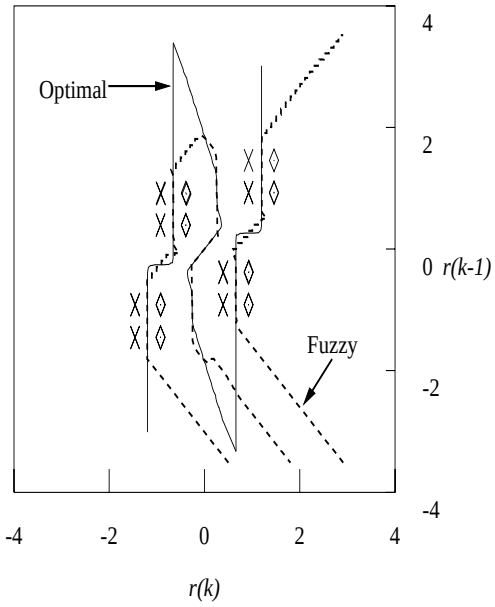


(d) Fuzzy#1(Product Inference, Centroid Defuzzification

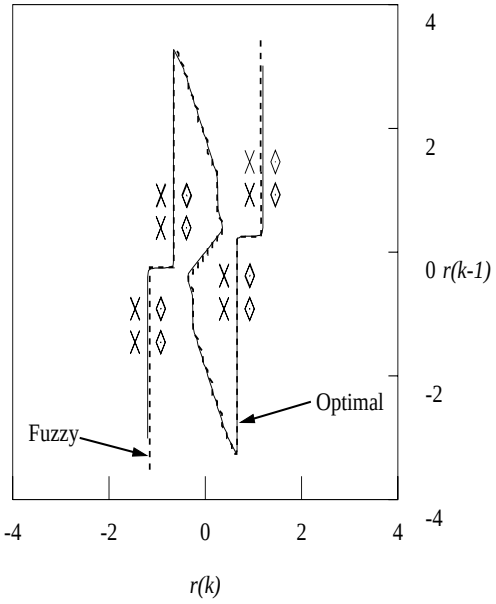
Figure 3.5: *Decision surface of different forms of fuzzy equalisers with channel* $H(z) = 0.2682 + 0.9298z^{-1} + 0.2682z^{-1}$ *for* $m = 2$, $d = 0$ *and SNR=15 dB using estimated channel states*



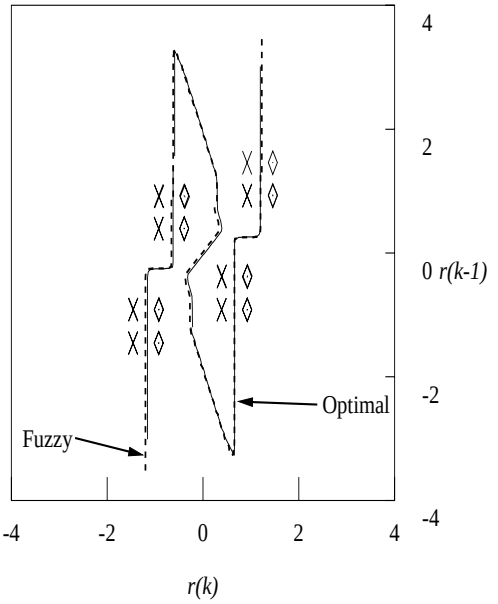
(a) Fuzzy#4 (Minimum Inference and Maximum Defuzzification)



(b) Fuzzy#3 (Minimum Inference and centroid Defuzzification)



(c) Fuzzy#2 (Product Inference and Maximum Defuzzification)



(d) Fuzzy#1 (Product Inference and Centroid Defuzzification)

Figure 3.6: *Decision boundary of different forms of fuzzy equalisers with channel* $H(z) = 0.2682 + 0.9298z^{-1} + 0.2682z^{-1}$ *for* $m = 2, d = 0$ *and SNR=15 dB using estimated channel states;* $\Diamond$*positive channel states and* $\times$ *negative channel states*

3.5 Fuzzy equaliser training

The fuzzy equaliser was presented in Section 3.4. The design of the fuzzy equaliser developed in (3.17) requires the knowledge of the channel states $\mathbf{c}_i$, $1 \leq i \leq N_s$ and the weights w_i. These equaliser parameters can be estimated during the training period and after training the equaliser can use its previous decisions in decision directed mode to update its parameters. The process of estimating these parameters is discussed here.

3.5.1 Step1: Channel state estimation

The estimation of the decision function using the fuzzy equaliser given by (3.15) and (3.17) needs the channel state information to form the rule base. Implicit estimation of the channel states requires channel information which in most cases is not available. However, the channel states can be estimated during the training period by any of the following techniques [53].

- The channel model can be identified using LMS /RLS algorithms. With the knowledge of the channel, it is straight forward to calculate the scalar channel states and their combinations which form the channel states. However, when the channel suffers from nonlinear distortion, estimation of the channel is a difficult process.
- The channel states can be directly estimated using a vector clustering algorithm. The number of channel states are exponentially related to the channel dispersion order and equaliser feed forward order. Equalisers with large number of channel states[3] would require a longer training sequence.
- The scalar channel states can be estimated using a scalar supervised clustering technique. These scalar channel states, in conjunction with the training signal, can provide the order in which they occur, and these can be used to estimate the channel states [57]. This process has been presented in Example 3.2. The number of scalar channel states depends only on the channel order and hence requires a smaller length of training sequence compared to direct channel state estimation. The scalar channel states always occur in pairs so that $C_j = -C_{M-j+1}$, $1 \leq j \leq M$. This feature of the scalar channel states is evident from Table 3.1. This would require only estimation of $\frac{M}{2} = 2^{n_c-1}$ scalar states, resulting in faster estimation. These scalar channel states can be estimated with the supervised

[3]This situation can occur if the equaliser order m is large or channel order n_c is large

κ-means algorithm which has been presented in Appendix A. The convergence curve for a typical scalar channel states estimation using κ-means clustering algorithm is presented in Figure 3.7. Here the channel used is $H(z) = H_4(z) = 0.5 + 0.81z^{-1} + 0.31z^{-2}$ with SNR=10 dB. The process of channel state estimate has been averaged over 20 experiments. From the training curves it is seen that the scalar channel states converge to the desired states in around 30 iterations. This fast training feature can provide considerable advantage in DCR applications.

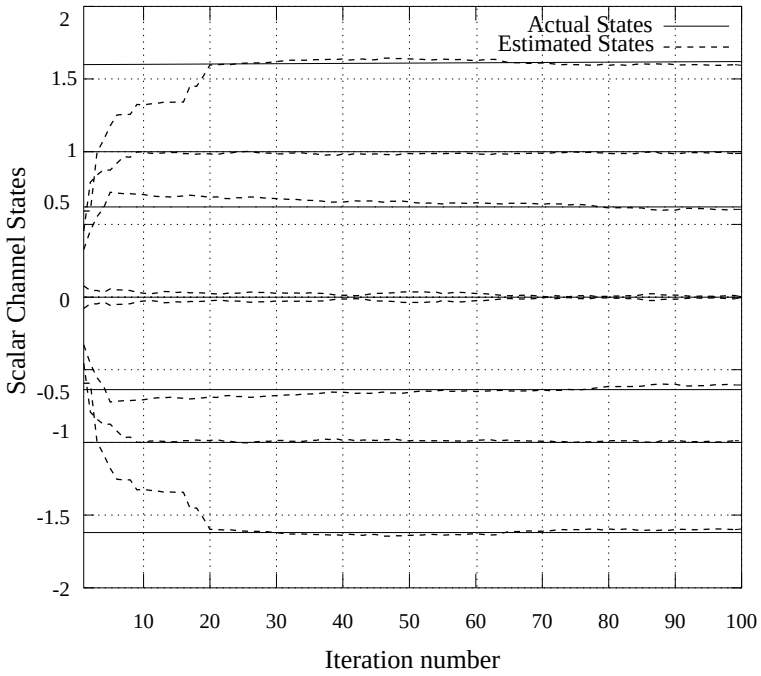


Figure 3.7: *Scalar Channel states training curve for channel* $H(z) = 0.5 + 0.81z^{-1} + 0.31z^{-2}$*, the actual channel states* $\pm 1.62, \pm 1.0, \pm 0.62, \pm 0.00$

This experiment indicates that the estimation of scalar channel states for the fuzzy equaliser requires only a few training samples even at low SNR.

3.5.2 Step2: Equaliser weight update

Once the scalar channel states have been estimated the fuzzy rules can be formed and the equaliser constructed with weights of the inference rules assigned $+1/-1$, depending on whether the rule belongs to $\mathbf{C}_d^+$ or $\mathbf{C}_d^-$. Estimating the channel states and the noise statistics can involve some error. In order to compensate for this the weights associated with the rules can be fine tuned with the LMS algorithm given in (3.5). This step would require only a few

samples as the initial weight assignment is very close to the final values. This process would not require additional training overhead, since the training signal used to estimate the channel states can be reused for equaliser weight training.

3.6 Advantages of fuzzy equaliser

The fuzzy implementation of NBESS provides the Bayesian equaliser decision function. A closer look at the Bayesian decision function in (3.8) and the fuzzy implementation of NBESS in (3.15) shows some of the advantages of the fuzzy implementation of the Bayesian equaliser. One of the advantages of the fuzzy equaliser is the need for only a small training sequence. This aspect has been discussed in the previous section. The other major advantages of this equaliser are discussed in this section, namely lower complexity and subset state selection.

3.6.1 Computational complexity

On completion of equaliser training, the equaliser parameters are fixed and the actual detection of transmitted symbol starts. The computational requirements of the fuzzy equaliser and NBESS are the same. The computations required for estimating each of the samples with the Bayesian equaliser and its RBF implementation, NBESS and fuzzy equaliser (Fuzzy#1 in Table 3.3) are listed in Table 3.4. The second part of the table provides the typical computational requirements for a equaliser with $m = 4$, $n_c = 3$ and $M = 8$. From this table the following points can be inferred with regards to the computational advantages of the fuzzy implementation of Bayesian equaliser:

Equaliser Type	**Add/ Sub**	**Mul**	**Div.**	e^{-x}
Bayesian(RBF)	$2mN_s$	mN_s	N_s	N_s
NBESS	$M + N_s$	$M + mN_s$	$M + 1$	M
Fuzzy	$M + N_s$	$M + mN_s$	$M + 1$	M
Bayesian (RBF)	512	256	64	64
NBESS	72	264	9	8
Fuzzy	72	264	9	8

Table 3.4: *Computational complexity comparison for the Bayesian equalisers, the NBESS and the fuzzy equalisers. Second part typical computational complexities for equalisers with* $m = 4$, $n_c = 3$, $N_s = 64$ *and* $M = 8$.

- The fuzzy implementation of the Bayesian equaliser provides a significant reduction in addition, division and $\exp(x)$ evaluations.

- The time shift property of the membership function generation provides a considerable reduction in evaluation of $\exp(x)$ functions and divisions.

- The evaluation of $\exp$ and division functions in a Bayesian equaliser is related to N_s which in turn is exponentially related to the sum of the equaliser and channel order. In the fuzzy equaliser it is related to M which is exponentially related to channel order only. Thus, as the equaliser order increases the reduction in computational complexity for fuzzy equaliser over the Bayesian equaliser is exponentially related.

- The minimum inference rule and the maximum defuzzification discussed in subsection 3.4.3 replace each of the product computations in the inference generator and the defuzzifier by a comparison operation. This is very easy to implement and fast to process in real time. The computation involved for the estimation of each symbol with this modification for the four forms of fuzzy equalisers are presented in Table 3.5. The second part of the Table provides the typical computational figures for an equaliser with $m = 4$, $n_c = 3$ and $M = 8$. From this it can be seen that using the minimum inference or maximum defuzzification process replaces the product computation by comparison operations. These provide an alternate approximation to the Bayesian equaliser with a reduction in the computational complexity. This provides a lot of scope for varied implementation of Bayesian equalisers w.r.t. computational complexity.

Fuzzy Type	**Inf. Type**	**Defuzz. Type**	**Add/ Sub**	**Mul**	**Div.**	e^{-x}	**Compare**
Fuzzy#1	Prod	COG	$M + N_s$	$M + mN_s$	$M + 1$	M	
Fuzzy#2	Prod	Max.	M	$M + (m-1)N_s + 1$	$M + 1$	M	N_s
Fuzzy#3	Min.	COG	$M + N_s$	$M + N_s$	$M + 1$	M	$(m-1)N_s$
Fuzzy#4	Min.	Max.	M	$M + 1$	$M + 1$	M	mN_s
Fuzzy#1	Prod	COG	72	264	9	8	
Fuzzy#2	Prod	Max.	8	201	9	8	64
Fuzzy#3	Min.	COG	72	72	9	8	192
Fuzzy#4	Min.	Max.	8	9	9	8	256

Table 3.5: *Computational complexity comparison for different forms of fuzzy equalisers*

In this section the computational complexity of the fuzzy equaliser has been compared with the Bayesian equaliser which can be implemented with a RBF networks. The Bayesian equaliser provides the optimum performance for symbol-by-symbol equalisers providing the lower bound

for the BER performance. The computational issues of the Bayesian equaliser against MLSE and linear equalisers are widely available in the literature [53, 132] and hence have not been discussed in this thesis.

3.6.2 Subset state selection

The Bayesian equaliser decision function in (3.8) is based on a weighted sum of N_s basis functions centred at the channel states. From the decision function it can be seen that the contribution of a channel state is inversely related to its distance from the input vector. Under this circumstance, if a set of channel states near the input vector can be found, the equaliser decision function can be approximated with this subset of the available N_s channel states. Chng [77] proposed a process of selecting a subset of available channel states to approximate the Bayesian equaliser with a smaller number of channel states. Other forms of subset centre selection with the RBF implementation of Bayesian equalisers have also been proposed [133]. With fuzzy implementation it is very easy to employ subset state selection to reduce the number of inference rules, which reduces the computational complexity. This only involves modification of the membership function. In general all M membership functions corresponding to an input provide non-zero output irrespective of the input scalar. If an input is far from a scalar centre, the membership function from that centre is negligible and can be neglected. Keeping this in view, it may be enough to use only a set of nearest centres from the observed received scalars for membership function calculation and the membership function contribution from other centres can be neglected. This provides a subset of non-zero membership functions out of the available M functions for each input. This would generate only a smaller number of nonzero inferences $N_s^{'} < N_s$ where $N_s^{'}$ is the subset of the N_s rules in the fuzzy equaliser. Using some simple checks to determine these rules the decision function can be computed. This process is illustrated with the following Example.

EXAMPLE 3.5

The system considered in this example was used in Example 3.3. The channel used in this study is $H(z) = 0.5 + 1.0z^{-1}$ and the equaliser order $m = 2$ and the decision delay $d = 0$. The equaliser has 8 channel states constructed from 4 scalar channel states. The fuzzy equaliser decision making capability for this system was presented in Figure 3.4. Here in this example the concept of subset centre selection is demonstrated and the decision making capacity of the fuzzy equaliser with membership function generated from a subset of scalar channel states is presented in Figure 3.8. The positive channel states

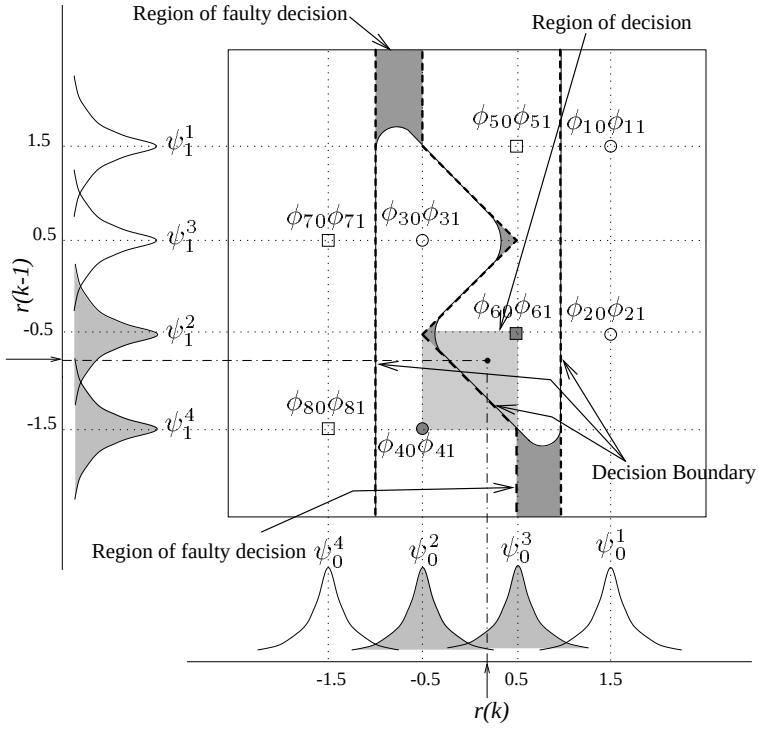


Figure 3.8: *Decision Boundary for subset centre selection with membership function modification, channel* $H(z) = 0.5 + 1.0z^{-1}$, $m = 2$ *and* $d = 0$*;* $\circ$ *positive channel states,* $\square$ *negative channel states*

are shown as $\circ$ and negative channel states are shown as $\square$. The membership functions for $r(k)$ and $r(k-1)$ are shown along the sides[4]. An input vector $[0.0, -0.75]^T$ is considered. Selecting the membership functions from scalar centres that are in immediate neighbourhood and making the membership function from other scalar centres 0, the input vector provides nonzero membership functions for ψ_0^2, ψ_0^3 and ψ_1^2, ψ_1^4 only. These, when translated with inference rules with channel states into $\mathbb{R}^2$, provide only two nonzero inference rules corresponding to the channel states $[-0.5, -1.5]^T$ and $[0.5, -0.5]^T$ which correspond to centres $\mathbf{c}_2 \in \mathbf{C}_d^+$ and $\mathbf{c}_4 \in \mathbf{C}_d^-$. The region of space that will be covered by these rules correspond to the channel states is shown as shaded region in the Figure. With this the decision function for this input region is a straight line equidistant from both centres in the space covered by the membership functions. With a change in the input vector different sets of inference rules corresponding to channel states would be selected providing a combined decision boundary as shown as shaded region in the Figure. All these individual decision boundaries join to provide a nonlinear decision

[4]Membership functions for $r(k-1)$ are the delayed membership function from $r(k)$.

boundary. The region in which the equaliser is unable to approximate the decision region is also shown in the figure. From this it can be seen that the decision boundary formed with the modified membership function is very close to the optimal one. Here the equaliser is capable of providing a nonlinear decision boundary where the channel states are nonlinearly separable by using only 2 inference rules out of the total of N_s rules.

This form of modification of the membership function can reduce the computational complexity of the equaliser considerably. The computation involved per sample calculation with this form of membership function is presented in Table 3.6. The second part of the Table represents the computations involved when the channel order $n_c = 3$, equaliser order $m = 4$ when the equaliser has $N_s = 64$ channel states and $M = 8$ scalar channel states. From Table 3.6 it can be observed that most of the product computations have been replaced by comparison operations.

This modification of the membership function provides a natural method for selecting a subset of the available channel states resulting in computational complexity reduction. However, if the channel states are very closely spaced this process of using only 2 membership functions in each signal dimensions may not provide good performance and more than 2 nonzero membership functions in each signal dimensions of the input vector may be required. With an increase in number of membership functions the number of non-zero inference rules increase, providing a better performance at the cost of higher computational complexity. However, if a subset of the available scalar channel states is used the numbers of selected fuzzy rules $N_s^{'}$ will always be less than maximum possible rules N_s. This provides a way of trading performance with computational complexity within the equaliser. This is illustrated in the following Example.

Inf Type	**Defuzz Type**	**Add/ Sub**	**Mul**	**Div.**	e^{-x}	**Compare**
Prod	COG	$M+2$	$M+2m$	3	2	
Prod	Max.	M	$M+2(m-1)+1$	3	2	<2
Min.	COG	$M+2$	$M+2$	3	2	$<2(m-1)$
Min.	Max.	M	$M+1$	3	2	$<m$
Prod	COG	10	16	3	2	
Prod	Max.	8	15	3	2	<2
Min.	COG	10	10	3	2	<6
Min.	Max.	8	9	3	2	<4

Table 3.6: *Computational complexity comparison for fuzzy equalisers with modified membership function generation for subset state selection; second part for equalisers with* $m = 4$, $n_c = 3$, $N_s = 64$ *and* $M = 8$.

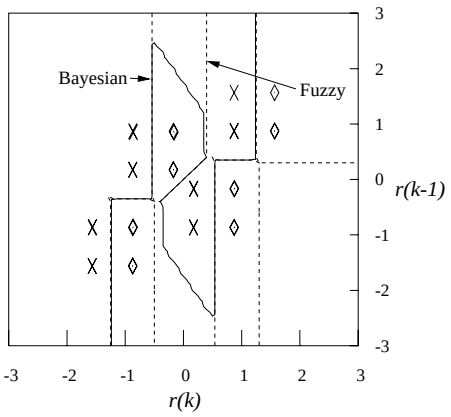
EXAMPLE 3.6

In this example the channel is

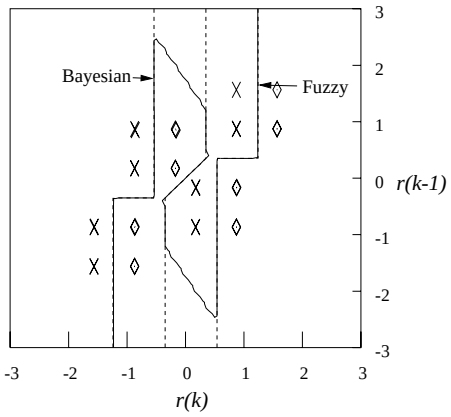
$$H(z) = H_5(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2} \tag{3.28}$$

This channel has its zeros located at $z_1 = -2$ and $z_2 = -0.5$. The equaliser length $m = 2$ and decision delay $d = 0$ is used. These equaliser and channel parameters provide $N_s = 16$ channel states and $M = 8$ scalar channel states. The system SNR=20 dB. The scalar channel states are located at $\pm 1.5668,\ \pm 0.8704,\ \mp 0.174,\ \mp 0.8704$. The fuzzy equaliser uses the knowledge of channel states and noise variance. With this the fuzzy equaliser membership function centres are positioned at $\delta_0^1 = 1.5668,\ \delta_0^2 = 0.8704,$ $\delta_0^3 = -0.174,\ \delta_0^4 = -0.8704,\ \delta_0^5 = 0.8704,\ \delta_0^6 = 0.174,\ \delta_0^7 = -0.8704$ and $\delta_0^8 = -1.5668$.

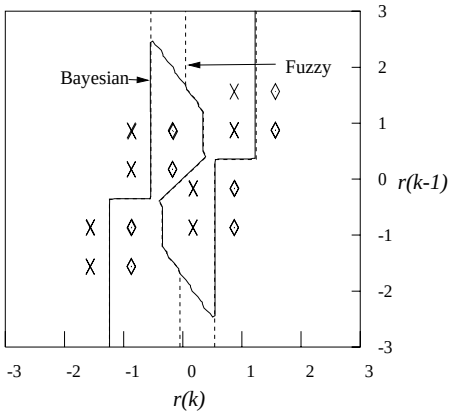
The decision boundary provided by the Fuzzy#1 equaliser using $2, 3, 4, 5, 6,$ and 8 scalar centres closest to the input vector is presented in Figure 3.9(a) through Figure 3.9(f). The optimal Bayesian equaliser decision boundary for this case is presented along with the fuzzy equaliser decision boundaries. From the optimal decision boundary it can be seen that the decision boundary is nonlinear and the fuzzy equalisers can successfully partition the channels states corresponding to $\mathbf{C}_d^+$ and $\mathbf{C}_d^-$, using only 2 fuzzy centres closest to the input scalars. But, the decision boundary is very different from the optimal one. With an increase in number of scalar centres used in decision function evaluation, the decision boundaries approach the Bayesian equaliser decision boundary. It is also observed that an increase in the number of non zero membership functions used makes the decision boundary closer to the optimal Bayesian decision boundary only for the regions in the decision space that are far from the channel states. The fuzzy equaliser with 8 nonzero membership functions provides the optimal equaliser decision boundary.



