

Archvied in Dspace@nitr

<http://dspace.nitrkl.ac.in/dspace>

# Adaptive Equalisation of Communication Channels Using ANN Techniques

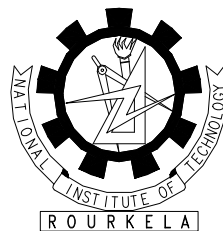
*For the degree of*

Doctor of Philosophy

**by**

Susmita Das

Under the Supervision of  
Dr. J.K.Satapathy, Professor



DEPARTMENT OF ELECTRICAL ENGINEERING  
NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA-769008

ORISSA  
2004

---

---

# ABSTRACT

---

Channel equalisation is a process of compensating the disruptive effects caused mainly by Inter Symbol Interference in a band-limited channel and plays a vital role for enabling higher data rate in digital communication. The development of new training algorithms, structures and the selection of the design parameters for equalisers are active fields of research which are exploiting the benefits of different signal processing techniques. Designing efficient equalisers based on low structural complexity, is also an area of much interest keeping in view of real-time implementation issue. However, it has been widely reported that optimal performance can only be realised using nonlinear equalisers. As Artificial Neural Networks are inherently nonlinear processing elements and possess capabilities of universal approximation and pattern classification, these are well suited for developing high performance adaptive equalisers.

This proposed work has significantly contributed to the development of novel equaliser structures with reduced structural complexity in the neural network paradigm based on both the feedforward neural network (FNN) and the recurrent neural network (RNN) topologies. Various innovative techniques like hierarchical knowledge reinforcement, genetic evolutionary concept, transform domain based approach and sigmoid slope tuning using fuzzy logic approach have been incorporated into an FNN framework to design highly efficient equaliser structures. Subsequently, novel hybrid configurations using cascaded modules of RNN and FNN have also been proposed in this thesis work. Further, suitable modifications in the Back-Propagation and Real-Time-Recurrent-Learning algorithms have been incorporated to update the connection weights of the proposed structures. Significant performance improvement over the conventional FNN and RNN based equalisers, faster adaptation and ease of implementation in real-time applications are the major advantages of the proposed neural equalisers. Exhaustive simulation studies carried out on various linear and nonlinear channels verify the efficacy of the proposed neural equalisers.

Further, all the proposed FNN based equalisers are of decision feedback type as inclusion of this technique significantly improves the performance alongwith a

considerable reduction in the structural complexity. A proper selection of feedforward order, decision delay and feedback order is a challenging task in such equalisers as these key design parameters play a crucial role for an impressive performance. A detailed study of various factors influencing the bit error rate performance of optimal Bayesian equaliser has been undertaken in the present work. This study has given an insight for proposing of a novel approach in the parameter selection issue, which can eliminate the use of cumbersome procedure of determining these design parameters from graphical analysis. Thus a major breakthrough has been achieved in successfully evaluating these parameters of the equaliser structure. The new methodology and its logical interpretation that led to the development of some empirical relationships have emerged as a powerful tool for selecting the key design parameters directly from the channel characteristics.

### **Keywords:**

*Adaptive Equalisation, Inter Symbol Interference, Communication Channel, Decision Feedback Equaliser, Nonlinear Equalisers, Feedforward Neural Network, Recurrent Neural Network, Bayesian Equaliser, Channel States, Optimal Decision Boundary, Bit-Error-Rate, Signal-to-Noise Ratio, Hierarchical Knowledge Reinforcement, Orthogonal Basis Function, Sigmoidal Activation function, Discrete Cosine Transform, Fuzzy Tuning, Hybrid Structure, Cascaded Equaliser, Pseudo-local gradient*

---

# CERTIFICATE

---

This is to certify that the thesis work entitled “**Adaptive Equalisation of Communication Channels Using ANN Techniques**” submitted by Mrs. Susmita Das is a record of bonafide work carried out under my supervision. This thesis fulfills all the requirements for the award of Doctor of Philosophy. This research is based on her own work and has not been submitted elsewhere for the award of any other degree to the best of my knowledge and belief.