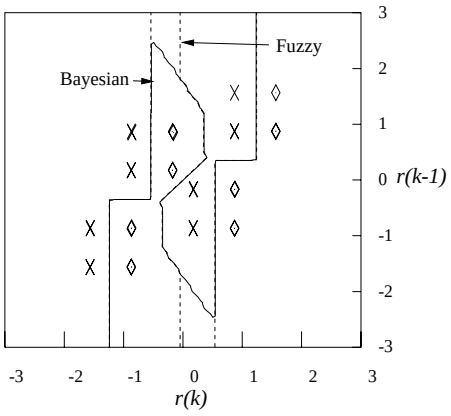
(a) 2 centres



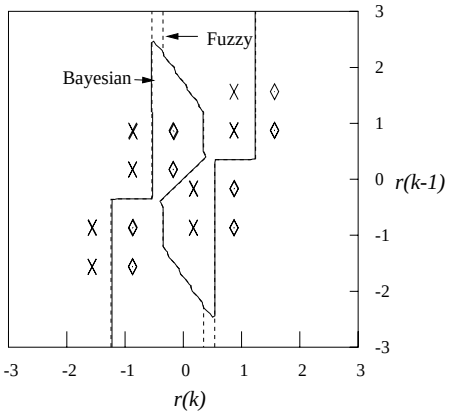
(b) 3 centres



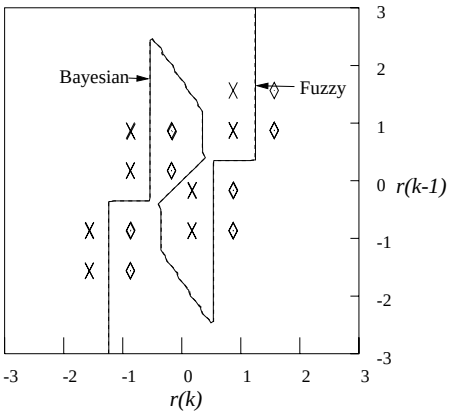
(c) 4 centres



(d) 5 centres



(e) 6 centres



(f) All 8 centres

Figure 3.9: *Decision boundary with subset centre selection in fuzzy Equalisers with channel* $H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$, $m = 2$, $d = 0$ *for SNR=20 dB;*◊ *positive channel states and* $\times$ *negative channel states*

3.7 Results and discussion

Fuzzy equalisers were developed in Section 3.4 and their advantages discussed in Section 3.6. From the study of the decision boundaries provided by the fuzzy equalisers it was seen that all forms of the fuzzy equalisers provide an efficient scheme for equalisation by providing a nonlinear decision boundary close to the optimal. The actual performance of an equaliser is the BER. This section investigates the BER performance of fuzzy equalisers for a variety of channels and equaliser parameters. Here all the experiments were continued untile either 1000 errors were observed or 10^8 symbols were transmitted.

3.7.1 Fuzzy implemented Bayesian equaliser

BER performance of different types of fuzzy equalisers developed in Section 3.4 were evaluated with extensive computer simulation.

In this study the channel used was

$$H_6(z) = 0.407 - 0.815z^{-1} - 0.407z^{-2} \tag{3.29}$$

The equaliser parameters were set to $m = 5$ and $d = 3$. This channel has 2 zeros situated at $z_1 = 2.4163$ and $z_2 = -0.4139$. The scalar channel states are located at ± 0.815, ± 0.001, ∓ 0.815 and ∓ 1.629. For the equaliser order of $m = 5$ there are $N_s = 128$ channel states constituting 128 fuzzy rules. The BER performance of the Fuzzy#1, Fuzzy#3, Fuzzy#4 and the Bayesian equalisers for SNR=1 dB to 14 dB, using Monte Carlo simulations is shown in Figure 3.10. Here the channel information was assumed to be available and with this scalar channel states were estimated. The 8 fuzzy equaliser membership function centres were positioned at $\delta_0^1 = 0.815, \delta_0^2 = 0.001, \delta_0^3 = -0.815, \delta_0^4 = -1.629, \delta_0^5 = 1.629, \delta_0^6 = 0.815, \delta_0^7 = -0.001, \delta_0^8 = -0.815$.

The following points can be observed from the BER curves for different equaliser configurations. The BER performance of the Fuzzy#1 equaliser is exactly same as the Bayesian equaliser which can be implemented with the RBF network. This result demonstrates that the Bayesian equaliser can be implemented by the Fuzzy#1 equaliser. The performance of the computationally efficient Fuzzy#3 and Fuzzy#4 equalisers, are close to the optimal, and they suffer from nearly 1 dB performance degradation at 10^{-5} BER and < 0.5 dB at 10^{-3} BER. This revalidates

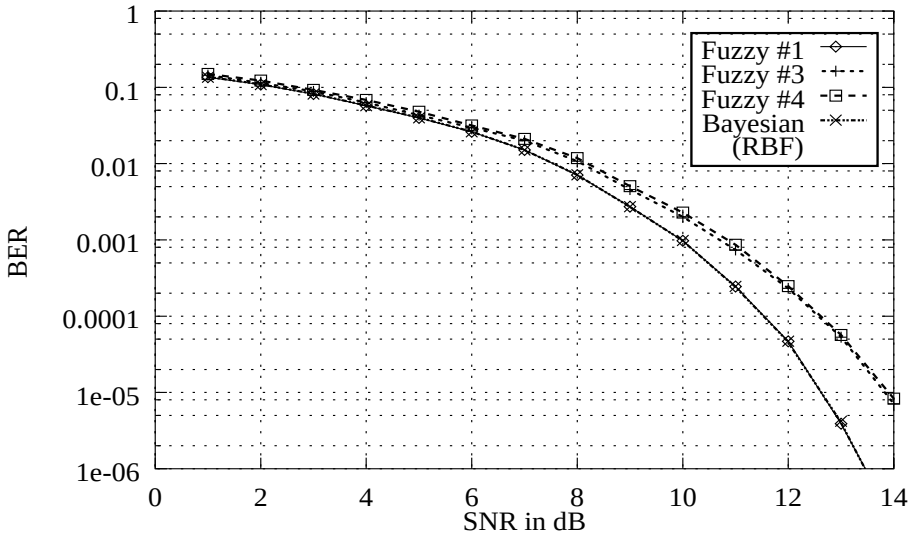


Figure 3.10: *BER performance for Fuzzy#1, Fuzzy#3, Fuzzy#4 and Bayesian equalisers for channel* $H_5(z) = 0.407 - 0.815z^{-1} - 0.407z^{-2}$, $m = 5,\ d = 3$ *with knowledge of the channel*

the use of the minimum inference rule and the maximum defuzzification process.

In the next experiment the fuzzy equaliser performance was evaluated by constructing the equalisers with estimated channel states. Here the channel used was $H(z) = H_5(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$. The equaliser order and the decision delay were set to $m = 4$ and $d = 1$. The actual scalar channel states for this channel are located at ± 1.5668, ± 0.8704, ∓ 0.174 and ∓ 0.8704. However, in this study the equaliser scalar channel states and the channel noise statistics were evaluated using the supervised κ-means clustering algorithm [5] with 200 training samples averaged over 50 experiments. The Fuzzy equaliser here uses 64 fuzzy IF ... THEN ... inference rules derived from the 64 channel states. The BER performance of different equalisers using Monte Carlo simulation is presented in Figure 3.11.

After the equaliser was constructed the equaliser weights were trained with the same set of training samples used for channel states estimation. The step size ϱ in the fuzzy LMS algorithm (3.5) was fixed at 0.01. The linear equaliser was trained with a conventional LMS algorithm. This training involved 1000 samples averaged over 50 experiments with a step size of 0.03. The Bayesian equaliser which can be implemented with RBF was simulated with the knowledge of the channel states and channel noise statistics to provide the lower bound for the equaliser performance. From the equaliser BER curves it can be seen that the Fuzzy#1 equaliser

[5] This algorithm is presented in Appendix A.

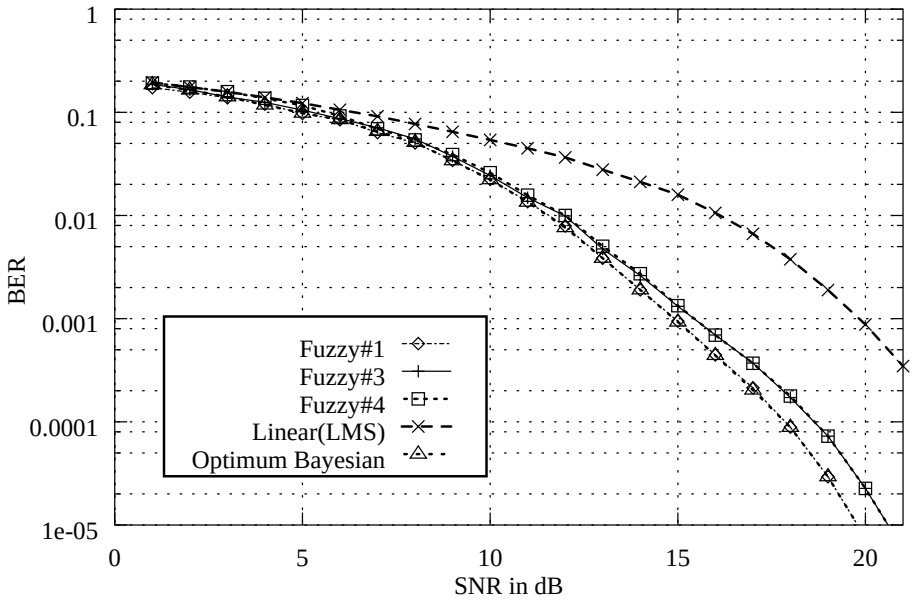


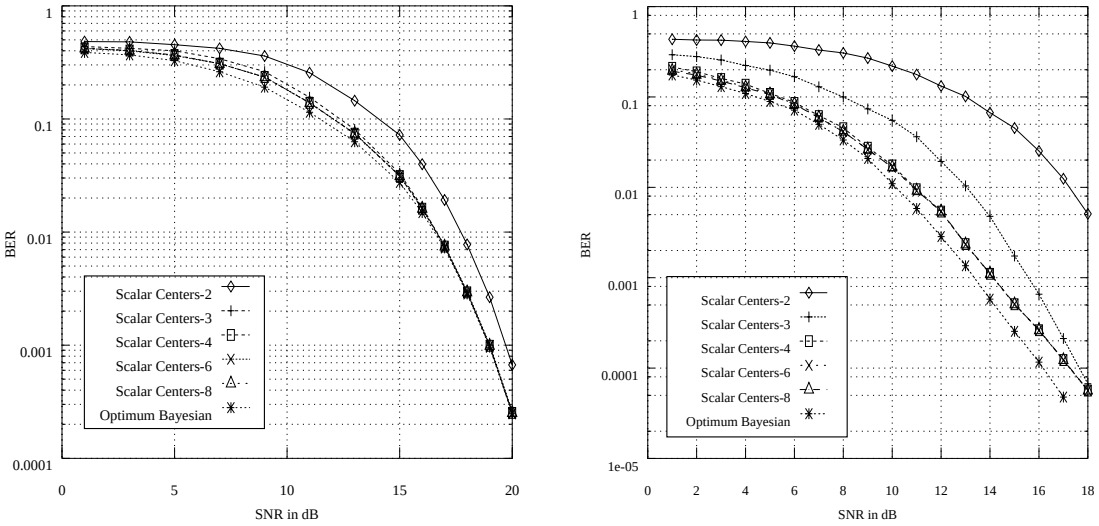
Figure 3.11: *BER performance for Fuzzy#1, Fuzzy#3, Fuzzy#4, Bayesian and LMS linear equalisers for channel* $H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$, $m = 4$, $d = 1$ *with estimated channel states and noise statistics*

performs nearly like Bayesian equaliser. However, the Fuzzy #3 and Fuzzy#4 equalisers suffer from minor performance degradation due to the simplified inference rule and/or defuzzification processes involved. This performance degradation is again around 1dB at 10^{-5} BER. All these equalisers outperform the linear equaliser.

3.7.2 Fuzzy equaliser with subset state selection

This subsection presents the BER performance of fuzzy equaliser with subset state selection by modification membership function generation. The channel used for this study was $H_5(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$, with $m = 5$. Two types of equalisers for this problem were investigated. In the first case the equaliser decision delay was set to $d = 0$ and in the second case it was set to $d = 3$. Since this channel is a mixed phase channel with a zero outside the unit circle in the z-plane, a linear equaliser with $d = 0$ can not equalise the channel [59] but, with $d = 3$, a linear equaliser can equalise it successfully. The optimal Bayesian equaliser for this problem has $N_s = 128$ channel states derived in terms of $M = 8$ scalar channel states. The fuzzy equaliser uses 128 fuzzy rules which are derived from the channel states. In the Monte Carlo simulations the number of nonzero membership functions for the fuzzy equalisers were varied from $M_1 = 2$ to $M_1 = 8$. $M_1 = 8$ provides the optimum Bayesian equaliser

when used with product inference and centroid defuzzifier. The fuzzy equalisers considered in this study used the minimum inference rule and the maximum defuzzification process which constitutes the Fuzzy#4 equaliser with minimum computational complexity. The scalar channel states and the channel noise statistics were estimated using 200 training samples averaged over 50 experiments and the equaliser weights were trained with the same training signal. The optimal Bayesian equaliser was simulated assuming the true channel information and noise statistics to estimate the channel states. The BER performance of the equalisers with Monte Carlo simulations for a wide range of SNR's is presented in Figure 3.12(a) and 3.12(b) for $d = 0$ and $d = 3$ respectively.



(a) Subset centre selection for $m = 5$, $d = 0$

(b) Subset centre selection for $m = 5$, $d = 3$

Figure 3.12: *BER performance of fuzzy equalisers with subset centre selection using channel* $H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$

From the BER plots it can be seen that the fuzzy equaliser with a subset of centres can provide a near optimal performance. The equaliser with 2 non zero membership functions suffers from performance degradation. This performance degradation can be attributed to the fact that, under many input conditions, none of the 128 rules is used in decision making, thus resulting in large errors. However, increasing M_1 from 4 to 8 does not provide any observable performance improvement. Hence, it can be inferred that, under this circumstance, using the 4 highest non zero membership function to the input scalars only is sufficient to provide a near optimal performance. It was also observed from the simulation studies that this condition of using the 4 nearest membership functions selects between 4-12 fuzzy rules from the available 128 rules.

With this it is seen that using only around 10% of channel states in form of fuzzy rules is sufficient to provide the optimal performance in this case.

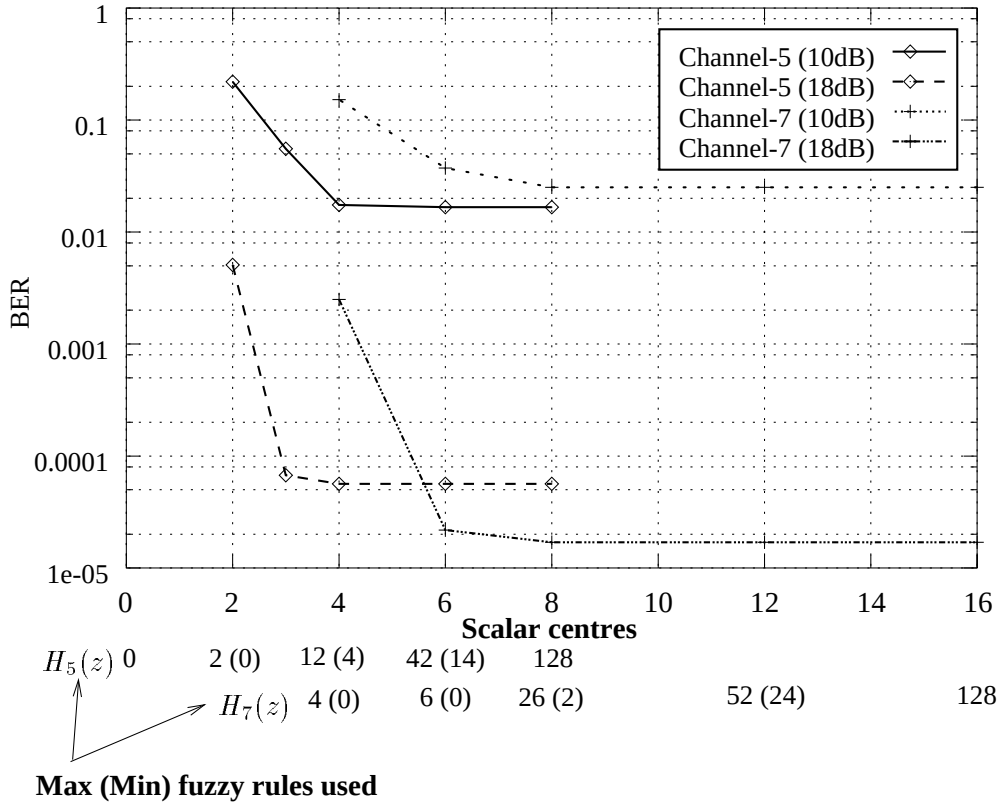


Figure 3.13: *Effect of number of subset scalar centre on BER performance of fuzzy equalisers for different channels*

In the next part of this study the effect of the number of non zero membership functions closest to the input scalars was studied for a fixed SNR. For this study two channels $H(z) = H_5(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$ and $H(z) = H_7(z) = 0.7255 + 0.5804z^{-1} + 0.3627z^{-2} + 0.0724z^{-3}$ were used. The equaliser length and delay were set to $m = 5, d = 3$ for H_4 and $m = 4, d = 0$ for H_6. These parameters provided best performance for the equalisers and use $N_s = 128$ fuzzy rules in both cases. The BER performance of both the equalisers against the number of non zero membership functions used, for SNR of 10 dB and 18 dB is presented in Figure 3.13. The x axis also shows the maximum and minimum number of fuzzy IF ... THEN ... inference rules used for a variety of input conditions.

From the results it is seen that if the fuzzy equaliser uses at least one rule in decision making for all varieties of input the performance of the equaliser approaches the optimal performance. It can also be seen that the fuzzy equalisers provide near optimal performance when only $\frac{1}{2}$

the membership functions are used. This used only around $10\% - 20\%$ of the fuzzy inference rules, as can be seen from Figure 3.13. For H_5 the performance of the fuzzy equaliser does not improve by using more than 4 nonzero membership functions closest to the input scalar. Similarly, for H_7 the performance reaches the optimal performance when a minimum of 8 membership functions are used. These membership functions translate to use of maximum of 12 and 26 fuzzy rules out of 128 rules for the channels H_5 and H_7 respectively.

3.8 Conclusion

The Bayesian equaliser was implemented with fuzzy systems and the performance of the fuzzy equaliser was evaluated. The following conclusions can be drawn from the study presented in this chapter.

- The fuzzy equaliser provides an efficient implementation of the Bayesian equaliser.
- The fuzzy implemented Bayesian equaliser provides a wider choice of equaliser structure compared to the RBF implementation of the Bayesian equaliser.
- All forms of fuzzy equalisers i.e. Fuzzy#1 and the computationally efficient Fuzzy#2, Fuzzy#3 and Fuzzy#4 provide a nonlinear decision boundary close to the optimal equaliser and provide very little performance degradation in terms of BER.
- Fuzzy equalisers incorporating subset centre selection provide efficient schemes for reducing computational complexity. Simulation studies suggest that the use of only 10%-20% of the channel states out of all channel states is sufficient to provide near optimal performance. These subset states can be automatically selected by a selective use of a subset of available membership functions.
- The computational complexity of the RBF implementation of the Bayesian equaliser is related to $N_s = 2^{m+n_c-1}$ (dependent on m and n_c), whereas the complexity of the fuzzy equaliser is related to N_s for multiplications but is related to $M = 2^{n_c}$ (dependent only on n_c) for summation, exponentiation and divisions.
- The training overhead in fuzzy equalisers is related to the estimation of M scalar parameters which provide fast training and ease of tracking in decision directed mode. This feature of fuzzy equalisers could make them suitable for use in mobile communication applications.

Chapter 4
Fuzzy Equaliser for Co-channel Interference Suppression

4.1 Introduction

The problem of channel equalisation in general was discussed in chapter 2 and that of the CCI was introduced in section 2.3. In Chapter 3 the Bayesian equaliser for the ISI channels was implemented with a fuzzy system. This chapter analyses the problem of channel equalisation in DCS which are affected by CCI. It was seen that the channel equalisation is a nonlinear problem. But the presence of CCI makes it more complex. Under most circumstances the decision boundary of the optimum equaliser for ISI channels can be approximated by a linear decision boundary with proper selection of decision delay d. However, in the presence of moderate to severe CCI, the optimal decision boundary changes, and in most circumstances it cannot be approximated with a linear boundary. These conditions demand the use of special forms of nonlinear equalisers that can compensate for this distortion.

Advances in TDMA mobile cellular communications and the rising demand for these services have been partly made possible by sophisticated equalisation techniques. But with the increase in the number of users CCI is becoming a limitation on the system performance. This chapter discusses the development of fuzzy equalisers for CCI channels. An equaliser not designed to mitigate the effects of CCI can suffer from major performance degradation in moderate to high CCI conditions. The optimum symbol-by-symbol equaliser for a CCI channel requires large computational complexity. This trend can be offset by efficient schemes for CCI mitigation with reduced computational burden. This chapter attempts to address some of the issues in this regard. A modified form of the fuzzy equaliser designed for ISI channels is presented. This equaliser possesses the capability of successfully equalising channels with CCI. Important issues discussed in this chapter are as follows.

- The fuzzy implementation of the Bayesian equaliser is derived and the computational issues for this equaliser are discussed.

- A modified form of the fuzzy equaliser discussed in chapter 3 is presented which provides efficient CCI compensation. This equaliser is termed a fuzzy–CCI equaliser. The fuzzy–CCI equaliser developed here works with an input pre-processor in conjunction with the fuzzy equaliser for ISI channels. The input pre-processor helps to remove the CCI efficiently.
- A wide variety of simulation studies are presented to validate the performance of the equaliser developed in this chapter.

The chapter is organised as follows. The next section provides a background to the concept of CCI compensation and also surveys the literature in relation to equalisation of CCI channels. In section 4.3 the normalised Bayesian CCI equaliser with scalar centres (NBSS–CCI) is derived and section 4.4 develops the fuzzy implementation of NBSS–CCI as well as presenting a modified form of the fuzzy equaliser developed for ISI channels to compensate for CCI. Section 4.5 presents the decision feedback concept in this scenario while section 4.6 presents the implementation issues. Section 4.7 includes the simulation results and finally section 4.8 provides the concluding remarks.

4.2 Background and literature review

A rise in demand for DCR has added more users and services to the existing facilities and with this the CCI is increasingly limiting the system performance. The main cause of CCI here, is the interference from the signal of a cell in the neighbouring cluster using the same carrier frequency as the desired user. This problem becomes more severe in a fading environment when the signal suffers from multi-path fading in addition to channel ISI and AWGN [134, 135]. Similar problems of CCI, ISI and AWGN are also encountered in other communication systems such as dual polarised microwave radio[47], twisted pair subscriber loops [47, 136], multiuser spread spectrum systems and multi pair cables. The problem of CCI is also encountered in digital magnetic data recording. This section presents the communication model for this problem where the communication system is affected by CCI. A general communication system in this type of environment was discussed in section 2.3. It is assumed that the receiver filter in the receiver front end removes the ACI efficiently and the equaliser only works to combat the effects of CCI, ISI and AWGN.

4.2.1 System model

The discrete time model of the communication system discussed in this chapter is presented in Figure 4.1. This model is widely used to represent a communication system corrupted with CCI, ISI and AWGN [47]. Here $H(z)$ is the channel transfer function which is corrupted with L interfering co-channels $H_{co\_i}(z)$, $1 \leq i \leq L$. The impulse response of the channel can be represented as [1],

$$H(z) = \sum_{j=0}^{n_c-1} a_{0,j} z^{-j} \tag{4.1}$$

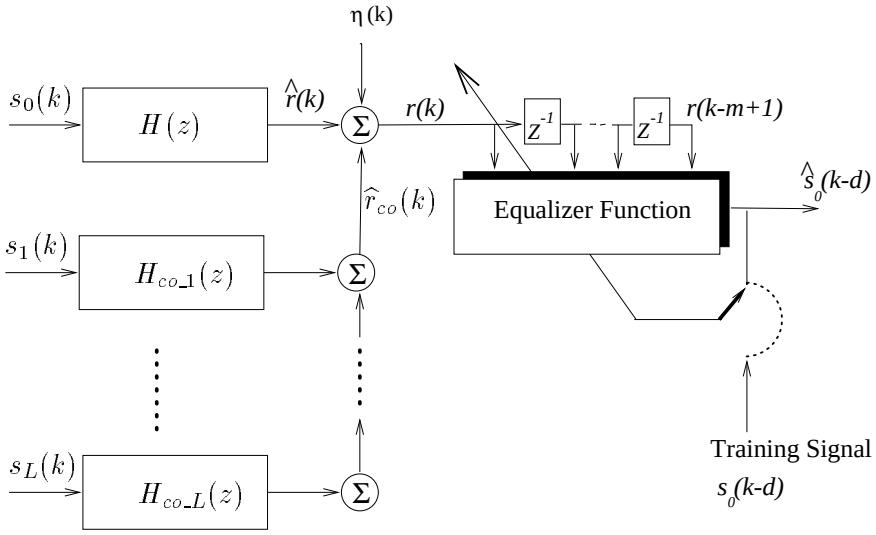


Figure 4.1: *Discrete-time model of a DCS corrupted with co-channel interference*

and the impulse response of the co-channels can be represented as,

$$H_{co\_i}(z) = \sum_{j=0}^{n_{ci}-1} a_{i,j} z^{-j} \qquad 1 \leq i \leq L \tag{4.2}$$

where n_{ci} and $a_{i,j}$ are the length and tap weights of the i th co-channel impulse responses. It is assumed that the communication system is binary. This makes the analysis simple and it can be extended to any communication system in general. The transmitted symbols $s_i(k)$, $0 \leq i \leq L$ for the channel ($i = 0$) and the co-channels ($1 \leq i \leq L$) are binary i.i.d., i.e. they comprise

[1] This impulse response for channel was derived in Chapter 2.

$\{\pm 1\}$ symbols. They satisfy the conditions

$$\mathcal{E}[s_i(k)] = 0 \tag{4.3}$$

$$\mathcal{E}[s_i(k_1)s_j(k_2)] = \delta(i-j)\delta(k_1-k_2) \tag{4.4}$$

where $\mathcal{E}[.]$ denotes the expectation operator and

$$\delta(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} \tag{4.5}$$

The channel output scalars can be represented as

$$r(k) = \widehat{r}(k) + \widehat{r}_{co}(k) + \eta(k) \tag{4.6}$$

where $\widehat{r}(k)$ is the desired received signal, $\widehat{r}_{co}(k)$ is the interfering signal and $\eta(k)$ is the noise component. The noise, $\eta(k)$, is assumed to be Gaussian with zero mean and a variance of $\mathcal{E}[\eta^2(k)] = \sigma_\eta^2$ and is uncorrelated with the data. The desired and the interfering signal can be represented as

$$\widehat{r}(k) = \sum_{j=0}^{n_c-1} a_{0,j} s_0(k-j) \tag{4.7}$$

$$\widehat{r}_{co}(k) = \sum_{i=1}^{L} \sum_{j=0}^{n_{ci}-1} a_{i,j} s_i(k-j) \tag{4.8}$$

With this the SNR, signal to interference ratio (SIR) and signal to interference plus noise ratio (SINR) can be defined as

$$\text{SNR} = \frac{\sigma_c^2}{\sigma_\eta^2} \tag{4.9}$$

$$\text{SIR} = \frac{\sigma_c^2}{\sigma_{co}^2} \tag{4.10}$$

$$\text{SINR} = \frac{\sigma_c^2}{\sigma_\eta^2 + \sigma_{co}^2} \tag{4.11}$$

where σ_{co}^2 is the co-channel signal power. With the transmitted signal power equal to unity, σ_c^2 and σ_{co}^2 can be defined as the channel and co-channel power respectively. The task of the equaliser depicted in Figure 4.1 is to estimate the transmitted sequence $s_0(k-d)$ based on the channel observation vector $\mathbf{r}(k) = [r(k),\ r(k-1),\ \ldots,\ r(k-m+1)]^T$. The equaliser estimated symbol $\widehat{s}_0(k-d)$, is desired to provide minimum BER w.r.t. $s_0(k-d)$. During the training period, the equaliser uses a copy of the transmitted sequence stored locally and during actual detection the past detected symbols can be used to update equaliser parameters in a decision directed mode. The equaliser does not have access to the transmission sequence $s_i(k),\ 1 \leq i \leq L$ corresponding to the co-channels.

4.2.2 Literature review

The problem of CCI was considered as inter channel interference in multichannel DCS's. The receivers designed for multi pair cables and the receivers in the up-link path of radio communication systems are required to optimise the detection of all transmitter sources. The optimum receivers under these circumstances using linear and MLSE algorithms were originally proposed in [137] and [138] respectively. The design aspects of receiver and transmitter filters for joint estimation of all the channels were analysed in [139]. The process of equalisation for joint estimation of signals in a multi channel TDMA mobile radio systems has been recently reported in [140]. Joint estimation of multiple channel signals for radio communication applications using MLSE and MAP algorithms were reported in [141] and blind estimation techniques for these applications were presented in [142].

The problem addressed in this chapter of the thesis is similar to the multichannel communication system but is limited to the system where the receiver recovers only the signal corresponding to the desired user. This corresponds to down-link in a typical mobile radio communication application. The techniques used for the joint estimation of multiple channel signals can be used here but the receiver can be further optimised to provide better performance for detection of only the single desired signal, while rejecting the interference. The interfering cross talk signal in DCS possesses cyclostationary property [47] and a receiver not optimised for cross talk can exhibit severe performance degradation. In [143] a special form of time dependent adaptive filter was shown to out perform conventional adaptive filters in CCI mitigation. The equalisation of cross talk in digital subscriber lines using an FSE [21] with decision feedback provides major performance gains [144] since FSE treats CCI as a cyclostationary interference which is different from stationary noise. The T-spaced equalisers treat CCI as stationary noise

and in the process of equalisation and exploit the staistical properties of the signal. However the CCI is similar to the siganl of interest since both consist of a finite set of discrete states. This accounts for the performance degradation of the T-spaced equalisers. The effects of the transmitter and receiver filter BW for CCI suppression in multiple twisted pair cables were analysed in [46, 145], where it was shown that every increase in BW size equal to symbol rate may provide the flexibility to completely suppress an additional cyclostationary interference. In [136] the design issues for transmitter and receiver filters in these environments were addressed. FSE with decision feedback in conjunction with large transmitter and receiver BW provided encouraging performance in a quasi-static fading environment [146, 147]. Even though the use of large transmitter and receiver BW in conjunction with a DFE with fractional tap spacing provides major performance advantages, these may not yield a solution to existing problems since an increase in transmission BW may not always be permissible.

The equaliser that can provide the minimum bit error rate (BER) under the above conditions is the infinite memory MLSE designed for CCI, which would require the knowledge of the co-channels. Normally the receiver does not have the access to the training signal for the interfering channels. Excluding this, the formulation of the MLSE detector for this problem would involve large computational complexity [92]. However, a finite memory symbol-by-symbol equaliser can be used for this problem in line with the equalisers developed for ISI channels. This equaliser would also require the knowledge of the channel and co-channel states making the equaliser training difficult. It has been seen that symbol-by-symbol linear equaliser suffer from performance degradation since the optimal decision boundary of an equaliser is generally nonlinear. For this reason nonlinear equalisers have been seen to provide better performance for the ISI channel. Some of these techniques were discussed in section 2.7. Similar nonlinear equalisation techniques have been attempted for equalisation for CCI channels. In [148], an equaliser designed using a RBF network was shown to out perform the linear equaliser. Similarly equalisers were designed for CCI channels with a functional link ANN [100] and a multi layer ANN [149]. A polynomial perceptron [88] with fractional sampling was also shown to perform satisfactorily for M-QAM communication systems. However, most of these studies considered high SIR conditions or high SNR conditions. These equalisers suffered severe performance degradation under low SIR with high SNR conditions. Equalisers based on the Mahalonobis distance classifier [150] with the Viterbi algorithm have shown good performance for stationary channels. But, these equalisers need a long decision delay like the Viterbi equalisers and their complexity grows with decision delay. This long delay is likely to cause

performance degradation in mobile communication applications since a long delay in channel estimation may result in large tracking errors.

In a recent study, Chen et. al. [92] proposed a Bayesian DFE that incorporates CCI compensation (Bayesian–CCIDFE). This equaliser can provide the optimum decision for the symbol-by-symbol equaliser. This equaliser was trained in two stages. The first stage uses supervised clustering and subsequently unsupervised clustering is used to remove the effects of CCI. This equaliser is computationally complex and the computational complexity grows if there is more than one co-channels. In this chapter a fuzzy system based equaliser is designed which addresses some of these issues. The complexity of this fuzzy equaliser is comparable to the Bayesian equaliser treating CCI as AWGN but provides a performance which is close to the Bayesian–CCIDFE presented in [92].

4.3 Normalised Bayesian equaliser in CCI, ISI and AWGN

The optimal decision function of the Bayesian equaliser for ISI channels was presented in section 2.5 and its normalised form with scalar states was presented in section 3.3. In this section the decision function for a normalised Bayesian equaliser with scalar states for CCI channel (NBSS–CCI) is derived.

In order to derive the NBSS–CCI the Bayesian equaliser decision function in (2.30) is considered first.

$$\mathfrak{F}\{\mathbf{r}(k)\} = \sum_{i=1}^{N_s} w_i \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_i\|^2}{2\sigma_\eta^2}\right) \tag{4.12}$$

where $N_s = 2^{n_c+m-1}$ is the number of channel states, w_i are the weights associated with each of the channel states. $w_i = +1$ if $\mathbf{c}_i \in \mathbf{C}_d^+$ and $w_i = -1$ if $\mathbf{c}_i \in \mathbf{C}_d^-$. The estimate of the symbol from the memoryless detector is defined as

$$\widehat{s}(k-d) = \begin{cases} 1 & \mathfrak{F}\{\mathbf{r}(k)\} \geq 0 \\ -1 & \mathfrak{F}\{\mathbf{r}(k)\} < 0 \end{cases} \tag{4.13}$$

To derive the decision function of the Bayesian equaliser for CCI channels (Bayesian–CCI), it is assumed that there is only one interfering co-channel. If there are more, the same analysis

can generally be extended. In the presence of CCI, the interfering signal $\widehat{\mathbf{r}}_{co}(k) = [\widehat{r}_{co}(k), \widehat{r}_{co}(k-1), \ldots, \widehat{r}_{co}(k-m+1)]^T$ will have a finite number of states. These states are described as co-channel states. There would be $N_{s,co} = 2^{n_{c1}+m-1}$ co-channel states $\mathbf{c}_{co,\alpha}$, $1 \leq \alpha \leq N_{s,co}$, which constitute the noise free received vectors due to the co-channel signal in the absence of the desired signal. The desired signal, due to channel ISI, provides N_s channel states $\mathbf{c}_i$, $1 \leq i \leq N_s$ in the absence of CCI. In the presence of CCI and ISI the noise free signal vectors will be the combination of all possible channel and co-channel states. With this the noise free received signal vector can be represented as $\mathbf{c}_i + \mathbf{c}_{co,l}$, $1 \leq i \leq N_s$ and $N_{s,co} = 2^{n_{c1}+m-1}$. The presence of the co-channel states modifies the decision function in (4.12) to

$$\mathfrak{F}_{CCI}\{\mathbf{r}(k)\} = \sum_{i=1}^{N_s} \sum_{\alpha=1}^{N_{s,co}} w_i \exp\left(\frac{-\|\mathbf{r}(k) - \{\mathbf{c}_i + \mathbf{c}_{co,\alpha}\}\|^2}{2\sigma_\eta^2}\right) \tag{4.14}$$

This forms the optimum equaliser decision function of a symbol spaced equaliser for a CCI channel. Here the decision function is affected by the channel states and co-channel states. The co-channels states surround the channel states. There are $N_{s,co}$ co-channel states corresponding to each channel state. All the co-channel states corresponding to a specific channel state inherit the weight associated with that channel state. With this understanding, the channel states in the Bayesian equaliser in (2.30) are replaced by a group of co-channel states due to the presence of CCI. This equaliser can be implemented with a RBF network, where the RBF uses $N_s N_{s,co}$ centres each with a spread $\sigma_r^2 = \sigma_\eta^2$ [148].

4.3.1 Normalised Bayesian CCI equaliser with scalar channel states(NBSS–CCI)

The Bayesian–CCI equaliser in (4.14) can be normalised to provide the actual detected samples rather than a decision function. This normalisation is in line with the Bayesian equaliser decision function for ISI channels in (3.9). With this, the Bayesian–CCI decision function can be presented as

$$\mathfrak{F}_{CCI}\{\mathbf{r}(k)\} = \frac{\displaystyle\sum_{i=1}^{N_s} \sum_{\alpha=1}^{N_{s,co}} w_i \exp\left(\frac{-\|\mathbf{r}(k) - \{\mathbf{c}_i + \mathbf{c}_{co,\alpha}\}\|^2}{2\sigma_\eta^2}\right)}{\displaystyle\sum_{i=1}^{N_s} \sum_{\alpha=1}^{N_{s,co}} \exp\left(\frac{-\|\mathbf{r}(k) - \{\mathbf{c}_i + \mathbf{c}_{co,\alpha}\}\|^2}{2\sigma_\eta^2}\right)} \tag{4.15}$$

Where the decision function in (4.14) has been normalised with the output of all the channel and co-channel state combinations. The process of working of this normalised Bayesian–CCI equaliser is presented in the following Example.

EXAMPLE 4.1

The process of channel and co-channel states formation with the decision making process in a CCI environment is discussed here. The channel used can be represented by its z-transform $H(z) = H_1(z) = 0.5 + 1.0z^{-1}$. The channel is corrupted by the co-channel $H_{co\_1}(z) = \lambda H_2(z) = \lambda(1.0 + 0.2z^{-1})$ where λ represents the scale factor for adjustment of SIR. Selecting $\lambda = 0.3467$ provides SIR=10 dB. The system SNR was considered to be 15 dB. The equaliser length and delay are set to $m = 2$ and $d = 0$. The channel states for this channel are presented in Table 2.1. The equaliser parameters provide $N_s = 8$ channel states and $N_{s,co} = 8$ co-channel states.

No.	$\mathbf{c}_{co,\alpha}$	$s_1(k)$	$s_1(k-1)$	$s_1(k-2)$	$\widehat{\mathbf{r}}_{co}(k)$	
					$c_{co,\alpha 0}$	$c_{co,\alpha 1}$
1	$\mathbf{c}_{co,1}$	1	1	1	1.2	1.2
2	$\mathbf{c}_{co,2}$	1	1	−1	1.2	0.8
3	$\mathbf{c}_{co,3}$	1	−1	1	0.8	−0.8
4	$\mathbf{c}_{co,4}$	1	−1	−1	0.8	−1.2
5	$\mathbf{c}_{co,5}$	−1	1	1	−0.8	1.2
6	$\mathbf{c}_{co,6}$	−1	1	−1	−0.8	0.8
7	$\mathbf{c}_{co,7}$	−1	−1	1	−1.2	−0.8
8	$\mathbf{c}_{co,8}$	−1	−1	−1	−1.2	−1.2

Table 4.1: *The co-channel state calculation for channel* $H(z) = 1.0 + 0.2z^{-1}$ *with* $m = 2$, $d = 0$, $N_{s,co} = 8$ *and* $\lambda = 1$

The co-channel states for the equaliser are presented in Table 4.1. Each of the components of the co-channel states in Table 4.1 is to be scaled with λ to provide the co-channel state at the desired SIR. The components of the co-channel states are presented as $c_{co,\alpha 0}$ and $c_{co,\alpha 1}$. Each of the 8 channel states are associated with 8 co-channel states. The locations of the channel states, co-channel states and the optimal decision boundary are presented in Figure 4.2. Here the positive channel states are presented as □ and the negative channel states are represented as ◯. Each of the channel states are associated with 8 co-channel states. The co-channel states associated with positive channel states are represented with ◊ and the co-channel states in association with negative channel states are presented with $+$ symbols.

From the Figure 4.2 it is seen that the presence of CCI increases the number of states

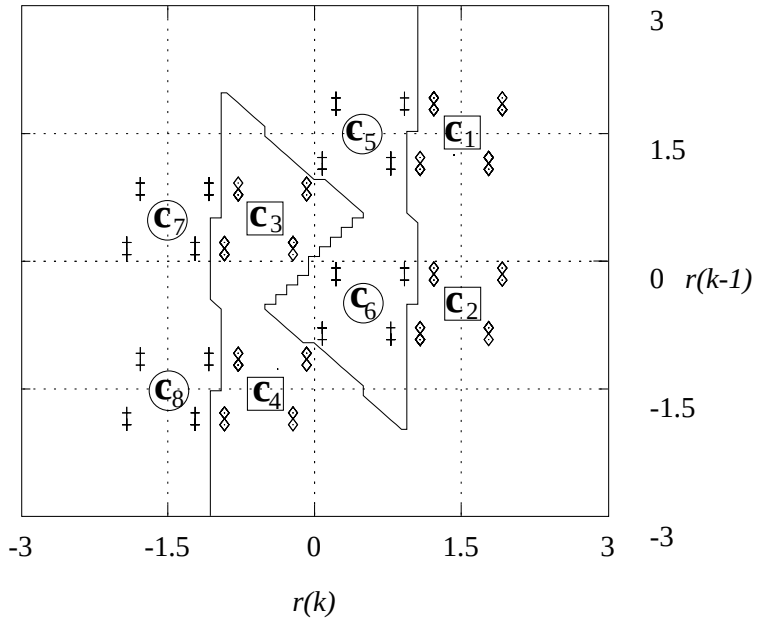


Figure 4.2: *Bayesian–CCI equaliser decision boundary with channel and co-channel states with channel* $H(z) = 0.5 + 1.0z^{-1}$*, co-channel* $H_{co,1}(z) = \lambda(1.0 + 0.2z^{-1})$*,* $m = 2$ *and* $d = 0$ *for SNR=15 dB and SIR=10 dB,*□ *positive channel states,* ◯ *negative channel states,* ◇ *co-channel states with positive channel states and* + *co-channel states with negative channel states*

used in the decision function calculation. Each of the N_s channel states $\mathbf{c}_i$, are surrounded by $N_{s,co}$ co-channel states. For this co-channel at a SIR=10 dB, the decision boundary is close to the Bayesian equaliser decision boundary that treats CCI as AWGN. This was presented in Figure 2.9. With a reduction in SIR the co-channel states move away from the channel states and an increase in the SIR moves the co-channel states closer to the channel states. When SIR=∞ the co-channel states merge with the channel states. From the Figure 4.2 it can be inferred that a reduction in SIR will result in the co-channel states corresponding to the positive and negative channel states to cross over, which may requires a very complex decision boundary. This situation is presented in Figure 4.3 for SIR=5 dB. From this Figure it can be seen that the co-channel states corresponding to positive and negative channel states have crossed over and in this situation the optimal decision boundary has become very complex. The decision boundary of a Bayesian equaliser or any other type of nonlinear equaliser treating CCI as AWGN would be similar to the decision boundary presented in Figure 4.2 and these equalisers would fail even for a noise free channel, with the interference remaining the same.

In line with NBESS, the equaliser decision function in (4.15) can also be represented in terms

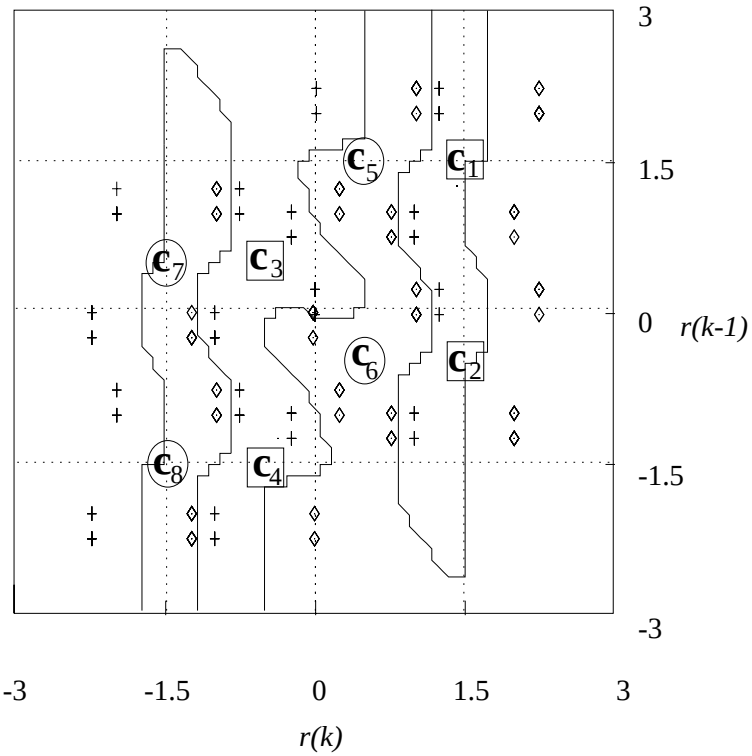


Figure 4.3: *Bayesian–CCI equaliser decision boundary with channel and co-channel states with channel* $H(z) = 0.5 + 1.0z^{-1}$ *and co-channel* $H_{co\_1}(z) = \lambda(1.0 + 0.2z^{-1})$, $m = 2$ *and* $d = 0$ *for SNR=15 dB and SIR=5 dB,*□ *positive channel states,* ◯ *negative channel states,* ◊ *co-channel states with positive channel states and* + *co-channel states with negative channel states*

of its scalar channel states. The channel and co-channel states are taken from combination of the scalar channel and scalar co-channel states. Each of the channel states $\mathbf{c}_i$ can be represented as

$$\mathbf{c}_i = \left[c_{i0},\, c_{i1}, \ldots, c_{il},\ \ldots,\ c_{i(m-1)}\right]^T \tag{4.16}$$

where c_{il}, $0 \leq l \leq (m-1)$ represents the $(l+1)$ component of the channel state $\mathbf{c}_i$, $1 \leq i \leq N_s$. Each of these components $c_{il} \in C_j$ and C_j, $1 \leq j \leq M$ are the scalar channel states. In a similar way each of the co-channel states $\mathbf{c}_{co,\alpha}$ can also be represented as

$$\mathbf{c}_{co,\alpha} = \left[c_{co,\alpha 0}, c_{co,\alpha 1}, \ldots, c_{co,\alpha l}, \ldots, c_{co,\alpha\,(m-1)}\right]^T \tag{4.17}$$

where $c_{co,\alpha l}$, $0 \leq l \leq (m-1)$ represents the $(l+1)$ component of the co-channel state $\mathbf{c}_{co,\alpha}$, $1 \leq \alpha \leq N_{s,co}$. Each of these components $c_{co,\alpha l} \in C_{co,j}$ and $C_{co,j}$, $1 \leq j \leq M_{co}$ constitutes

the scalar co-channel states. The number of scalar co-channel states $M_{co} = 2^{n_{c1}}$.

This concept of the scalar channel and scalar co-channel state combining to form the channel and co-channel state is presented in Figure 4.4. Here $m = 2$ so that $\mathbf{r}(\mathbf{k}) = [r(k), r(k-1)]^T$. One of the channel states $\mathbf{c}_i$ is presented with $\bigcirc$ symbol. The case considered here is similar to the Example 4.1. The system consists of 8 co-channel states and these co-channel states surround the channel state. These co-channel states are presented with $+$. The decision function in (4.15) calculates the Euclidean distance of each co-channel states with respect to the input vector with the function $\|\mathbf{r}(k) - \{\mathbf{c}_i + \mathbf{c}_{co,1}\}\|^2$. The process of this Euclidean distance calculation is presented in the Figure 4.4. This distance can also be represented as,

$$\|\mathbf{r}(k) - \{\mathbf{c}_i + \mathbf{c}_{co,1}\}\|^2 = |r(k) - \{c_{i0} + c_{co,10}\}|^2 + |r(k-1) - \{c_{i1} + c_{co,11}\}|^2 \tag{4.18}$$

where the Euclidean distance in $\mathbb{R}^2$ has been replaced by absolute distance in $\mathbb{R}$.