**Dr. Jitendriya Kumar Satapathy, Ph. D. (Bradford)**

**Professor & Head**

Department of Electrical Engineering,  
National Institute of Technology,  
Rourkela 769008, India

---

## **DECLARATION OF ORIGINALITY**

---

I, **Susmita Das**, hereby submitting the thesis entitled “**Adaptive Equalisation of Communication Channels Using ANN Techniques**” for the award of Doctor of Philosophy, declare that the research work undertaken by me was based on exhaustive study of published work and development of novel techniques. Further, this research has been carried out entirely by myself in the Department of Electrical Engineering at National Institute of Technology, Rourkela, India.

**(Susmita Das)**

National Institute of Technology,  
Rourkela, India

September 2004

---

## ACKNOWLEDGEMENTS

---

I have been very fortunate in having Dr. Jitendriya Kumar Satapathy, Professor & Head, Department of Electrical Engineering, National Institute of Technology(NIT), Rourkela as my thesis supervisor. He inspired my interest in neural networks and communications, taught me the essence and principles of research and guided me through the completion of the thesis work. Working with Prof. Satapathy is highly enjoyable, inspiring and rewarding experience. I am highly indebted to him and express my deep sense of gratitude for his guidance and support.

I humbly acknowledge the creative suggestions and constructive criticism of Prof. G. Panda, Prof. P.C. Panda, and Prof. P.K. Nanda, committee members, while scrutinising my research results. I express my sincere thanks to Prof. S. Patra for his valuable comments and advice.

I am highly indebted to the authorities of NIT, Rourkela for providing me various infrastructures like library, computational facility and Internet access, which have been very useful.

I owe my largest debt to my family. Husband Somanath instilled strength especially at times when life was tough and supported me throughout this difficult period with endurance. Son Siddharth and daughter Eeshanee have adjusted to my long working hours away from home so nicely and have never complained. My parents, brother, sister and sister-in-law have given me love and constant encouragement over the years.

I express special thanks to all my friends, colleagues and seniors who have inspired me a lot during the course of research work.

I dedicate this thesis to my family and friends.

# BIO-DATA

**NAME** : **SUSMITA DAS**

**DATE OF BIRTH** : 24<sup>th</sup> June, 1967

**ADDRESS** : D -11, NIT Campus  
Rourkela-769008, Orissa, India  
☎: +91 661 2476518, Extn-2402(O)  
+91 661 2472903 (R)  
e-mail : [sdas@nitrkl.ac.in](mailto:sdas@nitrkl.ac.in)  
  
Institute website – [www.nitrkl.ac.in](http://www.nitrkl.ac.in)

## ACADEMIC QUALIFICATION :

- B.Sc. Engg. (HONS) in Electrical Engineering from College of Engineering and Technology, Bhubaneswar, Orissa
- M.Sc. Engg. in Electrical Engineering with specialization in Electronics System and Communication, National Institute of Technology, Rourkela ,Orissa

## MEMBERSHIP OF PROFESSIONAL SOCIETIES :

- Life Member of Indian Society of Technical Education, India
- Life Member of Institution of Engineers, India
- Member of Institution of Electronics and Telecom Engineers, India

## PROFESSIONAL EXPERIENCE :

Joined as Lecturer in the Department of Electrical Engineering, National Institute of Technology, Rourkela in September 1991 and continuing as Lecturer (Selection Grade).

## MAJOR RESEARCH AREAS :

Digital Signal Processing, Digital communication, Artificial Neural Networks, Fuzzy logic, Channel Equalisation, System Identification



*Dedicated to my parents*

---

# CONTENTS

---

Abstract .....	i
Certificate .....	iii
Declaration of Originality .....	iv
Acknowledgements .....	v
Bio-Data .....	vi
Contents .....	vii
List of Figures .....	x
List of Tables .....	xiv
Acronyms and Abbreviations .....	xv
Nomenclatures .....	xvii
<b>Chapter 1: Introduction .....</b>	<b>01</b>
1.1 Motivation for work .....	03
<b>1.2 Literature survey .....</b>	<b>04</b>
1.3 Thesis contribution .....	07
1.4 Thesis layout .....	11
<b>Chapter 2: Background .....</b>	<b>13</b>
2.1 The channel equaliser .....	13
2.1.1 Adaptive equaliser classification .....	15
2.2 FIR model of a channel .....	16
2.3 Optimal symbol-by-symbol equaliser .....	18
2.3.1 Channel states .....	20
2.3.2 Bayesian equaliser decision function .....	21
2.3.3 Bayesian equaliser with decision feedback .....	22
2.4 Symbol-by-symbol linear equaliser .....	23
2.4.1 Decision feedback equaliser .....	24
2.5 Symbol-by-symbol adaptive non-linear equalisers .....	26
2.5.1 A multilayer perceptron decision feedback equaliser: MLPDFE ..	27
2.5.2 Recurrent neural network equaliser .....	28
2.6 Conclusion .....	29