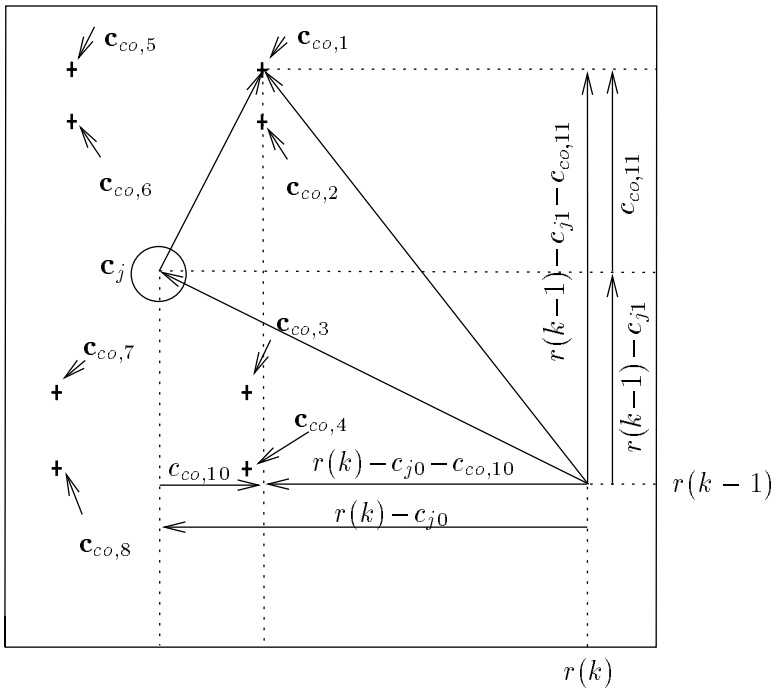


Figure 4.4: *Representation of channel states and co-channel states using scalar channel states and scalar co-channel states,* $\bigcirc$ *channel state and* $+$ *co-channel states*

With this understanding each combination of channel and co-channel states $\mathbf{c}_i + \mathbf{c}_{co,\alpha}$ can be represented in terms of their scalar components. Taking advantage of the $\exp(.)$ operator the

decision function in (4.15) can be conveniently represented as

$$\mathfrak{F}_{NBSS\text{–}CCI}\left\{\mathbf{r}(k)\right\} = \frac{\sum_{i=1}^{N_s}\sum_{\alpha=1}^{N_{s,co}} w_i \left\{\prod_{l=0}^{m-1} \exp\left(-\frac{|r(k-l)-\{c_{il}+c_{co,\alpha l}\}|^2}{2\sigma_\eta^2}\right)\right\}}{\sum_{i=1}^{N_s}\sum_{\alpha=1}^{N_{s,co}} \left\{\prod_{l=0}^{m-1} \exp\left(-\frac{|r(k-l)-\{c_{il}+c_{co,\alpha l}\}|^2}{2\sigma_\eta^2}\right)\right\}} \quad (4.19)$$

The normalised form of the Bayesian CCI equaliser presented in (4.15) and the equaliser in (4.19) provide the same decision function. But the equaliser in (4.19) can be implemented with lower computational complexity as it can take advantage of the regular array of the channel states and the time shifting property of the equaliser input. A comparison of the computational complexity of these two forms of equalisers is presented in Table 4.2. The second part of this Table presents the specific computation involved in estimation of each sample in a system with $n_c = n_{c1} = 3$ and $m = 5$.

From Table 4.2 it is evident that the normalised form of Bayesian equaliser with scalar states (NBSS–CCI) provides major computational advantages in the computation of addition, division and exp. The increase in multiplications is very little compared to computational savings for other operations.

Bayesian CCI (4.15)	Computation Aspects	NBSS–CCI (4.19)
$2mN_sN_{s,co}$	Addition	$N_sN_{s,co} + M_0M_1$
$mN_sN_{s,co}$	Multiplication	$mN_sN_{s,co} + M_0M_1$
$N_sN_{s,co} + 1$	Division	$M_0M_1 + 1$
$N_sN_{s,co}$	exp	M_0M_1
$163,840$	Addition	$16,448$
$81,920$	Multiplication	$81,984$
$16,385$	Division	65
$16,384$	exp	64

$M = 2^{n_c}$, $M_1 = 2^{n_{c1}}$,$N_s = 2^{n_c+m-1}$ and $N_{s,co} = 2^{n_{c1}+m-1}$

Table 4.2: *Computational complexity comparison for alternate implementations of Bayesian–CCI equaliser. The second part represents specific computational requirements for* $n_c = n_{c1} = 3, m = 5$

From the above discussion it is seen that NBSS–CCI provides implementation advantages compared to the Bayesian–CCI equaliser. In-line with the fuzzy implementation of NBESS the

NBSS–CCI can also be implemented with fuzzy systems. The computational complexity for the optimal Bayesian–CCI and the NBSS–CCI equalisers is very large for real time implementation. The use of the minimum inference rule and the maximum defuzzification process can reduce the computational complexity of fuzzy implementation of NBSS–CCI equaliser still further. But the computational complexity remains very large for real time implementation. The MLSE designed to remove CCI will be computationally more complex than the Bayesian–CCI equaliser [92]. A MLSE treating CCI as AWGN can be designed with low computational complexity but the performance of this equaliser degrades at low SIR's and there is further performance deterioration in fading channels. These issues relating to the performance comparison between the Bayesian equaliser and MLSE in a CCI environment have been analysed in [92]. For this reason a computationally efficient fuzzy equaliser for this problem is proposed in the next section and this equaliser is termed as fuzzy–CCI equaliser.

4.4 Fuzzy implementation of the NBSS–CCI

The NBSS–CCI decision function was derived in section 4.3. This equaliser provides the Bayesian–CCI equaliser implementation with reduced computational complexity. The NBESS was derived in section 3.3 and it was implemented with fuzzy systems in section 3.4. This equaliser efficiently implements Bayesian equaliser for ISI channels. Similar to the fuzzy implementation of NBESS the NBSS–CCI can also be implemented with fuzzy systems. For this the NBSS–CCI can be described with following equations.

$$\mathfrak{F}\{\mathbf{r}(k)\} = \frac{\sum_{i=1}^{N_s}\sum_{\alpha=1}^{N_{s,co}} w_i \left\{\prod_{l=0}^{m-1} \phi_{il}^{\alpha}\right\}}{\sum_{i=1}^{N_s}\sum_{\alpha=1}^{N_{s,co}} \left\{\prod_{l=0}^{m-1} \phi_{il}^{\alpha}\right\}} \tag{4.20}$$

where ϕ_{il}^{α} is the membership from scalar centres. This membership function can be presented as

$$\phi_{il}^{\alpha} \in \psi_l^{j,\alpha 1} \tag{4.21}$$

$$\psi_l^{j,\alpha 1} = \exp\left[-\left(\frac{|r(k-l) - \{C_j + C_{co,\alpha 1}\}|^2}{2\sigma_\eta^2}\right)\right] \tag{4.22}$$

where, $1 \leq i \leq N_s$, $1 \leq \alpha \leq N_{s,co}$, and $0 \leq l \leq (m-1)$. ϕ_{il}^{α}, $1 \leq j \leq M$ and $1 \leq \alpha 1 \leq M_{co}$, is the membership function corresponding to the $(l+1)$ component of the channel state i and co-channel state α. These channel state components belong to the j th and $\alpha 1$ th scalar channel and co-channel states respectively. The decision function in (4.20) is a fuzzy system with $N_s N_{s,co}$ fuzzy IF ... THEN ... inference rules with product inference, centroid defuzzifier and Gaussian membership function. This equaliser has all the properties of the fuzzy equaliser discussed in section 3.4. The computational complexity of this equaliser is similar to the NBSS–CCI equaliser and can be further reduced by the use of minimum inference and maximum defuzzification processes. The process of subset centre selection can also be applied to this equaliser. However, like the NBSS–CCI this equaliser is also computationally complex and practically difficult to realise.

Here a modified form of the fuzzy equaliser designed for ISI channels is presented. In order to derive this, the fuzzy equaliser derived in section 3.4 is considered first,

$$\mathfrak{F}\{\mathbf{r}(k)\} = \frac{\sum_{i=1}^{N_s} p_i \left\{ \prod_{l=0}^{m-1} \phi_{il} \right\}}{\sum_{i=1}^{N_s} \left\{ \prod_{l=0}^{m-1} \phi_{il} \right\}} \tag{4.23}$$

$$\psi_l^j = \exp\left\{ -\frac{1}{2} \left(\frac{r(k-l) - C_j}{\sigma_\eta} \right)^2 \right\} \tag{4.24}$$

$$\phi_{il} \in \psi_l^j \tag{4.25}$$

It has been seen that the presence of CCI creates more states. The co-channel states surround the channel states. Similarly the presence of CCI increases the number of the noise free received scalars called the scalar channel states. These would be the scalar co-channel states surrounding the scalar channel states. Now the noise free received scalars can be represented as

$$C_i + C_{co,1}, C_i + C_{co,2}, \ldots, C_i + C_{co,M_{co}} \quad \text{where } 1 \leq i \leq M$$

The presence of CCI increases the number of noise free received samples by a factor equal to the number of scalar co-channel states. With this understanding the membership function in (4.24) can be conveniently modified to provide suboptimal CCI compensation. This modified

membership function can be represented as

$$\psi_l^j = \sum_{\alpha=1}^{M_{co}} \left\{ \exp\left(-\frac{|r(k-l) - (C_j + C_{co,\alpha})|^2}{2\sigma_\eta^2} \right) \right\} \tag{4.26}$$

where, the membership function in (4.24) has been modified to find the sum of the membership functions corresponding to all the scalar co-channel states associated with each of the scalar channel states. This membership function in conjunction with the equaliser presented in (4.23) can provide suboptimal co-channel compensation. Another form of the membership function that can also be used is,

$$\psi_l^j = \max_{\alpha=1}^{M_{co}} \left\{ \exp\left(-\frac{|r(k-l) - (C_j + C_{co,\alpha})|^2}{2\sigma_\eta^2} \right) \right\} \tag{4.27}$$

where, the membership function evaluation is based on the maximum of the co-channel membership functions corresponding to each of the scalar channel states. This membership function in (4.27) has implementional advantages compared to (4.26) and can be efficiently implemented with

$$\psi_l^j = \exp\left\{ -\frac{\left\{ \min_{\alpha=1}^{M_{co}} \left[|r(k-l) - (C_j + C_{co,\alpha})| \right] \right\}^2}{2\sigma_\eta^2} \right\} \tag{4.28}$$

where the distance between the received scalars and the scalar channel states offset with the scalar co-channel states is first estimated and the minimum of the distances corresponding to the co-channel states is squared and passed through the exponential function after normalisation with the noise variance. This is the same as finding the maximum $\exp(.)$ of the distance from input scalar to the set of co-channel states corresponding to each of scalar channel states. From simulation studies it has been seen that the membership functions with (4.26) and (4.27) provide similar performances. But (4.27) can be implemented using (4.28) with minimum complexity. With this the fuzzy equaliser consist of N_s fuzzy IF ... THEN ... rules with product inference which are generated from the channel states information, membership function given by (4.28) and COG defuzzifier.

4.4.1 Fuzzy–CCI implementation

The schematic of the fuzzy–CCI equaliser discussed here is shown in Figure 4.5. The input scalar is processed by the membership function generator, whose centres are positioned at the scalar channel states. Each of the membership function sub-blocks generates the membership function from only one of the scalar co-channel states corresponding to each of the scalar channel states. The membership function generation is presented in (4.28). The output of the membership function generator is delayed and this forms the membership function for previously received signal samples. The product block has N_s sub-blocks and each of these sub-blocks receives membership functions from one of the centres corresponding to each input scalar. These membership functions are suitably combined to provide the modified channel state output. The membership function generators consist of M membership function sub-blocks. Each of the sub-blocks has M_{co} centres. The nearest co-channel state in a sub-block w.r.t. the input scalar provides the membership function to the product block. The product blocks corresponding to positive channel states are added to provide 'a' and those corresponding to negative channel states are added to provide 'b'. The equaliser decision function is represented as $\frac{a-b}{a+b}$.

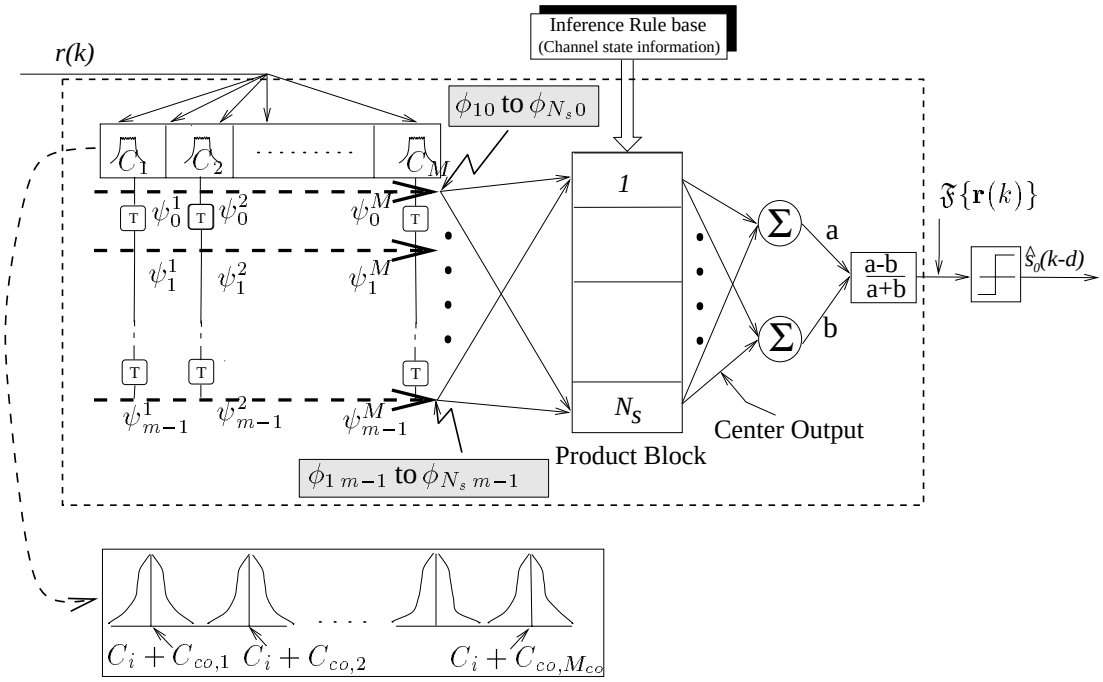


Figure 4.5: *Schematic of fuzzy–CCI equaliser*

An example is considered below to show the effect of membership function modification in

CCI mitigation.

EXAMPLE 4.2

This example takes into account the channel and co-channels considered in the Example 4.1. The channel used is

$$H(z) = H_1(z) = 0.5 + 1.0z^{-1}$$

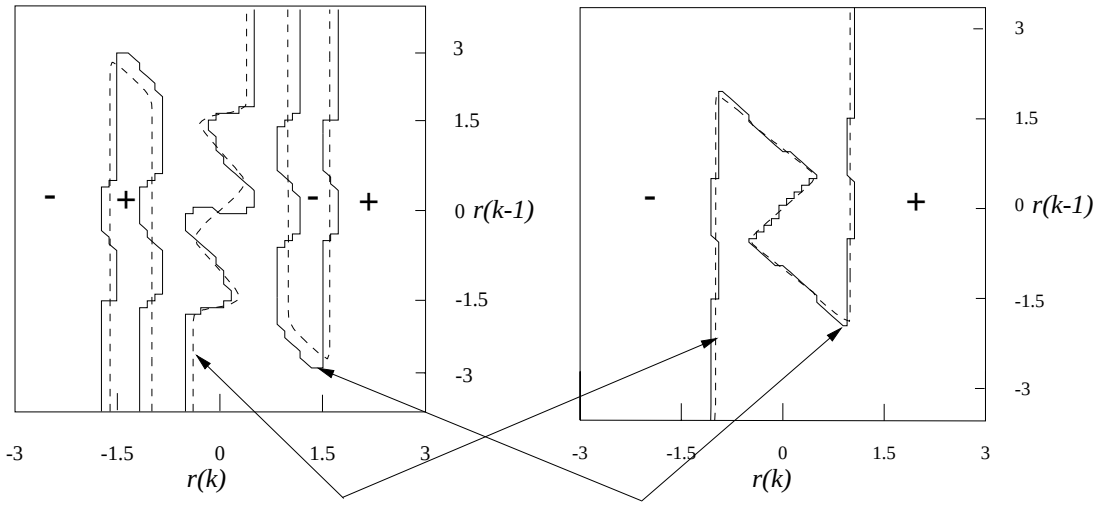
corrupted with CCI from

$$H_{co\_1}(z) = \lambda H_2(z) = \lambda(1.0 + 0.2z^{-1})$$

where, the scaling factor λ controls the SIR. A selection of $\lambda = 0.55142$ provides SIR=5 dB and $\lambda = 0.3467$ provides SIR=10 dB. Here the decision boundary of the optimal Bayesian–CCI is compared with the fuzzy–CCI equaliser presented in (4.23) using the membership function provided by (4.28). The Bayesian–CCI equaliser uses $N_s = 8$ channel states and each of the channel states is associated with $N_{s,co} = 8$ co-channel states. In all it uses 64 states and it can also be implemented with a RBF network using 64 centres. The fuzzy–CCI equaliser uses $N_s = 8$ fuzzy IF ... THEN ... rules with product inference and centroid defuzzifier. The fuzzy IF ... THEN ... rules are generated from the channel state information. The decision boundaries provided by the equalisers for SIR of 5 dB and 10 dB are presented in Figure 4.6(a) and 4.6(b) respectively. The decision regions corresponding to $\widehat{s}(k) = +1$ are marked with $+$ and the decision region corresponding to $\widehat{s}(k) = -1$ are marked with $-$ signs.

The effect of SNR on the decision boundary of an equaliser was presented in Example 2.2, where the effect of channel noise on the equaliser decision boundary was analysed and the change of optimal decision boundary for change in system SNR was also presented. Subsequently, Example 4.1 presented the effect of SIR on the decision boundary. It was seen that at SIR=10 dB the optimal decision boundary is similar to the decision boundary presented in Figure 2.9 which corresponds to the decision boundary without CCI.

From the decision boundary curves in Figure 4.6 it is seen that the fuzzy equaliser with modified membership function for CCI compensation, provides a decision boundary which is close to the optimal Bayesian–CCI equaliser decision boundary. The Bayesian CCI equaliser decision function consists of 64 channel and co-channel state combinations



Fuzzy-CCI (a)SIR=5 dB($\lambda = 0.55$)

Bayesian-CCI (a)SIR=10 dB($\lambda = 0.31$)

Figure 4.6: *Comparison of decision boundaries formed by fuzzy–CCI equaliser and Bayesian–CCI equaliser for channel* $H(z) = 0.5 + 1.0\,z^{-1}$ *and co-channel* $H_{co\text{-}1}(z) = \lambda(1.0 + 0.2z^{-1})$

whereas the fuzzy–CCI equaliser decision function consists of 8 rules with 4 scalar co-channel states associated with each of the scalar channel states for membership function generation. From the decision boundary it is seen that the fuzzy equaliser with a modified membership function can provide a near optimal decision function for channels with CCI using a similar number of channel states as the NBESS. However, NBESS treating CCI as AWGN would fail under a severe SIR condition using a similar architecture.

From the above example it is seen that the fuzzy–CCI equaliser can provide performance close to the Bayesian–CCI equaliser with a complexity similar to the NBESS. The computational complexities of the Bayesian–CCI equaliser, fuzzy–CCI equaliser and the Bayesian equaliser for ISI channels are presented in Table 4.3. From this Table it is seen that the computational complexity of the fuzzy–CCI equaliser is slightly lower than Bayesian equaliser that treats CCI as AWGN. The second part of this Table presents the specific computational requirements for estimation of each of the samples when $n_c = 3$, $n_{c1} = 3$, $m = 5$ which provides $N_s = 64$ and $N_{s,co} = 64$.

Computation Aspects	Bayesian–CCI (4.19)	Fuzzy–CCI (4.23, 4.35)	Bayesian (3.17)
Addition	$2mN_sN_{s,co}$	$N_s + MM_{co}$	$2mN_s$
Multiplication	$mN_sN_{s,co}$	$mN_s + M$	mN_s
Division	$N_sN_{s,co} + 1$	$M + 1$	N_s
Exponentiation	$N_sN_{s,co}$	M	N_s
Addition	$40,960$	128	640
Multiplication	$20,490$	328	320
Division	4096	9	64
Exponentiation	4096	8	64

$M = 2^{n_c},\ N_s = 2^{n_c+m-1},\ M_1 = 2^{n_{c1}},\ N_{s,co} = 2^{n_{c1}+m-1},\ N_f = 2^q$

Table 4.3: *Computational complexity comparison for the Bayesian–CCI, the Fuzzy–CCI and the Bayesian equalisers*

4.5 Decision feedback in CCI equalisers

The past decisions of the equaliser can be fed back to provide the DFE structure. The structure of the DFE used here is presented in Figure 4.7. This equaliser uses the information contained in the observed channel output vector $\mathbf{r}(k)$ and the past detected symbol vector

$$\widehat{\mathbf{s}}_f(k) = [\widehat{s}_0(k-d-1), \widehat{s}_0(k-d-2), ..., \widehat{s}_0(k-d-q)]^T \tag{4.29}$$

to estimate $\widehat{s}_0(k-d)$. Here q is the equaliser feedback order. Without loss of generality, the equaliser parameters can be selected as [53] $d = n_c - 1$ to cover the entire channel dispersion with $m = d + 1 = n_c$ and $q = n_c + m - d - 2 = n_c - 1$.

When decision feedback is employed, the feedback vector $\widehat{\mathbf{s}}_f(k)$ can assume one of $N_f = 2^q$

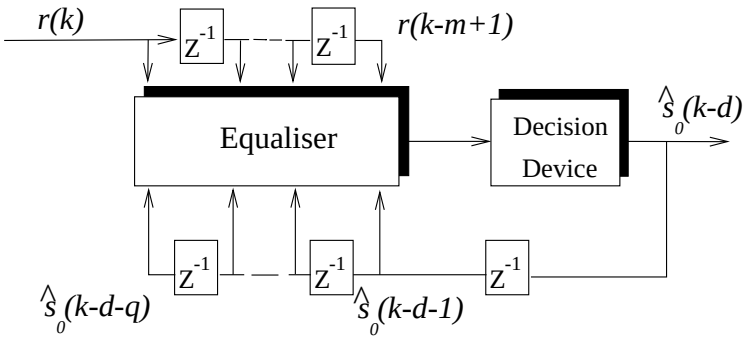


Figure 4.7: *Schematic of a DFE*

states, and the equaliser forms the decision based on $\frac{N_s}{N_f}$ channel states for each of the feedback states [53]. Thus the N_s channel states in (4.14) can be grouped into N_f subsets based on the feedback state and each of the feedback states contain $N_{sf} = \frac{N_s}{N_f}$ states.

$$\bigcup_{i=1}^{N_s} \mathbf{c}_i = \bigcup_{j=1}^{N_f} \bigcup_{l=1}^{N_{sf}} \mathbf{c}_l^j \tag{4.30}$$

where, $\bigcup$ represents the union operation and (j)th corresponding to the feedback state and l corresponding to the channel state in each of the feedback states. With this, the process of decision feedback with Bayesian equalisers can be considered as a process of subset state selection resulting in a reduction of computational complexity. The DFE with linear feed forward filter[2] is a process where ISI associated with the detected samples is cancelled with the feedback filter [43].

The Bayesian–CCI equaliser with decision feedback can be represented as

$$\mathfrak{F}_{CCIDFE}\{\mathbf{r}(k) \mid \widehat{\mathbf{s}}_f(k) = s^j\} = \sum_{i=1}^{N_{sf}} \sum_{l=1}^{N_{s,co}} w_i \exp\left(\frac{-\|\mathbf{r}(k) - \mathbf{c}_i^j - \mathbf{c}_{co,l}\|^2}{2\sigma_\eta^2}\right) \tag{4.31}$$

This equaliser is termed as Bayesian–CCIDFE. Here the term $\mathbf{c}_i^j$ corresponds to the channel state i for feedback state j and $1 \leq i \leq N_{sf}$ and $1 \leq j \leq N_f$. This forms the optimum symbol-by-symbol DFE decision function for a CCI channel. In a similar way the NBSS–CCIDFE can be represented as

$$\mathfrak{F}_{CCIDFE}\{\mathbf{r}(k) \mid \widehat{\mathbf{s}}_f(k) = s^j\} = \frac{\displaystyle\sum_{i=1}^{N_{sf}} \sum_{\alpha=1}^{N_{s,co}} w_i \left\{\prod_{l=0}^{m-1} \exp\left(-\frac{\|r(k-l) - c_{il}^j - c_{co,\alpha l}\|^2}{2\sigma_\eta^2}\right)\right\}}{\displaystyle\sum_{i=1}^{N_{sf}} \sum_{\alpha=1}^{N_{s,co}} \left\{\prod_{l=0}^{m-1} \exp\left(-\frac{\|r(k-l) - c_{il}^j - c_{co,\alpha l}\|^2}{2\sigma_\eta^2}\right)\right\}} \tag{4.32}$$

where c_{il}^j corresponds to the $l+1$ component of the vector channel state $\mathbf{c}_i$, corresponding to

[2] This equaliser is referred to as the nonlinear equaliser in the communication literature.

the feedback state j, and $1 \leq j \leq N_f$ and $c_{co,\alpha l}$ corresponds to the $(l+1)$ component of vector co-channel state $\mathbf{c}_{co,\alpha}$. Each of the components of channel and co-channel states are taken from the set of $M = 2^{n_c}$ scalar channel and $M_{co} = 2^{n_{c1}}$ scalar co-channel states. The normalised form of the equaliser presented in (4.31) and the equaliser in (4.32) provide the same decision function but the equaliser in (4.32) can be implemented with lower computational complexity like equalisers without decision feedback. The components of channel and co-channel states belong to the scalar channel and co-channel states

$$c_{il} \in C_j \quad \text{where } 1 \leq i \leq N_{sf},\ 0 \leq l \leq (m-1),\ 1 \leq j \leq N_f \text{ and}$$
$$\mathbf{c}_{co,\alpha l} \in C_{co,\alpha 1} \quad \text{where } 1 \leq \alpha \leq N_{s,co},\ 0 \leq l \leq (m-1),\ 1 \leq \alpha 1 \leq M_{co}$$

where, $\alpha 1$ is a single index and the terms have their usual meanings. The fuzzy–CCIDFE equaliser can be presented as

$$\mathfrak{F}\{\mathbf{r}(k)|\widehat{\mathbf{s}}_f(k) = s^j\} = \frac{\sum_{i=1}^{N_{sf}} w_i \left\{ \prod_{l=0}^{m-1} \phi_{il}^j \right\}}{\sum_{i=1}^{N_s} \left\{ \prod_{l=0}^{m-1} \phi_{il}^j \right\}} \tag{4.33}$$

$$\phi_{il}^j \in \psi_l^\alpha \tag{4.34}$$

$$\psi_l^\alpha = \exp\left(- \frac{\left\{ \min_{\alpha 1=1}^{M_{co}} \left[|r(k-l) - (C_\alpha + C_{co,\alpha 1})| \right] \right\}^2}{2\sigma^2} \right) \tag{4.35}$$

where σ^2 is optimised to provide the best performance. Under high CCI (low SIR) this can be set to σ_η^2 and under low CCI (high SIR) this can be set to $\sigma_\eta^2 + \sigma_{co}^2$.

4.5.1 Fuzzy implementation (Fuzzy–CCIDFE)

The fuzzy–CCIDFE can be implemented in a similar way to the fuzzy–CCI equaliser which is presented in Figure 4.5. For DFE implementation the fuzzy rules that form the rule base for the inference system consist of N_f groups of rules each with N_{sf} rules, unlike the equaliser in Figure 4.5 which has N_s rules. Depending on the feedback state a set of rules are used for decision function calculation.

4.6 Fuzzy CCI equaliser: Implementation issues

This section analyses the training issues and the computational complexities related to the fuzzy–CCIDFE. Here the training issues are considered first.

4.6.1 Adaptive implementation

The fuzzy–CCI equaliser and fuzzy–CCIDFE require the knowledge of equaliser channel states and the scalar co-channel states. The equaliser design also requires the knowledge of channel noise statistics. The channel states of the equaliser can be estimated from the scalar channel states as discussed in Chapter 3. The process of estimation of parameters for the Bayesian–CCIDFE have been analysed in [92]. The problem associated with this equaliser training is the estimation of the co-channel states. The co-channel states of the Bayesian–CCI equaliser can be estimated using unsupervised clustering. This technique requires long training sequence and in addition convergence is not guaranteed. The co-channel states can also be estimated from scalar co-channel states. The scalar co-channel states can be estimated with an unsupervised clustering algorithm and observation of the state transitions can provide the channel states [92]. This scheme could also require a long training sequence, particularly under poor SNR conditions. The fuzzy–CCI equalisers reported here do not require the co-channel states but only the scalar co-channel states, which are fairly simple to estimate with an unsupervised clustering algorithm. The fuzzy–CCIDFE discussed above can be trained in 2 steps. The first step in training involves estimation of the scalar channel and scalar co-channel states and the second step involves learning weights with the LMS algorithm.

Step-I: Determination of channel and co-channel states The scalar channel and scalar co-channel states of the equaliser can be estimated by the κ-means clustering algorithm. The equaliser channel states can be estimated from scalar channel states. This process of the estimating scalar channel states and forming of the channel states from these has been analysed in section 3.5. The estimation of channel states with supervised clustering process can provide $\sigma_{\eta}^2 + \sigma_{co}^2$. Subsequently the scalar co-channel states can be estimated. Here the estimation of scalar co-channel states is analysed.

Co-channel states: Once the channel states have been determined the channel residual $r_{res}(k) = r(k) - C_j$ (here C_j refers to the scalar channel state j) can be estimated. The channel residual arises from the CCI and AWGN. An unsupervised clustering algorithm such as the κ-

means or enhanced κ-means [151] clustering algorithm can provide the scalar co-channel states and the noise variance (σ_η). The process of the estimation of scalar channel states using the κ-means algorithm and the estimation of scalar co-channel states using the enhanced κ-mean algorithm have been discussed in detail in Appendix A.

Step-II Weight training On completion of the channel and co-channel scalar state estimation, the equaliser can be constructed (Figure 4.5). The initial weights (w_i) of the equaliser can be assigned $+1$ if $\mathbf{c}_i \in \mathbf{C}_d^+$ else they can be assigned -1. The LMS algorithm presented in (3.5) can be used to fine tune the equaliser weights so as to reduce the error at the equaliser output due to the channel states estimation error.

The process of training the fuzzy–CCI and its decision feedback form is quick as the number of scalar channel states and scalar co-channel states are small. The estimation of scalar channel states, scalar co-channel states and the weight training can be done in sequence one after the other. The same set of training sequence can be reused for all the three procedures to maximise its use.

4.6.2 Advantages of fuzzy–CCI and fuzzy–CCIDFE

The fuzzy–CCI equaliser and fuzzy–CCIDFE presented in this chapter has several advantages over the Bayesian–CCI (4.19) equaliser. These advantages are listed below.

- The fuzzy–CCI equaliser can provide near optimal performance with substantial reduction in computational complexity. The computational complexities of the fuzzy–CCI, the Bayesian–CCI and the Bayesian equalisers were presented in Table 4.3. The computational complexity of the respective decision feedback equalisers is presented in Table 4.4. From this Table it can be seen that the complexity of the fuzzy–CCIDFE is comparable to the Bayesian–DFE that treats CCI as noise. The Bayesian–CCIDFE is difficult to implement in real time applications. The second part of the Table 4.4 presents the specific computational requirements when $n_c = 3$, $n_{c1} = 3$ with which the parameters are set to $m = 3$, $d = 2$ and $q = 2$.

- The structures of the fuzzy–CCI equaliser and fuzzy equaliser for ISI channels are the same, excluding the membership function generation which is the input processor in the equaliser. This makes the equaliser very flexible. The co-channel compensation module

in the form of membership function modification can be introduced when the SIR drops below acceptable limits.

- The scalar channel and co-channel states provide a suitable method of finding the condition under which co-channel compensation is not required. If

$$\min_{\substack{i=1 \\ j=i}}^{\substack{i=M-1 \\ j=M}} (c_i - c_j) \geq 2 \max_{\alpha=0}^{M_{co}} C_{co,\alpha} \tag{4.36}$$

co-channel compensation is not required. In this inequality, the left hand side represents the smallest distance between any two scalar channel states and the right hand side represents the maximum scalar co-channel state corresponding to any channel state. If this condition is not true then co-channel compensation in the form of membership modification should be used.

Computation Aspects	Bayesian–CCIDFE (4.31)	Fuzzy–CCIDFE (4.33, 4.35)	Bayesian–DFE
Addition	$2mN_{sf}N_{s,co}$	$N_{sf} + MM_{co}$	$2mN_{sf}$
Multiplication	$mN_{sf}N_{s,co}$	$mN_{sf} + M$	mN_{sf}
Division	$N_{sf}N_{s,co}$	$M + 1$	N_{sf}
Exponentiation	$N_{sf}N_{s,co}$	M	N_{sf}
Addition	1532	72	48
Multiplication	768	32	24
Division	256	9	8
Exponentiation	256	8	8

$M = 2^{n_c},\ N_s = 2^{n_c+m-1},\ M_1 = 2^{n_{c1}},\ N_{s,co} = 2^{n_{c1}+m-1},\ N_f = 2^q$

Table 4.4: *Computational complexity comparison of Bayesian–CCIDFE, Fuzzy–CCIDFE and Bayesian–DFE, Second part represents the specific computational complexity requirement when* $n_c = n_{c1} = 3,\ m = 3,\ d = 2,\ q = 2$, *providing* $N_s = 32$, $N_f = 4$, *and* $N_{sf} = 8$

- Training the fuzzy–CCI equaliser is simple as it uses scalar unsupervised clustering for the co-channel state estimate. But, the Bayesian–CCI requires unsupervised vector clustering for the co-channel states estimation. These aspects were discussed in sub-section 4.6.1. In the presence of more than one co-channel the estimation of the co-channel states is very difficult as the number of co-channels becomes very large. Simulation studies suggest estimation of scalar co-channel states is relatively simple.

The decision making capacity of fuzzy–CCIDFE is analysed here with an example.

EXAMPLE 4.3

The decision making capability of fuzzy–CCIDFE for CCI channels is analysed here. The channels used in this study were,

$$H(z) = H_1(z) = 0.5 + 1.0z^{-1}$$

corrupted with CCI from the channel

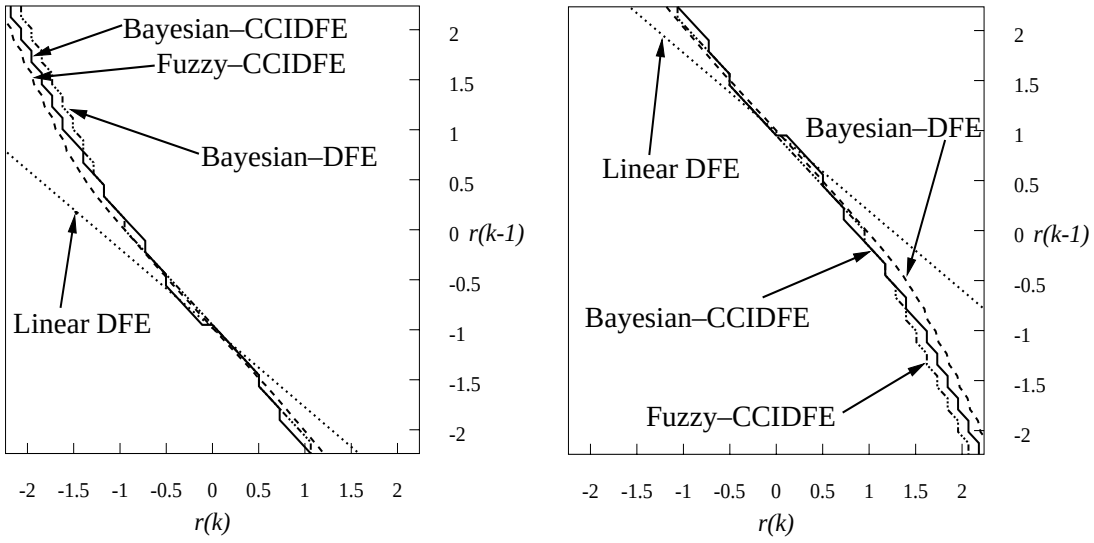
$$H_{co\_1}(z) = \lambda H_2(z) = \lambda(1.0 + 0.2z^{-1})$$

where λ controls the SIR. The system SNR=15 dB. The equaliser length $m = 2$, delay $d = 1$ and feedback order $q = 1$ are selected for optimum performance. These parameters provide $N_f = 2$ feedback states corresponding to $\widehat{s}(k-2) = -1$ and $\widehat{s}(k-2) = 1$. The number of channel states are $N_{sf} = 4$ and $N_{s,co} = 8$. The decision making capacity of 4 forms of equalisers are analysed here. These equalisers are

- Bayesian–CCIDFE
- Bayesian–DFE
- fuzzy–CCIDFE and
- linear DFE

The Bayesian–CCIDFE uses 4 out of 8 channel states corresponding to each feedback state. Each of these channel states is surrounded by 8 co-channel states. This corresponds to using 32 channel states in the decision functions for estimation of each symbol. In a RBF implementation this would require 32 centres corresponding to each feedback state. The Bayesian–DFE is the Bayesian equaliser with decision feedback that treats CCI as noise. This equaliser uses 4 out of 8 channel states corresponding to each of the feedback states. Similarly, the fuzzy–CCIDFE uses only 4 out of 8 fuzzy IF ... THEN ... rules derived from the channel states corresponding to each feedback state and the membership function of the equaliser is determined with (4.35). The membership function block uses 4 co-channel states corresponding to each of the scalar channel states for calculating the membership function in the presence of CCI. The equaliser uses the estimated scalar channel states. These channel states are estimated with a supervised clustering algorithm and the scalar co-channel states are estimated with an unsupervised clustering algorithm.

The linear DFE uses the LMS training algorithm to train the weights of the equaliser. The performance of these equalisers for a SIR of 10 dB and 4 dB is presented in Figure 4.8 and Figure 4.9 respectively.

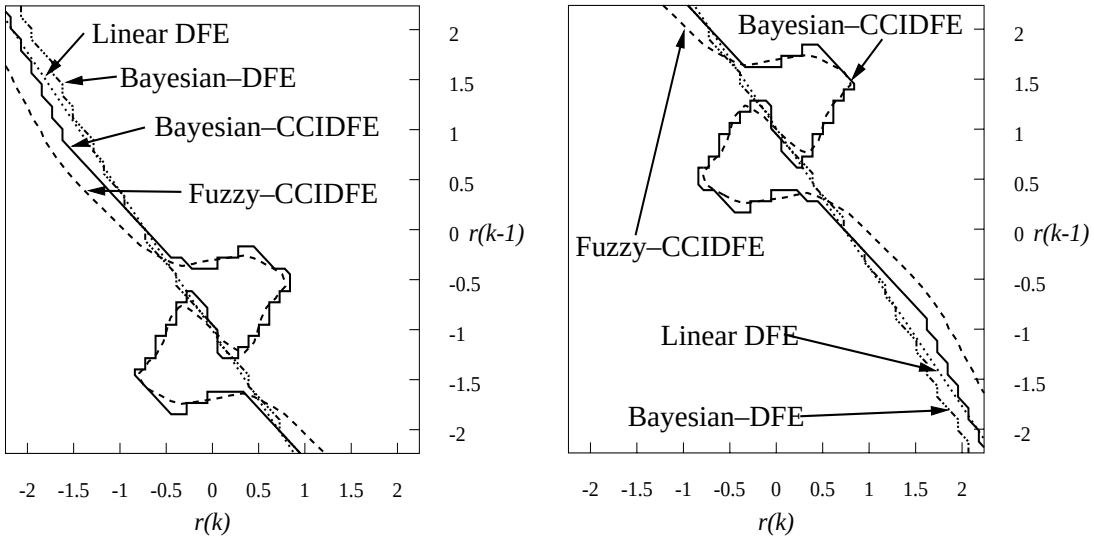


(a) $\widehat{s}(k-2) = -1$ **(b)** $\widehat{s}(k-2) = +1$

Figure 4.8: *Comparison of decision boundaries for DFE equalisers with channel* $H(z) = 0.5 + 1.0z^{-1}$ *and co-channel* $H_{co\_1}(z) = (1.0 + 0.2z^{-1})$ *for SIR=10 dB and SNR=15 dB with* $m = 2$, $d = 1$, $q = 1$

Figure 4.8(a) and 4.8(b) represents the decision boundaries for feedback signal corresponding to $\widehat{s}(k-2) = -1$ and $\widehat{s}(k-2) = +1$ respectively for SIR=10 dB. From Figure 4.8 it is seen that the optimum equaliser Bayesian–CCIDFE decision boundary is nearly linear and both the fuzzy–CCIDFE and Bayesian–DFE provide a decision boundary which is very close to the optimal. The partitioning of the channel and associated co-channel states show that the states $\mathbf{c}_i + \mathbf{c}_{co,\alpha} \in \mathbf{C}_d^+$ and $\mathbf{c}_i + \mathbf{c}_{co,\alpha} \in \mathbf{C}_d^-$ are linearly separable. The linear LMS equaliser also provides a decision boundary which is close to the optimal. From these observations it can be inferred that at this SIR=10 dB CCI can be treated as AWGN for equaliser design.

In the next stage, the SIR was reduced to 4 dB. The equaliser decision boundaries for $\widehat{s}(k-2) = -1$ and $\widehat{s}(k-2) = +1$ are presented in Figure 4.9(a) and Figure 4.9(b) respectively. From the decision boundaries it is observed that the decision boundary provided by the Bayesian–CCIDFE is now nonlinear. The partitioning of the channel and associated co-channel states show that the states $\mathbf{c}_i + \mathbf{c}_{co,\alpha} \in \mathbf{C}_d^+$ and $\mathbf{c}_i + \mathbf{c}_{co,\alpha} \in \mathbf{C}_d^-$



(a) $\widehat{s}(k-2) = -1$ **(b)** $\widehat{s}(k-2) = +1$

Figure 4.9: *Comparison of decision boundaries for DFE equalisers with channel* $H(z) = 0.5 + 1.0z^{-1}$ *and co-channel* $H_{co\_1}(z) = (1.0 + 0.2z^{-1})$ *for SIR=4 dB and SNR=15 dB,* $m = 2$, $d = 1$, $q = 1$

are nonlinearly separable. This nonlinearity can be attributed to severe CCI. Under this condition the LMS linear equaliser and Bayesian–DFE equaliser decision boundaries are similar to the decision boundaries at SIR=10 dB and they fail to equalise the channel. The fuzzy–CCIDFE provides a decision boundary close to the Bayesian–CCIDFE. It is interesting to note that the fuzzy–CCIDFE using only 4, fuzzy IF ... THEN ... rules derived from channel state information with product inference provides a decision boundary close to the one provided by the Bayesian–CCIDFE using 32 states. Further, with similar computational complexities the Bayesian–DFE using 4 channel states fails to provide the required decision boundary.

From this Example it can be resolved that the membership function pre-processor with the fuzzy equaliser provides scope for CCI mitigation.

The following section presents the BER performance of fuzzy equalisers in CCI channels.

4.7 Results and discussion

The BER performance of fuzzy equalisers proposed in this chapter were evaluated with extensive Monte Carlo simulations with a wide variety of channel and co-channel combinations. The transmitted signal $s(k)$ in all experiments were generated randomly from an i.i.d. sequence of $\{\pm 1\}$. The BER performance of equalisers were evaluated by observing 1000 errors in a maximum of 10^8 transmitted samples.

4.7.1 Fuzzy–CCI equaliser

First the BER performance of five forms of equalisers without decision feedback was investigated. These equalisers are

- Bayesian–CCI
- Bayesian equaliser treating CCI as noise
- Fuzzy–CCI equaliser with maximum of co-channel membership functions corresponding to each scalar channel states (4.28)
- Fuzzy–CCI equaliser with sum of co-channel membership functions corresponding to each scalar channel state (4.26)
- Linear equalisers with RLS training algorithm

The channel and the co-channel impulse responses for this experiment were,

$$\begin{aligned} H(z) &= H_5(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2} \\ H_{co\_1}(z) &= H_8(z) = \lambda(0.6 + 0.8z^{-1}) \end{aligned} \tag{4.37}$$

The equaliser parameters were selected as

$m = 4$ and $d = 1$

The SIR was set to 10 dB. The equalisers were designed with knowledge of the channel and the co-channel. One exception was the linear equaliser which was trained with the RLS algorithm using 1000 training samples and the filter weights were averaged over 50 experiments. Other

equalisers did not undergo any training. The BER performance of these equalisers is presented in Figure 4.10. From this Figure it is seen that the linear equaliser and Bayesian equaliser performed very poorly and the BER do not improve beyond $10^{-1.5}$ and $10^{-1.8}$ respectively, irrespective of additive noise power. The fuzzy–CCI equaliser performs close to the Bayesian–CCI equaliser. Here the Bayesian–CCI equaliser uses $N_s = 64$ channel states, each of these channel states is associated with $N_{s,co} = 32$ co-channel states and in all, it uses 2048 states to estimate each of the transmitted samples. The Fuzzy–CCI equaliser uses 64 $\mathrm{IF}\ \ldots\ \mathrm{THEN}\ \ldots$ rules derived from channel state information and the Bayesian equaliser uses 64 states. The Fuzzy–CCI equaliser uses $M = 8$ scalar channel states and each of the scalar channel states is associated with $M_{co} = 4$ scalar co-channel states. With this, it is seen that the modification of the fuzzy equaliser membership function provides an efficient equalisation technique. Here it is also seen that, membership function generation with the sum of co-channel membership functions (4.26) and the maximum of the co-channel membership functions (4.28) provide similar performance. Similar results were also observed for other channel and co-channel combinations with varying SIR's. Based on this, maximum of co-channel membership functions (4.28) were used for implementational advantages in all further investigations.

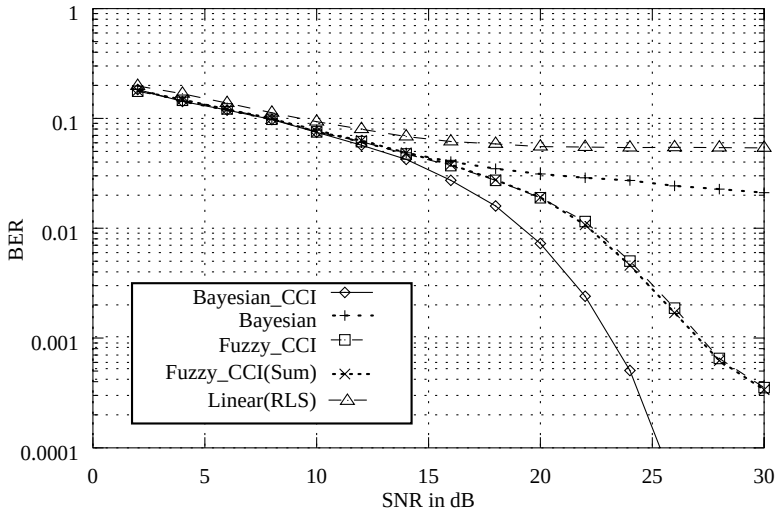


Figure 4.10: *BER performance for different equalisers under SIR=10dB for Channel* $H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$, *Co-channel* $H_{co\_1}(z) = \lambda(0.6 + 0.8z^{-1})$, $m = 4$ *and* $d = 1$ *with the knowledge of channel and co-channel*

4.7.2 Performance with decision feedback

In the next phase of the experiments, the equaliser parameters were estimated with a training signal. Here only decision feedback structures were considered. The DFE parameters were set to

$$m = n_c,\ d = n_c - 1 \text{ and } q = n_c - 1$$

The following types of equalisers were investigated,

- Bayesian–CCIDFE
- Bayesian–DFE treating CCI as noise
- fuzzy–CCIDFE equaliser with maximum of co-channel membership functions corresponding to each scalar channel states (4.28)
- Linear DFE with RLS training algorithm

The channel and the co-channel are characterised by their impulse responses,

$$H(z) = H_8(z) = 0.2294 + 0.4588z^{-1} + 0.688z^{-2} + 0.4588z^{-3} + 0.2294z^{-4} \quad (4.38)$$
$$H_{co\_1}(z) = H_1(z) = \lambda(0.5 + 1.0z^{-1})$$

where the equaliser channel states were first estimated with a supervised κ-means clustering algorithm and subsequently the scalar co-channel states were estimated with the unsupervised enhanced κ-means clustering algorithm. The channel SNR and SIR were also estimated during the training phases. During the supervised clustering process for estimation of scalar channel states, $\sigma_\eta^2 + \sigma_{co}^2$ was estimated and during the scalar co-channel state estimation σ_η^2 was estimated. The estimation error associated with σ_η^2 was high, as it involves an unsupervised clustering algorithm in a noisy environment. These estimated scalar channel and scalar co-channel states were used to construct the fuzzy–CCIDFE equaliser and the scalar channel states were used to construct the Bayesian–DFE equaliser. The Bayesian–DFE can also be treated as the Fuzzy–DFE equaliser with product inference and centroid defuzzifier as discussed in Chapter 3. The Bayesian–DFE used estimated channel states and the channel states spread parameters associated with centres were set to $\sigma_\eta^2 + \sigma_{co}^2$. Both the fuzzy–CCIDFE and the Bayesian–DFE

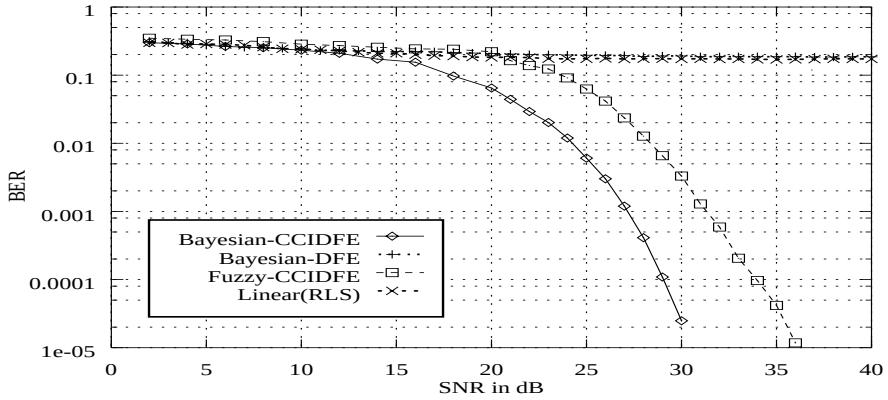
equaliser were trained with 500 training samples and the parameters were averaged over 50 experiments. The linear DFE was trained with 1000 training samples with the RLS algorithm and the equaliser weights were averaged over 50 individual experiments. The equaliser parameters after training were maintained fixed during the transmission period. The Bayesian–CCIDFE used true channel and co-channel states and channel noise statistics which provided the best possible performance of a symbol-by-symbol equaliser under the specified conditions. In line with the discussions in this chapter the equaliser parameters were set to

$$m = 5,\ d = 4 \text{ and } q = 4$$

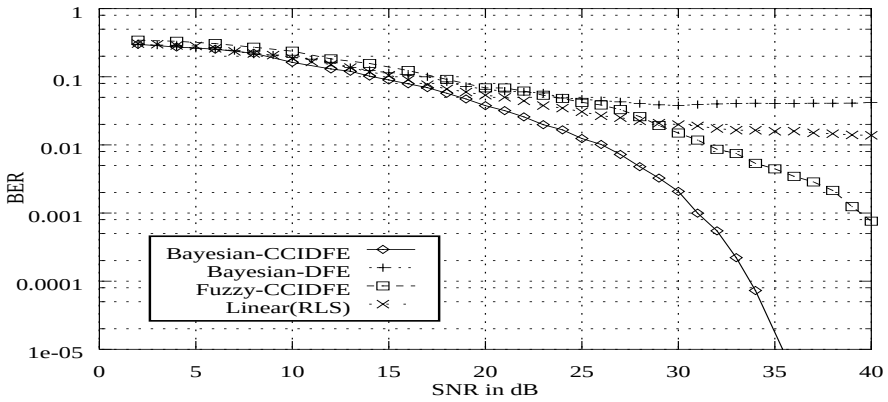
From the channel and co-channel impulse responses it is seen that the equaliser has $M = 32$ scalar channel states and $M_{co} = 4$ scalar co-channel states. The equaliser has $N_{sf} = 32$ channel states corresponding to each of $N_f = 16$ feedback states. The fuzzy–CCIDFE uses 4 scalar co-channel states with each scalar channel state to estimate the membership function corresponding to each of the scalar channel states. The Bayesian–DFE and fuzzy–CCIDFE compute the decision function with 32 channel states out of a total of $N_s = 512$ channel states. The Bayesian–CCIDFE uses 32 channel states out of 512 channel states and each channel state is affected by $N_{s,co} = 64$ co-channel states. With this, the Bayesian–CCIDFE uses $N_{sf}N_{s,co} = 1024$ states to estimate each of the transmitted symbols. This equaliser can be treated as a RBF network with 1024 centres corresponding to each of the 16 feedback states. Each of the centres used by the equalisers are of order five. The fuzzy equaliser membership spread parameter σ was set to σ_η^2, estimated from unsupervised clustering when $\sigma_\eta^2 < \sigma_{co}^2$. It was set to $\sigma_\eta^2 + \sigma_{co}^2$ at other times.

The BER performance with Monte Carlo simulations for the 4 types of equalisers for an SIR=10 dB, 15 dB and 20 dB are presented in Figures 4.11(a), 4.11(b) and 4.11(c) respectively. From the Figure following observations can be made.

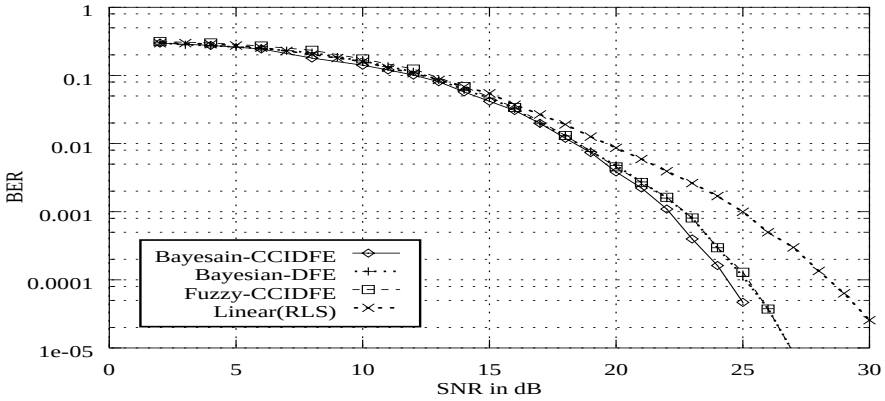
1. For SIR=10 dB the Bayesian–DFE and the linear DFE (with RLS training) fail completely to equalise the channel. It is interesting to note that the fuzzy–CCIDFE provides a performance which is better than Bayesian–DFE but inferior to the Bayesian–CCIDFE. The fuzzy–CCIDFE suffers from a performance degradation of < 6 dB at a BER of 10^{-4} w.r.t the optimal equaliser. The performance degradation can also be partly attributed to the error in estimation of co-channel states and σ_η^2. Another cause for the inferior



(a) SIR=10dB



(b) SIR=15 dB



(c) SIR=20 dB

Figure 4.11: *BER performance of different equalisers with channel* $H(z) = 0.2294 + 0.4588z^{-1} + 0.688z^{-2} + 0.4588z^{-3} + 0.2294z^{-4}$, *co-channel* $H_{co\_1}(z) = \lambda(0.5 + 1.0z^{-1})$, $m = 5$, $d = 4$ *and* $q = 4$ *with estimated channel and co-channel states*

performance of the fuzzy–CCIDFE w.r.t. the Bayesian–CCIDFE is due to the fact that the Bayesian–CCIDFE uses 1024 centres in RBF implementation to estimate each of the samples where as the fuzzy–CCIDFE uses only 32 fuzzy IF ... THEN ... rules derived from channel state information, to estimate each sample. The Bayesian–DFE using 32 states fails to equalise this channel.

2. For SIR=15 dB, the performance of linear DFE and Bayesian–DFE improves compared to SIR=10 dB. But here also the Bayesian–DFE and linear DFE fail to provide a BER performance of better than $10^{-1.6}$ and $10^{-1.8}$ respectively even when the SNR$\rightarrow \infty$. It is also interesting to note that the linear DFE provides better performance than the Bayesian–DFE. The reason for this is that the RLS DFE optimises its weights in the process of training, so that the decision function is in the form of a hyper plane close to the optimal equaliser decision boundary. But the Bayesian–DFE provides a decision function without any optimisation for CCI. The fuzzy–CCIDFE equaliser performs better than the linear DFE and the Bayesian–DFE but its performance is poorer than the Bayesian–CCIDFE. The performance of the Bayesian–CCIDFE and the fuzzy–CCIDFE are inferior to their respective performances at SIR=10 dB. The performance degradation can be attributed to the fact that some of the co-channel states corresponding to positive and negative channel states under this circumstance are very close. When the SIR=10 dB these channel states cross over, leading to increased distance between them, which provides better performance. More simulations results in this context will be presented in the next subsection.

3. For a SIR=20 dB, performance of the Bayesian–DFE, the fuzzy–CCIDFE and the Bayesian–CCIDFE are nearly similar. Under low CCI conditions, the co-channel states are situated very close to the channel states in multidimensional space. Due to this fact, the nonlinear decision boundary provided by fuzzy–CCIDFE and Bayesian–DFE are close to the decision boundary of Bayesian–CCIDFE. The RLS DFE used here provides only a linear approximation of the optimal nonlinear decision surface and hence its performance is the poorest.

4.7.3 Equaliser performance against varying SIR

This subsection examines the equaliser performance against varying levels of SIR with fixed SNR. The problem considered in the previous subsection is considered again. The BER per-

formance of Bayesian–CCIDFE, fuzzy–CCIDFE and Bayesian–DFE at an SNR=25 dB and for varying SIR from 1 dB to 19 dB is presented in Figure 4.12

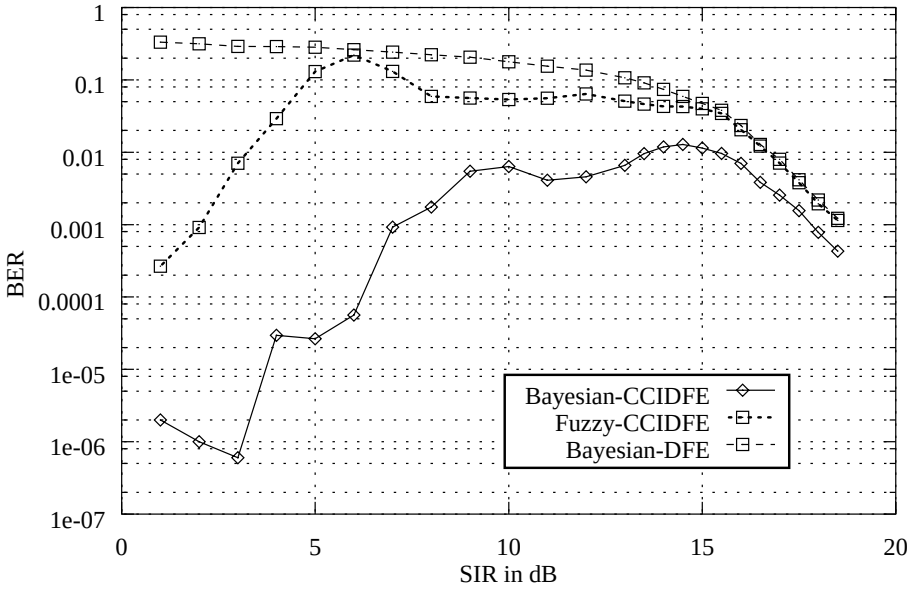


Figure 4.12: *BER performance of different equalisers for varying SIR with Channel* $H_(z) = 0.2294 + 0.4588z^{-1} + 0.688z^{-2} + 0.4588z^{-3} + 0.2294z^{-4}$*, co-channel* $H_{co\_1}(z) = \lambda(0.5 + 1.0z^{-1})$*,* $m = 5$*,* $d = 4$ *and* $q = 4$ *using actual channel and co-channel states under SNR=25 dB*

From the performance curves, it is interesting to note that the BER performance of the optimal equaliser (Bayesian–CCIDFE) worsens with an increase in SIR from 3 dB to 14 dB. The performance of the equaliser is worst at around SIR=14 dB. The equaliser BER improves monotonically after an SIR=14 dB. The fuzzy–CCIDFE performance drops with an increase in SIR between 1 dB to 6 dB. Subsequently the equalisers performs poorly up to a SIR=14 dB. When the SIR improves beyond 14 dB the equaliser performance improves monotonically like the Bayesian–CCIDFE. The Bayesian–DFE equaliser provides very poor performance for SIR=1 dB to 15 dB. Subsequently the performance is close to the Bayesian–CCIDFE. When the SIR is better than 15 dB the performance of all the equalisers is similar. These results validate the performance drop of the Bayesian–CCIDFE and the fuzzy–CCIDFE at SIR=15 dB compared to the performance at SIR=10 dB.

4.7.4 Fuzzy equaliser performance in presence of multiple co-channels

A further experiment considered the performance of the fuzzy equaliser for a channel corrupted with 2 co-channel interferes. Here the channel and co-channel impulse responses are,

$$H(z) = H_5(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$$
$$H_{co\_1}(z) = \lambda H_1(z) = \lambda_1(0.5 + 0.1z^{-1})$$
$$H_{co\_2}(z) = \lambda_1 H_4(z) = \lambda(0.5 + 0.81z^{-1} + 0.31z^{-2}) \tag{4.39}$$

The equaliser parameters are set to

$$m = 3,\ d = 2 \text{ and } q = 2 \tag{4.40}$$

The co-channel power scaling parameters λ and λ_1 were adjusted to divide the interference power equally between both of the co-channels. This system has $N_{sf} = 8$ channel states corresponding to each of $N_f = 4$ feedback states. Each of these channel states is affected by $N_{s,co1}N_{s,co2} = 32 \times 16 = 512$ co-channel states. With this the Bayesian–CCIDFE evaluates 4096 out of 16348 states for estimation of each symbol. The Fuzzy–DFE and the Bayesian–DFE use only 8 channel states corresponding to each of the feedback states. The BER performance of the fuzzy–CCIDFE and the Bayesian–DFE for SIR of 5 dB, 10 dB and 15 dB are investigated. The number of scalar co-channel states used by the fuzzy–CCIDFE in the membership function estimation was limited to 8 instead of the possible $2^{n_{c1}} \times 2^{n_{c2}} = 32$. This could be viewed as an error in the estimation of co-channel order. The optimal Bayesian–CCIDFE performance was not simulated due to its large computational complexity. The BER performance of the fuzzy–CCIDFE and Bayesian–DFE is presented in Figure 4.13. From the simulation results it is seen that the fuzzy–CCIDFE fails under severe CCI (SIR=5 dB) with multiple co-channels. But under moderate CCI (SIR=10 dB) it is able to perform better than the Bayesian–DFE for comparable network complexities. The Bayesian–DFE fails to provide a BER of better than $10^{-1.7}$ even under infinite SNR, but the fuzzy–CCIDFE BER performance shows improvement with an increase in SNR. However for 15 dB SIR the effect of co-channel compensation is minimal and the fuzzy–CCIDFE performs only marginally better. Under this condition co-channel compensation pre-processor can be removed.

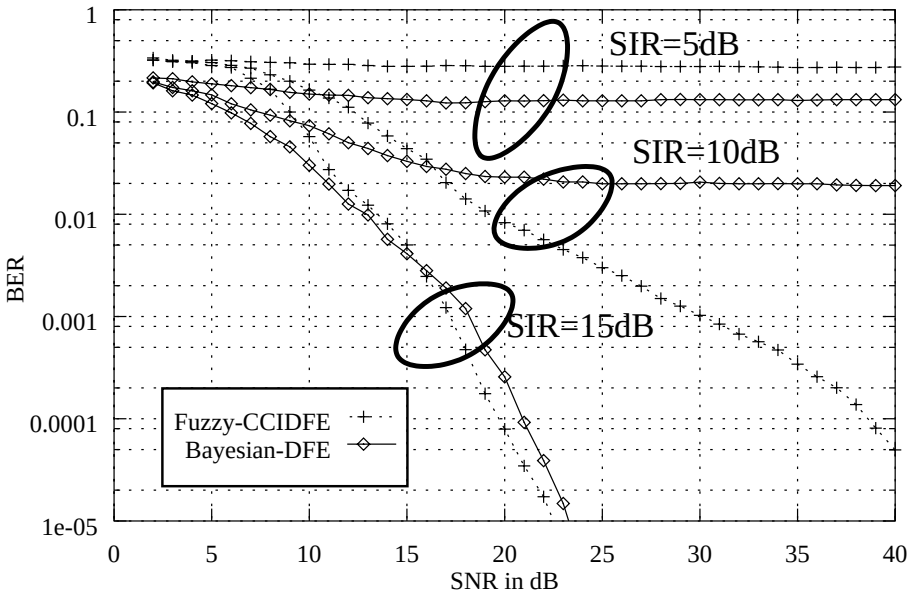


Figure 4.13: *BER performance of fuzzy–CCIDFE and Bayesian–DFE for channel* $H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$*, co-channels* $H_{co\_1}(z) = \lambda(0.5 + 0.81z^{-1} + 0.31z^{-2})$, $H_{co\_2}(z) = \lambda_1(0.5 + 1.0z^{-1})$, $m = 3$, $d = 2$ *and* $q = 2$ *under CCI=5 dB, 10 dB and 15 dB*

4.7.5 Effect of number of estimates of scalar co-channel states

In order to investigate the effect of the number of estimated co-channel states on equaliser BER performance, the number of scalar co-channel states in the unsupervised clustering algorithm was varied in the preceding study and the equaliser BER performance was evaluated. This process can be viewed as an error in estimating the length of the co-channel impulse response. This also provides a limit on the computational complexity of the fuzzy–CCIDFE with respect to performance with variation in the number of co-channel states in membership function estimation. The performance of the fuzzy–CCIDFE for 4, 8, 16 and 32 co-channel states (resulting from estimate of $n_{c1} = 2, 3, 4$ and5 respectively) for 10 dB SIR is presented in Figure 4.14. Here the performance of Fuzzy–DFE equaliser is also presented. The fuzzy–DFE is similar to the Bayesian–DFE equaliser that treats CCI as noise. From the Figure 4.14 it is seen that using a very small number of co-channel states degrades the equaliser performance substantially. With the assumption of 8, 16 or 32 co-channel states, however, the performance tradeoff is small. The performance of the equaliser with fewer number of co-channel states ($n_{c1} = 2$) is closer to the fuzzy–DFE[3] as seen from the figure.

[3]This equaliser is same as the Bayesian–DFE

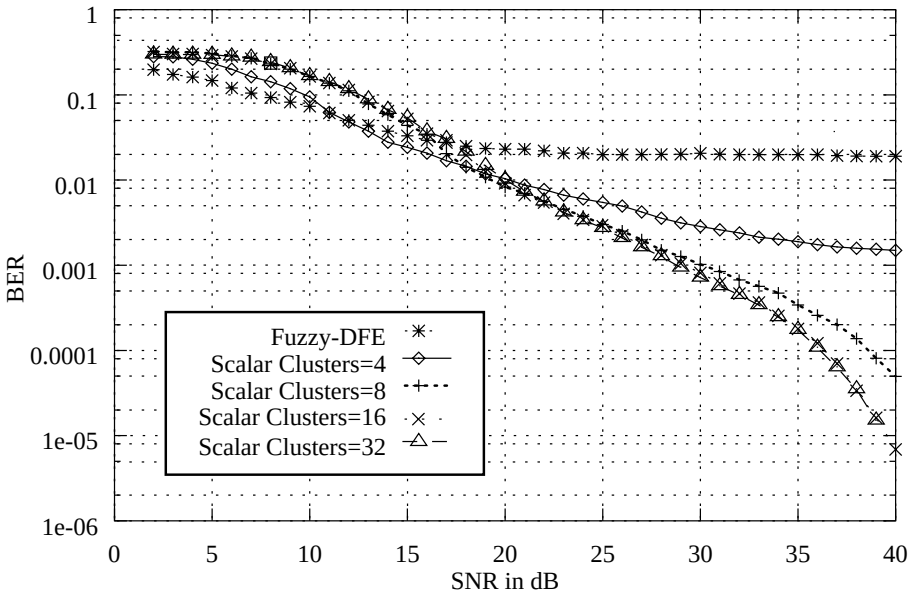


Figure 4.14: *Effect of number of co-channel clusters on equaliser performance for SIR=10 dB, Channel* $H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$*, co-channels* $H_{co\_1}(z) = \lambda(0.5 + 0.81z^{-1} + 0.31z^{-2})$, $H_{co\_2}(z) = \lambda_1(0.5 + 1.0z^{-1})$, $m = 3$, $d = 2$ *and* $q = 2$

From this study it can be inferred that the assumption of co-channel order $n_{c1} \leq 3$ is typically sufficient to provide performance almost as good as achieved by fuzzy–CCI equaliser with true number of co-channel states. The performance gain for $n_{c1} \geq 4$ is very little.

4.7.6 DFE error propagation performance

The last part of the simulation study investigates the error propagation characteristics of the fuzzy–CCIDFE equalisers. Here the error propagation characteristics of fuzzy–CCIDFE is compared with the optimal Bayesian–CCIDFE. The channels and co-channels used in this study are characterised by their impulse responses

$$\begin{aligned} H(z) &= H_5(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2} \\ H_{co\_1}(z) &= H_8(z) = \lambda(0.6 + 0.8z^{-1}) \end{aligned} \tag{4.41}$$

where λ is set to adjust the system SIR. The equaliser parameters were selected as.

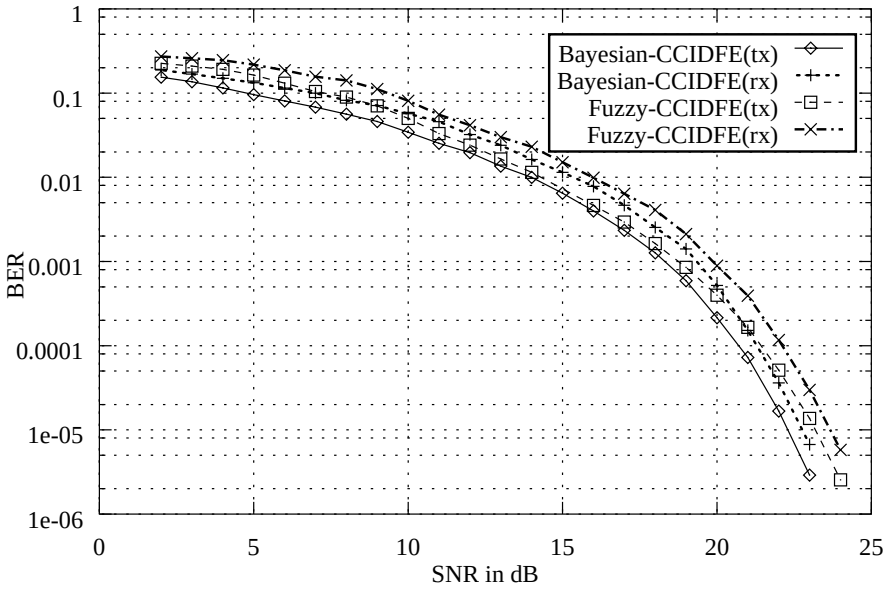
$m = 3,\ d = 2$ and $q = 2$.

The performance of the fuzzy–CCIDFE and the Bayesian–CCIDFE equaliser for SIR=10 dB and 15 dB are presented in Figure 4.15(a) and 4.15(b) respectively.

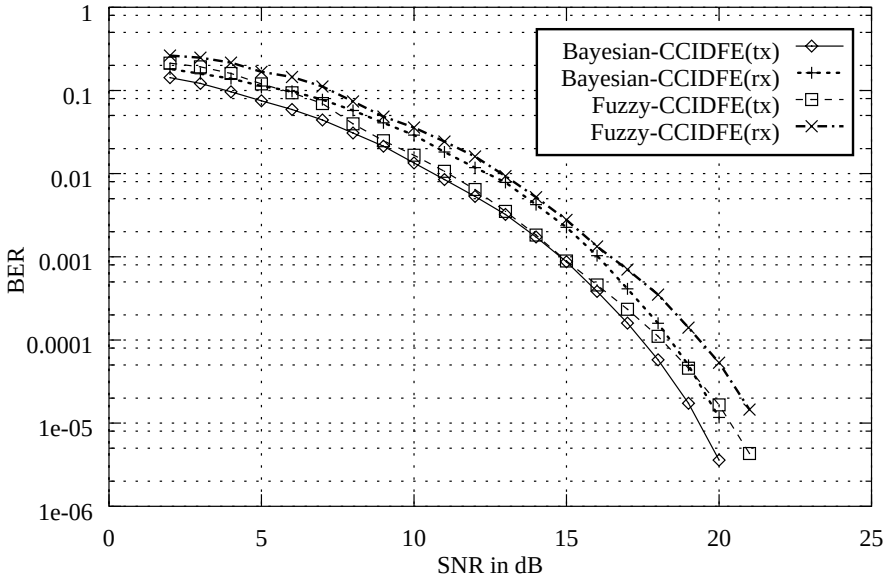
In Figures 4.15(a) and 4.15(b) the curves fuzzy–CCIDFE(rx) and fuzzy–CCIDFE(tx) represent the fuzzy–CCIDFE equaliser BER performance with detected symbol feedback and transmitted symbol feedback respectively. Similarly Bayesian–CCIDFE(rx) and Bayesian–CCIDFE(tx) represent the Bayesian–CCIDFE equaliser BER performance with detected symbol feedback and transmitted symbol feedback respectively . The fuzzy equaliser here is trained with 500 samples and the equalisers parameters are averaged over 20 experiments. The scalar channel and scalar co-channel states of the fuzzy–CCIDFE are estimated with the κ-means and enhanced κ-means clustering algorithms respectively. The membership function centre spread parameter was set to the estimated spread parameter from the channel state estimate for better performance. A study of the BER performance of the equalisers shows that fuzzy–CCIDFE provides a performance very close to Bayesian–CCIDFE and the error propagation characteristics for both the equalisers are nearly the same.

In the previous section, the condition for CCI compensation for fuzzy–CCIDFE (4.36) was presented. The scalar co-channels are estimated by unsupervised clustering and in low SNR conditions the estimation of the scalar co-channel states is not accurate. From the simulation studies the following rule has been determined to justify the necessity of using the pre-processor for equalising the CCI.

- The scalar co-channel states can be determined with an assumption of $n_{c1} = 1$ and $n_{c1} = 3$ ($n_{c1} > 3$ does not provide much performance improvement). This would provide $M_1 = 2$ and $M_1 = 8$ scalar co-channel states respectively.
- If the scalar co-channel for $M_1 = 2$ is less than half the distance between the closest scalar channel states, co-channel compensation is not necessary. Otherwise the scalar co-channel states estimated with $n_{c1} = 3$ should be used to modify the membership function generation so as to incorporate CCI compensation.



(a) SIR=10 dB



(b) SIR=15 dB

Figure 4.15: *Error propagation performance of Bayesian–CCIDFE and Fuzzy–CCIDFE equalisers with channel* $H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$*, co-channel* $H_{co\_1}(z) = \lambda(0.6 + 0.8z^{-1})$*,* $m = 3$*,* $d = 2$ *and* $q = 2$ *with estimated channel and co-channel states*

4.8 Conclusion

The problem of channel equalisation when the channel is corrupted by CCI from other users in an ISI environment has been discussed in this chapter. It is seen that fuzzy system based symbol-by-symbol spaced equaliser is capable of removing ISI in presence of CCI. This equaliser is similar to the fuzzy equaliser proposed in Chapter 3 but with a pre-processor for CCI mitigation. The pre-processor calculates the smallest absolute distance between the input scalars and scalar channel states offset with scalar co-channel states. The minimum of these distances corresponding to each of the scalar channel states is used for membership function generation. This modified membership function in conjunction with the fuzzy equaliser presented in chapter 3 is used for successful equalisation of CCI channels. This pre-processor can be removed at high SIR without performance degradation.

The fuzzy equaliser analysed here works with Gaussian membership functions, product inference and a centroid defuzzifier. Only this form of the equaliser has been analysed and simulation results have been presented in this chapter. Other forms of fuzzy equalisers with combinations of minimum inference rules and maximum defuzzification rules can provide similar performance with a reduction in computational complexity. These complexity issues have not been addressed here since they are a direct extension of the analysis presented in chapter 3.

Extensive Monte Carlo BER simulation studies demonstrate that the fuzzy equaliser presented here provides efficient equalisation even under severe CCI conditions. This equaliser is also seen to provide moderate to good performance for channels corrupted with two co-channel interferers, where RBF and linear equalisation with decision feedback fail and the computational complexity of the Bayesian–CCI prohibits its application.

Chapter 5
Conclusion

5.1 Introduction

The research carried out for this thesis primarily discusses fuzzy system based channel equalisers in digital communication receivers. The fuzzy implementation of Bayesian equaliser based on MAP criteria has been presented and the capability of fuzzy equalisers in a CCI environment has been analysed. This chapter summarises the work reported in this thesis, specifying the limitations of the study and provides some pointers to future development.

Following this introduction section 5.2 lists the achievements from the work. Section 5.3 provides the limitations and section 5.4 presents few pointers towards future work.

5.2 Achievement of the thesis

The work presented in this thesis can be seen as made up of two distinct parts. The first part presents the development of a fuzzy equaliser for ISI channels[1]. Secondly, a fuzzy equaliser is developed for equalisation of CCI channels [2]. Major points of the thesis, highlighting the contributions at each stage, are presented below.

Chapter 3 of the thesis presents a new fuzzy implementation of the Bayesian equaliser. It is seen that the Bayesian equaliser uses estimates of noise free received signal vectors called channel states to formulate the decision function. The Bayesian equaliser can be efficiently implemented using the estimates of noise free received scalars called scalar channel states and this implementation has been termed NBESS. Actual imeplementation of NBESS provide a reduction in computational complexity over the conventional Bayesian equaliser. NBESS can also be implemented using RBF with scalar centres [78]. Subsequently, the design of fuzzy equalisers using FAF is presented and it is shown that this FAF equaliser is suboptimal. The majority of

[1] This part has been presented in Chapter 3
[2] This part has been presented in Chapter 4

fuzzy equalisers designed with FAF were based on two types of FAFs, namely the RLS and LMS fuzzy filters presented in[105]. The equalisers based on the fuzzy RLS filter are computationally complex where as equalisers with the fuzzy LMS filter [105], although use less number of rules than the fuzzy RLS filter, the number of rules are at least equal to the number of channel states. This makes both of these forms of popular fuzzy equalisers [105, 116–119] more computationally complex than the NBESS. The computational complexity associated with these equalisers has limited their application to equalisers with $m = 2$. The work reported in this chapter finds the close relationship between the NBESS and the FAF equalisers, providing the parametric implementation of the NBESS using FAF. This fuzzy implemented Bayesian equaliser uses Gaussian membership functions, product inference in the form of $\mathrm{IF} \ldots \mathrm{THEN} \ldots$ rules and a COG defuzzifier. The use of fuzzy systems in implementing the Bayesian equaliser provides flexibility in the design of Bayesian equalisers. With the application of different forms of inference rules and defuzzification processes other forms of near optimal equalisers can also be designed. The use of scalar channel states in these equalisers leads to efficient techniques for subset centre selection providing major reduction in computational complexity. Some of the results presented in this chapter have been published in [78, 152, 153]. The parametric implementation of Bayesian equalisers using fuzzy systems makes the equaliser traceable and it can be implemented directly in applications where MAP or RBF equalisers are being used. Some of the major contributions from this chapter are summarised here. Fuzzy equalisers

- provide a parametric implementation of the Bayesian equaliser;
- are computationally more efficient than other forms of Bayesian equalisers, such as the RBF, from an implementation viewpoint;
- provide efficient schemes for subset centre selection resulting in major reduction in computational complexity;
- in an adaptive implementation, can be trained with small training sequences making them suitable for mobile radio communication application;
- have an ability to use different forms of inference rules, defuzzification processes providing alternate schemes to facilitate compromise between equaliser performance and computational complexity.

Chapter 4 of the thesis presents the development of a fuzzy equaliser for a CCI channel. It is seen that the Bayesian equaliser treating CCI as additive Gaussian noise fails under low to

moderate SIR. For this reason the optimal T-spaced symbol-by-symbol equaliser for this problem is derived. This Bayesian–CCI equaliser suffers from large computational complexity. A normalised form of the Bayesian–CCI equaliser which uses scalar channel and co-channel states instead of channel and co-channel states for this problem is derived and is termed as NBSS–CCI. The NBSS–CCI is seen to provide the Bayesian–CCI equaliser decision function with a reduction in computational complexity. It can be implemented with the normalised RBF network with scalar centres and with FAF in common with the equalisers for ISI channels. Despite the computational advantages of NBSS–CCI over Bayesian–CCI, the computational requirement of NBSS–CCI is seen to be very large for real time implementation in DCS. These computational complexity issues prompted the design of a new fuzzy equaliser for CCI mitigation. This fuzzy–CCI equaliser is a modified form of the fuzzy equaliser for ISI channels, providing an improvement in performance w.r.t. the Bayesian equaliser treating CCI as AWGN for comparable complexity. Performance of the fuzzy–CCI equaliser is seen to be close to that of the Bayesian–CCI equaliser under most conditions. The concept of decision feedback is introduced and the decision function of the fuzzy–CCI, the NBSS–CCI and the Bayesian–CCI are modified for DFE structure. Extensive Monte Carlo simulations for BER performance of different equalisers demonstrate the performance capabilities of the fuzzy equaliser in CCI mitigation. It is shown that the fuzzy–CCI equaliser is a fuzzy equaliser for ISI channels with a pre-processor for removal of the CCI. This pre-processor can be removed under moderate to low CCI. The conditions under which the pre-processor can be removed is defined in terms of the equaliser channel states and the scalar co-channel states. Some of the results reported in this chapter have appeared in [154, 155]. The major contributions from this chapter are listed below. The fuzzy–CCI equaliser:

- provides an efficient equalisation of channels affected by CCI, ISI and AWGN;
- has computational complexity comparable to the Bayesian equaliser treating CCI as AWGN whereas it provides a performance comparable to the Bayesian–CCI equaliser;
- uses a pre-processor for CCI mitigation; this pre-processor can be removed under moderate to low CCI; the use of this preprocessor makes the switching of the equaliser from high CCI to low CCI environment easy;
- is a more general form of the fuzzy equaliser developed for ISI channels; computational complexity reduction methods proposed for fuzzy equalisers in Chapter 3 by the use of

different form of fuzzy inference rules and defuzzification processes can also be generally applied to fuzzy–CCI equalisers.

5.3 Limitations of the work

This section presents some of the limitations of the work reported in this thesis.

This thesis presents the development of fuzzy equalisers for DCS. Fuzzy equalisers developed here implement the Bayesian equaliser with reduction in computational complexity. The computational complexity of Bayesian equalisers are related to (K^{n_c}) where K is the size of the symbol alphabet. This large complexity limits the use of these forms of nonlinear equalisers to communication systems where channel dispersion is relatively small, of the order of $n_c \approx 5$. Additionally efficient use of the available radio frequency spectrum demands efficient modulation schemes like QPSK, 4 level PAM and 8–PSK etc. to increase transmission speed with limited BW. The equaliser algorithm developed here is limited to 2-level PAM modulation. But, it can be extended to other efficient modulation schemes in line with RBF implementation of Bayesian equalisers [31, 32].

The other limitation of the work reported in the thesis lies in the stationary channel model. The impulse responses of mobile radio channels are characterised by multi-path fading. This requires the equalisers in the receivers to track the channel characteristics, which is achieved by interposing blocks of training data with actual data blocks. The adaptive equalisers use this training data to set the parameters and during actual transmission the equaliser decisions are used in a decision directed mode to track the channel fading characteristics. The performance of the proposed fuzzy equalisers is expected to be similar to RBF implementation of Bayesian equalisers under these conditions [132, 156], since the fuzzy equaliser provides parametric implementation of Bayesian equaliser.

Some of the other issues that have not been addressed in this thesis are the effects of presence of ACI, nonlinearities in the receiver amplifiers, timing recovery in the receiver, diversity combining issues related to nonlinear equalisation techniques.

5.4 Scope for further research

To conclude the thesis, the following are some pointers for further work which can lead to interesting results.

The first suggested area in which research can be undertaken follows from the limitation of the work presented in this chapter. RBF implementation of Bayesian equalisers have provided good performance for mobile communication channels [156, 157]. The long delay associated with MLSE causes severe performance degradation in fading channels. Additionally design of MLSE for CCI environment can be computationally complex [92]. Under these circumstances fuzzy equalisers could provide major performance advantages. The study of fuzzy equalisers for mobile communication systems like GSM[3] systems could provide alternative equalisation strategies.

Recently it has been observed that fractionally spaced equalisers can provide additional benefit in interference mitigation in the form of CCI and ACI [145, 146, 158]. One of the possible directions for research is investigating fractionally spaced fuzzy equalisers for interference limited communication system applications.

[3]GSM stands for global system for mobile communication

References

[1] R. W. Lucky, "Automatic Equalization of Digital Communication," *Bell System Tech. J*, vol. 44, pp. 547–588, April 1965.

[2] B. Widrow and M. E. Hoff(Jr), "Adaptive Switching Circuits," in *IRE WESCON Conv.*, vol. 4, pp. 94–104, August 1960.

[3] G. D. Forney, "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Transactions on Information Theory*, vol. IT-18, pp. 363–378, May 1972.

[4] G. D. Forney, "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–278, March 1973.

[5] F. R. Magee Jr and J. G. Proakis, "Adaptive Maximum-Likelihood Sequence Estimation for Digital Signaling in the Presence of Intersymbol Iterference," *IEEE Transactions on Information Theory*, vol. IT-19, pp. 120–124, January 1973.

[6] R. P. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE, ASSP Magazine*, vol. 4, April 1987.

[7] D. R. Hush and B. G. Horne, "Progress in Supervised Neural Networks: Whats New Since Lippmann?," *IEEE Signal Processing Magazine*, vol. 10, January 1993.

[8] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[9] C. A. Micchelli, "Interpolation of Scattered Data: Distance Matrix and Conditionally Positive Functions," *Constructive Approximation*, vol. 2, no. 1, pp. 11–22, 1986.

[10] S. Haykin, *Neural Networks - A Comprehensive Foundation*. New York: Macmillan, 1994.

[11] L.-X. Wang and J. M. Mendel, "Fuzzy Adaptive Filters, with Application to Non-linear Channel Equalization," *IEEE Transactions on Fuzzy Systems*, vol. 1, pp. 161–170, August 1993.

[12] C.-T. Lin and C.-F. Juang, "An Adaptive Neural Fuzzy Filter and its Applications," *IEEE Transactions on System Man and Cybernatics - Part-B: Cybernatics*, vol. 27, pp. 635–656, August 1997.

[13] J. C. Bezdek and P. F. Castelaz, "Prototype classification and feature selection with fuzzy sets," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-7, pp. 87–92, February 1977.

[14] K. Nozaki, H. Ishibuchi, and H. Tanaka, "Adaptive Fuzzy Rule-Based Classification Systems," *IEEE Transactions on Fuzzy Systems*, vol. 4, pp. 238–250, August 1996.

[15] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Transactions on Neural Networks*, vol. 3, pp. 683–697, September 1992.

[16] H. Nyquist, "Certain Topics in Telegraph Transmission Theory," *Transactions of AIEE*, vol. 47, pp. 617–644, April 1928.

[17] D. A. George, R. R. Bowen, and J. R. Storey, "An Adaptive Decision Feedback Equaliser," *IEEE Transactions on Communication Technology*, vol. COM-19, pp. 281–293, June 1971.

[18] D. D. Falconer, "Joint Adaptive Equalization and Carrier Recovery in Two-dimensional Digital Communication Systems," *Bell System Technical Journal*, vol. 55, pp. 317–334, March 1976.

[19] D. Godard, "Channel Equalization Using Kalman Filter for Fast Data Transmission," *IBM Journal Res. Development*, vol. 18, pp. 267–273, May 1974.

[20] J. Makhoul, "A Class of All-Zero Lattice Digital Filters," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-26, pp. 304–314, August 1978.

[21] J. R. Treichler, I. Fijakow, and C. R. Johnson, "Fractionally Spaced Equalizers - How Long Should They Realy be?," *IEEE Signal Processing Magazine*, pp. 65–81, May 1996.

[22] S. U. H. Qureshi, "Adaptive Equalization," *Proceedings of the IEEE*, vol. 73, pp. 1349–1387, September 1985.

[23] S. Siu, G. J. Gibson, and C. F. N. Cowan, "Decision Feedback Equalization using Neural Network Structures and Performance Comparison with Standard Architecture," *Proceedings-I of the IEE*, vol. 137, pp. 221–225, August 1990.

[24] 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.

[25] M. J. D. Powell, "Radial basis function for multivariable interpolation: A review," in *Algorithms for Approximation of Functions and Data* (J. C. Mason and M. G. Cox, eds.), pp. 143–167, Oxford University Press, 1987.

[26] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal Least Squares Learning Algorithms for Radial Basis Function Networks," *IEEE Transactions on Neural Networks*, vol. 2, pp. 302–309, March 1991.

[27] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Reconstruction of Binary Signals using an Adaptive Radial Basis Function Equalizer," *Signal Processing (Eurasip)*, vol. 22, pp. 77–93, January 1991.

[28] B. Mulgrew, "Nonlinear Signal Processing for Adaptive Equalisation and Multi-user Detection," in *Proceedings of the European Signal Processing Conference*, EUSIPCO, (Island of Rhodes, GREECE), pp. 537–544, 8-11 September 1998.

[29] J. Cid-Sueiro and A. R. Figueiras-Vidal, "Channel Equalization with Neural Networks," in *Digital Signal Processing in Telecommuniocations - European Project COST#229 Technical Contributions* (A. R. Figueiras-Vidal, ed.), pp. 257–312, London, U.K.: Springer-Verlag, 1996.

[30] B. Mulgrew, "Applying Radial Basis Functions," *IEEE Signal Processing Magazine*, vol. 13, pp. 50–65, March 1996.

[31] S. Chen, S. McLaughlin, and B. Mulgrew, "Complex-valued Radial Basis Function Network, Part I: Network Architecture and Learning Algorithms," *Signal Processing (Eurasip)*, vol. 35, pp. 19–31, January 1994.

[32] S. Chen, S. McLaughlin, and B. Mulgrew, "Complex-valued Radial Basis Function Network, Part II: Application to Digital Communication Channel Equalization," *Signal Processing (Eurasip)*, vol. 36, pp. 175–188, March 1994.

[33] X. Liu and T. Adali, "Recurrent Canonical Piecewise Linear Network and Its Application to Adaptive Equalization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, (Washington, DC, USA), pp. 1969–1973, IEEE, 3-6 June 1996.

[34] R. Parishi, E. D. D. Claudio, G. Orlandi, and B. D. Rao, "Fast Adaptive Digital Equalization by Reccurent Neural Networks," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2731–2739, November 1997.

[35] J. Cid-Sueiro, A. Artes-Rodriguez, and A. R. Figueiras-Vidal, "Recurrent Radial Basis Function Network for Optimal Symbol-by-Symbol Equalisation," *Signal Processing (Eurasip)*, vol. 40, pp. 53–63, October 1994.

[36] S. Theodoridis, C. F. N. Cowan, C. P. Callender, and C. M. S. See, "Schemes for Equalisation of Communication Channels with Non-linear Impairments," *IEE Proceedings - Communication*, vol. 142, pp. 165–171, June 1995.

[37] T. Adali, X. Liu, and M. K. Sönmez, "Conditional Distribution Learning with Neural Network and Its Application to Channel Equalization," *IEEE Transactions on Signal Processing*, vol. 45, pp. 1051–1064, April 1997.

[38] C. Z. W. H. Sweatman, B. Mulgrew, and G. J. Gibson, "Two Algorithms for Neural Network Design and Training with Application to Channel Equalization," *IEEE Transactions on Neural Networks*, vol. 9, pp. 533–543, May 1998.

[39] D. G. Oh, J. Y. Choi, and C. W. Lee, "Adaptive Nonlinear Equalizer with Reduced Computational Complexity," *Signal Processing (Eurasip)*, vol. 47, pp. 307–317, December 1995.

[40] D. G. M. Cruickshank, "Radial Basis Function Receivers for DS-CDMA," *Electronics Letters*, vol. 32, pp. 188–190, 01 February 1996.

[41] R. Tanner, D. G. M. Cruickshank, C. Z. W. H. Sweatman, and B. Mulgrew, "Receivers for Nonlinearly Separable Scenarios in DS-CDMA," *Electronics Letters*, vol. 33, pp. 2103–2105, December 1997.

[42] U. Mitra and H. V. Poor, "Neural Network Techniques for Adaptive Multiuser Demodulation," *IEEE Transactions on Communications*, vol. 12, pp. 1460–1470, December 1994.

[43] J. G. Proakis, *Digital Communications*. New York: Mc-Graw Hill Inc., 1995.

[44] J. Ido, M. Okada, and S. Komaki, "New Neural Network Based Nonlinear and Multipath Distortion Equalizer for FTTA Systems," *IEICE Transactions : Communication*, vol. E80-B, pp. 1138–1144, August 1997.

[45] R. Steel(Ed), *Mobile Radio Communication*. Pentec Press, London, 1992.

[46] B. R. Peterson and D. D. Falconer, "Minimum Mean Square Equalization in Cyclostationary and Stationary Interference-Analysis and Subscriber Line Calculations," *IEEE Journal on Selected Areas in Communication*, vol. 9, pp. 931–940, August 1991.

[47] J. C. Campbell, A. J. Gibbs, and B. M. Smith, "The Cyclostationary Nature of Crosstalk Interference from Digital Signals in Multipair Cables - Part I: Fundamentals, Part II: Applications and Further results," *IEEE Transactions on Communications*, vol. COM31, pp. 629–649, May 1983.

[48] J. H. Winter, "Optimum Combining in Digital Mobile Radio with Cochannel Interference," *IEEE Journal on Selected Areas in Communication*, vol. SAC-2, pp. 528–539, July 1984.

[49] J. G. Proakis, "Adaptive Equalization for TDMA Digital Mobile Radio," *IEEE Transactions on Vehicular Technology*, vol. 40, pp. 333–341, May 1991.

[50] K. Feher, "MODEMS for Emerging Digital Cellular-Mobile Radio system," *IEEE Transactions on Vehicular Technology*, vol. 40, pp. 355–365, May 1991.

[51] B. R. Peterson and D. D. Falconer, "Supression of Adjacent Channel Interference in Digital Radio by Equalization," in *Proceeding of IEEE International Conference on Communications*, (Chicago, IL, USA), pp. 0657–0661, IEEE, 14-18 June 1992.

[52] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.

[53] 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.

[54] J. F. Hayes, T. M. Cover, and J. B. Riera, "Optimal Sequence Detection and Optimal Symbol-by-Symbol Detection: Similar Algorithms," *IEEE Transactions on Communications*, vol. COM-30, pp. 152–157, January 1982.

[55] S. Haykin, *Adaptive Filter Theory*. Englewood Cliff, NJ, USA: Prentice Hall, 1991.

[56] B. Mulgrew, P. M. Grant, and J. S. Thompson, *Digital Signal Processing: Concepts and Applications*. Houndmills, Basingstoke, U.K.: Macmillan, 1 ed., 1999.

[57] S. Chen, B. Mulgrew, and P. M. Grant, "A Clustering Technique for Digital Communication Channel Equalization Using Radial Basis Function Networks," *IEEE Transactions on Neural Networks*, vol. 4, pp. 570–579, July 1993.

[58] K. Abend and B. D. Fritchman, "Statistical Detection for Communication Channel with Intersymbol Interference," *Proceedings of the IEEE*, vol. 58, pp. 779–785, May 1970.

[59] A. P. Clark, *Equalizers for Digital Modems*. Pentech Press, London, 1985.

[60] G. Ungerboeck, "Theory on the Speed on Convergence in Adaptive Equalizers for Digital Communication," *IBM Journal of Research and Development*, vol. 16, pp. 546–555, 1972.

[61] J. Salz, "Optimum Mean-Square Decision Feedback Equalization," *Bell System Tech. J*, vol. 52, pp. 1341–1373, October 1973.

[62] M. T. Özeden, A. H. Kayran, and E. Panayirci, "Adaptive Volterra Channel Equalisation with Lattice Orthogonalisation," *IEE Proceedings - Communication*, vol. 145, pp. 109–115, April 1998.

[63] W. S. Gan, "Adaptive Channel Equalisation Using the Functional-link (FL) Technique," *Journal of Electrical and Electronics Engineering, Australia*, vol. 16, pp. 65–72, March 1996.

[64] K. Y. Lee, S.-Y. Lee, and S. McLaughlin, "A Neural Network Equaliser with the Fuzzy Decision Learning Rule," in *Proceedings of 7th IEEE Workshop on Neural Networks for Signal Processing: NNSP-97*, (Amelia Islands, FL, USA), pp. 551–559, IEEE, 24-26 September 1997.

[65] N. Dyn, "Interpolation of Scattered data by Radial Functions," in *Topics in Multivariate Interpolation* (C. K. Chiu, L. L. Schumaker, and F. Ultreras, eds.), pp. 47–61, New York: Academic Press, 1987.

[66] E. S. Chng and B. Mulgrew, "Gradient Radial Basis Function Networks for Nonlinear and Nonstationary Time Series Prediction," *IEEE Transactions on Neural Networks*, vol. 7, pp. 190–194, January 1996.

[67] S. Chen, S. A. Billings, C. F. N. Cowan, and P. M. Grant, "Non-linear system identification using radial basis functions," *International Journal of Systems Science*, vol. 21, pp. 2513–2539, December 1990.

[68] S. Chen, S. A. Billings, and P. M. Grant, "Recursive Hybrid Algorithm for Non-linear System Identification using Radial Basis Function Networks," *International Journal of Control*, vol. 55, pp. 1051–1070, May 1992.

[69] I. Cha and S. A. Kassam, "Interference Cancellation using Radial Basis Function Network," *Signal Processing (Eurasip)*, vol. 47, pp. 247–268, December 1995.

[70] Q. Zhao and Z. Bao, "Radar Target Recognition Using a Radial Basis Function Neural Network," *Neural Networks, Elsevier Science*, vol. 9, no. 4, pp. 709–720, 1996.

[71] J. A. Leonard and M. A. Kramer, "Radial Basis Function Networks for Classifying Process Faults," *IEEE Communications Systems Magazine*, pp. 31–38, April 1991.

[72] L. Tarassenko and S. Roberts, "Supervised and Unsupervised Learning in Radial Basis Function Classifiers," *IEE Proceedings - Vision Image Signal Processing*, vol. 141, pp. 210–216, August 1994.

[73] M. A. Tugay and Y. Tanik, "Properties of the Momentum LMS Algorithm," *Signal Processing (Eurasip)*, vol. 18, no. 2, pp. 117–127, 1989.

[74] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal Least Squares Algorithms for Training Multioutput Radial Basis Function Networks," *Proceedings-F of the IEE : Radar and Signal Processing*, vol. 139, pp. 378–384, December 1992.

[75] J. Moody and C. J. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989.

[76] I. Cha and S. A. Kassam, "Channel Equalization Using Complex Radial Basis Function Networks," *IEEE Journal on Selected Areas in Communication*, vol. 13, pp. 122–131, January 1995.

[77] E.-S. Chng, H. Yang, and W. Skarbek, "Reduced Complexity Implementation of Bayesian Equalizer Using Local RBF Network for Channel Equalization Problem," *Electronics Letters*, vol. 32, pp. 17–19, January 1996.

[78] S. K. Patra and B. Mulgrew, "Computational Aspects for Adaptive Radial Basis Function Equalizer Design," in *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 1, (Hong Kong), pp. 521–524, IEEE, 9-12 June 1997.

[79] G. Cybenko, "Approximation by Superposition of Sigmoidal Function," *Math. Contr. Signals Syst.*, vol. 2, no. 4, 1989.

[80] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representation by Error Propagation," in *Parallel and distributed processing: Exploration in microstructure of cognition* (D. E. Rumelhart and J. L. McClelland, eds.), pp. 318–362, MIT Press, 1986.

[81] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Adaptive Equalization of Finite Non-Linear Channels Using Multilayer Perceptron," *Signal Processing (Eurasip)*, vol. 20, pp. 107–119, June 1990.

[82] M. Meyer and G. Pfeiffer, "Multilayer Perceptron Based Decision Feedback Equalisers for Channels with Intersymbol Interferance," *Proceedings-I of the IEE*, vol. 140, pp. 420–424, December 1993.

[83] P.-R. Chang and B.-C. Wang, "Adaptive Decision Feedback Equalization for Digital Satellite Channels Using Multilayer Neural Networks," *IEEE Transactions on Communications*, vol. 13, pp. 316–324, February 1995.

[84] W. R. Kirkland and D. P. Taylor, "On the Application of Feedforward Neural Networks to Channel Equalization," in *Proceedings of Intl. Joint Conf. on Neural Networks*, IJCNN, (New York), 1992.

[85] P. R. Chang, B. F. Yeh, and C. C. Chang, "Adaptive Packet Equalization for Indoor Radio Channels using Multilayer Neural Networks," *IEEE Transactions on Vehicular Technology*, vol. 43, pp. 773–780, August 1994.

[86] K. A. Al-Mashouq and I. S. Reed, "The use of Neural Nets to Combine Equalization with Decoding for Severe Intersymbol Interference Channels," *IEEE Transactions on Neural Networks*, vol. 5, pp. 982–988, November 1994.

[87] S. K. Nair and J. Moon, "Data Storage Channel Equalization Using Neural Networks," *IEEE Transactions on Neural Networks*, vol. 7, pp. 1037–1048, September 1997.

[88] Z. Xiang, G. Bi, and T. Le-Ngoc, "Polynomial Perceptron and their Applications to Fading Channel Equalization and Co-channel Interference Suppression," *IEEE Transactions on Signal Processing*, vol. 42, pp. 2470–2480, September 1994.

[89] Y. Iiguni, H. Sakai, and H. Tokumaru, "A Real-time Learning Algorithm for Multilayered Neural Networks Based on Extended Kalman Filter," *IEEE Transactions on Signal Processing*, vol. 40, pp. 959–966, April 1992.

[90] R. Parisi, E. D. Di Claudio, G. Orlandi, and B. D. Rao, "A Generalized Learning Paradigm Exploiting the Structure of Neural Nets," *IEEE Transactions on Neural Networks*, vol. 7, pp. 1450–1460, November 1996.

[91] J. J. Xue and X. H. Yu, "A mean Field Annealing Partially-Connected Neural Equalizer for Pan-European GSM System," in *Proceeding of IEEE International Conference on Communications*, vol. 2, (Dallas, TX, USA), pp. 701–505, IEEE, 23-27 June 1996.

[92] S. Chen, S. McLaughlin, B. Mulgrew, and P. M. Grant, "Bayesian Decision Feedback Equaliser for Overcoming Co-channel Interference," *IEE Proceedings - Communication*, vol. 143, pp. 219–225, August 1996.

[93] L. O. Chua and L. Yang, "Cellular Neural Networks: Theory," *IEEE Transactions on Circuits and Systems*, vol. 35, pp. 1257–1272, October 1988.

[94] B. W. Lee and B. J. Sheu, "Parallel Hardware Annealing for Optimal Solutions on Electronic Neural Networks," *IEEE Transactions on Neural Networks*, vol. 4, pp. 588–598, July 1993.

[95] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks*. Addition Wesley, Reading, MA, 1989.

[96] S. Chen, G. J. Gibson, and C. F. N. Cowan, "Adaptive Channel Equalization using a Polynomial-Perceptron Structure," *Proceedings-I of the IEE*, vol. 137, pp. 257–264, October 1990.

[97] C.-H. Chang, S. Siu, and C.-H. Wei, "A Polynomial-Perceptron based Decision Feedback Equalizer with a Robust Learning Algorithm," *Signal Processing (Eurasip)*, vol. 47, pp. 145–158, November 1995.

[98] W. S. Gan, J. J. Soreghan, and T. S. Durrani, "A new Functional-link based Equaliser," *Electronics Letters*, vol. 28, pp. 1643–1645, 13 August 1992.

[99] J. C. Patra and R. N. Pal, "A Functional Link Artificial Neural Network for Adaptive Channel Equalization," *Signal Processing (Eurasip)*, vol. 43, pp. 181–195, May 1995.

[100] A. Hussain, J. J. Soraghan, and T. S. Durrani, "A New Adaptive Functional-link Neural-Network- Based DFE for Overcoming Co-Channel Interference," *IEEE Transactions on Communications*, vol. 45, pp. 1358–1362, November 1997.

[101] Z.-J. Xiang, G. G. Bi, and C. B. Schlegel, "A New Lattice Polynomial Perceptron and its Application to Fading Channel Equalization and Adjacent-channel Interference Supression," in *Proceeding of IEEE International Conference on Communications*, vol. 1, (Seatle, WA, USA), pp. 1478–1482, IEEE, 18-22 June 1995.

[102] K. Hasiog̃lu and M. Abdelhafez, "Reconstruction of PAM signals using multi layer perceptron with multilevel sigmoidal function," in *Proceedings of 7th European Signal Processing Conference*, EUSIPCO, (Edinburgh, U.K.), pp. 1811–1814, 13-16 September 1994.

[103] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*. New York, USA: John Wiley and Sons, Inc., 1994.

[104] W. Pedrycz, *Fuzzy Control and Fuzzy Systems*. Taunton Studies Press, 1993.

[105] L.-X. Wang, *Adaptive Fuzzy Systems and Control : Design and Stability Analysis*. Englewood Cliffs, N.J., USA: Prentice Hall, 1994.

[106] R. J. Marks, ed., *Fuzzy Logic Technology and Applications*. IEEE technology Update, USA: IEEE, 1994.

[107] J. M. Mendel, "Fuzzy Logic System for Engineering: A Tutorial," *Proceedings of the IEEE*, vol. 83, pp. 345–377, March 1995.

[108] P. K. Simpson, "Fuzzy Min-Max Neural Networks - Part 1: Classification," *IEEE Transactions on Neural Networks*, vol. 3, pp. 776–786, September 1992.

[109] P. K. Simpson, "Fuzzy Min-Max Neural Networks - Part 2: Clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, pp. 32–45, February 1993.

[110] P. J. King and E. H. Mamdani, "The Application of Fuzzy Control System to Industrial Processes," *Automatica*, vol. 13, pp. 235–242, 1977.

[111] K. G. Shin and X. Cui, "Design of a Knowledge-Based Controller for Intelligent Control Systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, pp. 365–375, March/ April 1991.

[112] K. H. Kienitz, "Controller Design Using Fuzzy Logic - A Case Study," *Automatica*, vol. 29, pp. 549–554, March 1993.

[113] L. Wang and J. M. Mendel, "Generating Fuzzy Rules by Learning from Examples," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, pp. 1414–1427, November/ December 1992.

[114] L.-X. Wang and J. M. Mendel, "Fuzzy Basis Function, Universal Approximation and Orthogonal Least Square Learning," *IEEE Transactions on Neural Networks*, vol. 3, pp. 807–814, September 1992.

[115] K. Y. Lee, "Fuzzy Adaptive Decision Feedback Equaliser," *Electronics Letters*, vol. 30, pp. 749–751, 12 May 1994.

[116] K. Y. Lee, "Complex RLS Fuzzy Adaptive Filter and its Application to Channel Equalisation," *Electronics Letters*, vol. 30, pp. 1572–1574, 15 September 1994.

[117] S. Y. Lee, J. B. Kim, C. J. Lee, K. Y. Lee, and C. W. Lee, "Complex RLS Fuzzy Adaptive Decision Feedback Equalizer," *IEICE Transactions : Communication*, vol. E79-B, pp. 1911–1913, December 1996.

[118] K. Y. Lee, "Complex Fuzzy Adaptive Filter with LMS Algorithm," *IEEE Transactions on Signal Processing*, vol. 44, pp. 424–426, February 1996.

[119] S. Y. Lee, J. B. Kim, K. Y. Lee, and C. W. Lee, "Complex RLS Adaptive Equalizer with Decision Feedback," in *Conference notes IEEE GLOBCOM*, (London, U.K.), pp. 191–195, 18-22 November 1996.

[120] P. Sarwal and M. D. Srinath, "A Fuzzy Logic System for Channel Equalization," *IEEE Transactions on Fuzzy Systems*, vol. 3, pp. 246–249, May 1995.

[121] J. Nie and T. H. Lee, "A Rule-Based Channel Equalizer with Learning Capability," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 1, (Perth, Australia), pp. 606–611, IEEE, November 1995.

[122] W. S. Gan, "Design of a Fuzzy Step Size LMS Algorithm," *IEE Proceedings - Vision Image Signal Processing*, vol. 144, pp. 261–266, October 1997.

[123] W. S. Gan, "Fuzzy Step Size Adjustment for the LMS Algorithm," *Signal Processing (Eurasip)*, vol. 49, pp. 145–149, March 1996.

[124] X.-Y. Ho and X.-H. Yu, "Fuzzy Stocastic Gradient Decision Feedback Equalizer for VSB Terrestial HDTV Broadcasting," *IEEE Transactions on Broadcasting*, vol. 43, pp. 36–46, March 1997.

[125] C. T. Lin and C. F. Juang, "An Adaptive Neural Fuzzy Filter and its Application," in *Proceeding of IEEE International Conference on Fuzzy Systems*, vol. 1, (New Orleans, LA, USA), pp. 564–569, IEEE, 8-11 September 1996.

[126] R. P. Lippmann, "A Critical Overview of Neural Network Pattern Classifiers," in *Proceedings of 1991 IEEE Works Shop on Neural Network for Signal Processing*, (Priceton, NJ, USA), pp. 266–275, IEEE, 30 September- 2 October 1991.

[127] R. P. Lippmann, "Pattern Classification Using Neural Networks," *IEEE Communications Magazine*, vol. 27, pp. 47–64, November 1989.

[128] J. S. R. Jang and C. T. Sun, "Functional Equivalance Between Radial Basis Function Networks and Fuzzy Inference Systems," *IEEE Transactions on Neural Networks*, vol. 4, pp. 156–159, January 1993.

[129] H. M. Kim and J. M. Mendel, "Fuzzy Basis Functions: Comparisons with Other Basis Functions," *IEEE Transactions on Fuzzy Systems*, vol. 3, pp. 158–167, May 1995.

[130] R. Shorten and R. M. Smith, "Side Effects of Normalising Radial Basis Function Networks," *International Journal on Neural Systems*, vol. 7, pp. 167–179, May 1996.

[131] T. Poggio and F. Girosi, "Network for Approximation and Learning," *Proceedings of the IEEE*, vol. 78, pp. 1481–1497, September 1990.

[132] 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. 1937–1946, May 1995.

[133] J. Lee, C. D. Beach, and N. Tepedelenlioglu, "Channel Equalization Using Radial Basis Function Network," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, (Washington, DC, USA), pp. 1924–1928, IEEE, 3-6 June 1996.

[134] B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communication Systems Part I:Characterizatioin," *IEEE Communications Magazine*, pp. 90–100, July 1997.

[135] B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communication Systems Part II: Mitigation," *IEEE Communications Magazine*, pp. 148–155, July 1997.

[136] M. L. Honig, P. Crespo, and K. Steiglitz, "Suppression of Near- and Far-End Crosstalk by Linear Pre- and Post-Filtering," *IEEE Journal on Selected Areas in Communication*, vol. 10, pp. 614–629, April 1992.

[137] W. V. Etten, "An Optimum Linear Receiver for Multiple Channel Digital Transmission System," *IEEE Transactions on Communications*, pp. 828–834, August 1975.

[138] W. V. Etten, "Maximum Likelihood Receiver for Multiple Channel Transmission Systems," *IEEE Transactions on Communications*, pp. 276–283, February 1976.

[139] J. Salz, "Digital Transmission Over Cross-Coupled Linear Channels," *AT & T Tech Journal*, vol. 64, pp. 1147–1159, July-August 1985.

[140] S. W. Wales, "Technique for Cochannel Interference Supression in TDMA Mobile Radio Systems," *IEE Proceedings - Communication*, vol. 142, pp. 106–114, April 1995.

[141] K. Giridhar, S. Chari, J. J. Shynk, and R. P. Gooch, "Joint Demodulation of Co-channel Signals Using MLSE and MAPSD Algorithms," in *Proceedings of International Conference on Accoustics, Speech and Signal Processing*, vol. IV-Statistical Analysis and Array Processing, (Minneapolis, USA), pp. 160–163, IEEE, 27-30 April 1993.

[142] K. Giridhar, J. J. Shynk, A. Mathur, S. Chari, and R. P. Gooch, "Nonlinear Techniques for the Joint Estimation of Co-channel Signals," *IEEE Transactions on Communications*, vol. 45, pp. 473–484, April 1997.

[143] J. H. Reed and T. C. Hsia, "The Performance of Time-Dependent Adaptive Filters for Interference Rejection," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, pp. 1375–1385, August 1990.

[144] M. Abdulrahman and D. D. Falconer, "Cyclostationary Crosstalk Supression by Decision Feedback Equalization on Digital Subscriber Loops," *IEEE Journal on Selected Areas in Communication*, vol. 10, pp. 640–649, April 1992.

[145] B. R. Petersen, *Equalization in Cyclostationary Interference*. PhD thesis, Dept of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada, January 1992.

[146] N. W. K. Lo, *Adaptive Equalization for a Multipath Fading Channel in Presence of Interference*. PhD thesis, Dept of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada, April 1994.

[147] N. W. K. Lo, D. D. Falconer, and A. U. H. Sheikh, "Adaptive Equalization for Co-Channel Interference in a Multipath Fading Environment," *IEEE Transactions on Communications*, vol. 43, pp. 1441–1453, February/March/April 1995.

[148] S. Chen and B. Mulgrew, "Overcoming Co-channel Interference Using an Adaptive Radial Basis Function Equalizer," *Signal Processing (Eurasip)*, vol. 28, pp. 91–107, July 1992.

[149] Z. Xian and G. Bi, "Complex Neuron Model with its Application to M-QAM dataCommunication in Presence of Co-Channel Intereference," in *Proceedings of International Conference on Accoustics, Speech and Signal Processing*, (San Francisco, USA), pp. II–(306)–II–(308), IEEE, 23-26 March 1992.

[150] K. Georgoulakis and S. Theodoridis, "Efficient Clustering Technique for Channel Equalization in Hostile Environment," *Signal Processing (Eurasip)*, vol. 58, pp. 153–164, April 1997.

[151] C. Chinrungrueng and C. H. Sequin, "Optimal Adaptive κ-Means Algorithm with Dynamic Adjustment of Learning Rate," *IEEE Transactions on Neural Networks*, vol. 6, pp. 157–169, January 1995.

[152] S. K. Patra and B. Mulgrew, "Efficient Architecture for Bayesian Equalization using Fuzzy Filters," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, pp. 812–820, July 1998.

[153] S. K. Patra, M. Mulgrew, and P. M. Grant, "Subset Centre Selection with Fuzzy Implemented Radial Basis Function Equalisers," in *First International Symposium on Communication System and Digital Signal Processing*, (Sheffield Hallan University, Sheffield, U.K.), pp. 21–24, 6-8 April 1998.

[154] S. K. Patra and B. Mulgrew, "Co-Channel Interference Supression using a Fuzzy Filter," in *Proceedings of the European Signal Processing Conference*, EUSIPCO, (Island of Rhodes, GREECE), pp. 1609–1612, 8-11 September 1998.

[155] S. K. Patra and B. Mulgrew, "Fuzzy Implementation of Bayesian Equalizer in the Presence of Intersymbol and Co-Channel Interference," *Accepted for Publication - IEE Proceedings - Communications*, 1998.

[156] J. L. Valenzuela and F. Casadevall, "Performance of Adaptive Bayesian Equalizers in Outdoor Environments," in *Proceeding of Vehicular Technology Conference*, vol. 3, (Phoenix, Arizona, USA), pp. 2143–2147, IEEE, 4-7 May 1997.

[157] Q. Gan, R. Subramaniam, N. Sundarrajan, and P. Saratchandran, "Design for Centres of RBF Neural Networks for Fast Time-varying Channel Equalisation," *Electronics Letters*, vol. 32, pp. 2333–2334, 5 December 1996.

[158] N. W. K. Lo, D. D. Falconer, and A. U. H. Sheikh, "Adaptive Equalization for a Multipath Fading Environment with Interference and Noise," in *Proceeding of Vehicular Technology Conference*, (Stochholm, Sweden), pp. 252–266, IEEE, 8-10 June 1994.

Appendix A
Clustering Algorithm

This appendix presents the κ-means supervised clustering algorithm for the estimation of scalar channel states and the enhanced κ-means[151] unsupervised clustering algorithm for the estimation of the scalar co-channel states required for training the fuzzy and fuzzy–CCI equalisers.

A.1 Estimation of scalar channel states

During the training period the transmitted symbol sequence is known to the receiver. At the time k, it can be inferred from $r(k)$ which member of desired scalar channel state occurred.

The noisy observation of Gaussian clusters of $r(k)$ are centred at desired scalar states C_i. With this, the supervised κ-means procedure can be used to effectively filter out noise. The computational algorithm for this procedure is outlined below,

if $(\mathbf{s}(k) == \mathbf{s}_i)\{$
$\quad C_i(k) = counter_i * C_i(k-1) + r(k);$
$\quad counter_i = counter_i + 1;$
$\quad C_i(k) = C_i(k)/counter_i;$
$\quad C_{M-i+1} = -C_i;$
$\}$

The scalar channel states in a non-stationary environment can be estimated using the following algorithm.

if $(\mathbf{s}(k) == \mathbf{s}_i)\{$
$\quad$ if $(C_i(k) == 0.0);\{$
$\quad\quad C_i(k) == r(k);$
$\quad\quad C_{M-i+1}(k) = -C_i(k);$
$\quad\}$
$\quad$ else$\{$

$$C_i(k) = C_i(k-1) + \mu * r(k);$$
$$C_{M-i+1} = -C_i;$$
$\}\}$

where μ is the learning rate for the states. After the scalar channel states are estimated the combination of scalar channel states with the training signal can be used to construct the channel states information.

A.2 Estimation of scalar co-channel states

The scalar co-channel states can be estimated by the enhanced κ-means clustering algorithm [151]. This clustering assumes the variance of all clusters is equal, which is the case in equalisation applications. The scalar co-channel states can be estimated in the following steps,

1. Compute the channel residual

$$r_{res}(k) = r(k) - C_i(k) \tag{A.1}$$

2. Compute the cluster variance weighted squares distance between the residual $r_{res}(k)$ and the scalar co-channel states $C_{co,\alpha}(k-1),\ 1 \leq \alpha \leq M_{co}$.

$$\begin{aligned}\bar{\zeta}_\alpha(k) &= \sigma_{co,\alpha}(k-1)\zeta_\alpha(k) \\ &= \sigma_{co,\alpha}(k-1)\left(r_{res}(k) - C_{co,\alpha}(k-1)\right)^2,\ 1 \leq \alpha \leq M_{co}\end{aligned} \tag{A.2}$$

   Here $\sigma_{co,\alpha}(k-1)$ is the current variance of αth cluster and $\zeta_\alpha(k)$ is the squared distance between $r_{res}(k)$ and $C_{co,\alpha}(k-1)$.

3. Evaluate the minimum weighted distance

$$\bar{\zeta}_{\alpha^*}(k) = \min\{\bar{\zeta}_\alpha(k),\ 1 \leq \alpha \leq M_{co}\} \tag{A.3}$$

4. Update the α^*th and $(M_{co} - \alpha^* + 1)$th co-channel states.

$$C_{co,\alpha^*}(k) = C_{co,\alpha^*}(k-1) + \mu\left(r_{res}(k) - C_{co,\alpha}(k-1)\right) \tag{A.4}$$
$$C_{co,M_{co}-\alpha^*+1} = -C_{co,\alpha^*}(k) \tag{A.5}$$

where μ is the adaptive gain.

5. The cluster variance are updated according to the rule

$$\sigma_{co,\alpha}(k) = \nu\sigma_{co,\alpha}(k-1),\ 1 \leq \alpha \leq M_{co} \text{ and } l \neq \alpha^*, M_{co} - \alpha^* + 1 \tag{A.6}$$

$$\sigma_{co,\alpha^*}(k) = \sigma_{co,M_{co}-\alpha^*+1}(k) = \nu\sigma_{co,\alpha^*}(k-1) + (1.0-\nu)\zeta_{\alpha^*}(k) \tag{A.7}$$

where ν is positive constant slightly less than 1.0.

The initial spreads $\sigma_{co,\alpha},\ 1 \leq \alpha \leq M_{co}$ can be set to small values. The symmetry structure of the states is exploited by setting $C_{co,M_{co}-\alpha+1} = -C_{co,\alpha}(k)$ and $\sigma_{co,\alpha}(k) = \sigma_{c,M_{co}-\alpha+1}(k)$, which helps to speed up convergence and all the $\sigma_{co,\alpha}$ can be set to uniform values for the equalisation problem.

Appendix B
Channel Impulse Responses used in the Thesis

Following channels have been used for evaluation of fuzzy equalisers developed in this thesis.

No.	Impulse response	Zero location	Channel type
$H_1(z)$	$0.5 + 1.0z^{-1}$	-2.0	nonminimum phase
$H_2(z)$	$1.0 + 0.2z^{-1}$	-0.2	minimum phase
$H_3(z)$	$0.2682 + 0.9296z^{-1} + 0.2682z^{-2}$	$-3.1484, -0.3176$	mixed phase
$H_4(z)$	$0.5 + 0.81z^{-1} + 0.31z^{-2}$	$-0.62, -1.0$	mixed phase
$H_5(z)$	$0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$	$-2.0, -0.5$	mixed phase
$H_6(z)$	$0.407 - 0.815z^{-1} - 0.407z^{-2}$	$-0.4139, 2.4163$	mixed phase
$H_7(z)$	$0.7255 + 0.584z^{-1} + 0.3627z^{-2} + 0.0724z^{-3}$	$-0.2608 \pm \mathrm{j}\,0.5333, -0.283$	minimum phase
$H_8(z)$	$0.6 + 0.8z^{-1}$	-1.33333	nonminimum phase
$H_9(z)$	$0.2294 + 0.4588z^{-1} + 0.688z^{-2} + 0.4588z^{-3} + 0.2294z^{-4}$	$-0.4852 \pm \mathrm{j}\,0.8744, -0.5148 \pm \mathrm{j}0.8573$	mixed phase

$\mathrm{j} = \sqrt{-1}$

Appendix C
Publications

The following publications have either been published or accepted for publication in journals or conferences. The publications marked with a † have been included in this appendix.

Published in Journals

- S. K. Patra and B. Mulgrew, "Fuzzy Implementation of Bayesian Equalizer in Presence of intersymbol and Co-channel Interference," *To be published, IEE Proceedings - Communication*, 1998.
- † S. K. Patra and B. Mulgrew, "Efficient Architecture for Bayesian Equalization using Fuzzy Filters," *IEEE Transaction Circuits and Systems-II:Analog and Digital Signal Processing*,, vol. 45, number. 7, pp. 812–820,July 1998.

Published in Conferences

- † S. K. Patra and B. Mulgrew, "Co-Channel Interference Supression using a Fuzzy Filter," *Proceedings of European Signal Processing Conference-1998*, (Islands of Rhodes, GREECE), pp. 1609-1612, 8-11 September 1998.
- † S. K. Patra, M. Mulgrew and P. M.Grant, "Subset Centre Selection with Fuzzy Implemented Radial Basis Function Equalisers,"*Proceedings of 1st International Symposium on Communication System and Digital Signal Processing*, (Sheffield Hallan University, Sheffield, U.K.),pp. 21-25, 6-8 April 1998.
- † S. K. Patra and B. Mulgrew, "Computational Aspects for Adaptive Radial Basis Function Equalizer Design," in *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 1, (Hong Kong), pp. 521–524, 9-12 June 1997.