<b>Chapter 3: Factors Influencing Equaliser's Performance and Parameter Selection</b> .....	<b>30</b>
3.1 Factors influencing the performance of optimal symbol-by-symbol (Bayesian) equaliser .....	31
3.1.1 Additive noise level .....	31
3.1.2 Decision delay .....	37
3.1.3 Equaliser order .....	41
3.1.4 Effect of decision feedback .....	43
3.1.5 Importance of selection of ' $n_b$ ' and ' $d$ ' in a DFE structure .....	53
3.2 A new approach for selection of equaliser parameters .....	56
3.2.1 Equaliser without decision feedback .....	56
3.2.1.1 Symmetric channels .....	57
3.2.1.2 Asymmetric channels .....	61
3.2.2 Equaliser with decision feedback .....	68
3.3 Selection of feedforward order ' $m$ ' .....	70
3.4 Conclusion .....	74
<b>Chapter 4: Proposed FNN Based Equalisers</b> .....	<b>75</b>
4.1 Hierarchical knowledge based feedforward neural network (HKFNN) equaliser .....	76
4.1.1 Learning algorithm .....	76
4.2 Orthogonal basis function based feedforward neural network (OBFNN) equaliser .....	79
4.2.1 Development of the concept for weight adaptation in the proposed structure .....	81
4.2.2 Learning algorithm .....	84
4.3 Transform domain based feedforward neural network (TDFNN) equaliser .....	86
4.3.1 Learning algorithm .....	88
4.4 A Fuzzy tuned feedforward neural network (FZTUNFNN) equaliser ...	90
4.4.1 Description of the proposed concept of sigmoid slope tuning ...	91
4.4.1.1 Description of fuzzy logic controller technique .....	91
4.5 Simulation study .....	96
4.5.1 Performance analysis of the proposed FNN based Equalisers .....	100
4.6 Conclusion .....	116

<b>Chapter 5: Proposed RNN Based Cascaded Equalisers. . . . .</b>	<b>117</b>
5.1 FNN – RNN cascaded (FRCS) equaliser . . . . .	119
5.1.1 Description of the proposed structure . . . . .	119
5.1.2 Development of a novel concept for network adaptation . . . . .	121
5.1.3 Training algorithm. . . . .	125
5.2 Hierarchical knowledge reinforced FNN-RNN cascaded (HKFRCS) equaliser . . . . .	126
5.2.1 Training algorithm . . . . .	126
5.3 RNN-FNN cascaded (RFCS) equaliser . . . . .	129
5.3.1 Training algorithm . . . . .	129
5.4 RNN-Transform cascaded (RTCS) equaliser . . . . .	132
5.4.1 Training algorithm . . . . .	134
5.5 A fuzzy tuned recurrent neural network (FZTUNRNN) equaliser . . . . .	137
5.5.1 Details of proposed method . . . . .	137
5.6 Simulation study and performance analysis of the proposed RNN based equalisers . . . . .	138
5.7 Conclusion . . . . .	155
<b>Chapter 6: Conclusion . . . . .</b>	<b>156</b>
6.1 Summary and achievement of the thesis . . . . .	157
6.2 Limitations of the work . . . . .	163
6.3 Scope of future work . . . . .	164
<b>References . . . . .</b>	<b>165</b>
<b>Appendix-A : Back Propagation algorithm . . . . .</b>	<b>174</b>
<b>Appendix-B : RTRL algorithm . . . . .</b>	<b>175</b>
<b>Appendix-C : A fuzzy controller structure . . . . .</b>	<b>177</b>
<b>Appendix-D : Transfer Functions of the channels . . . . .</b>	<b>179</b>
<b>Appendix-E : Frequency Responses of the channels . . . . .</b>	<b>180</b>
<b>List of Publications . . . . .</b>	<b>183</b>

---

## LIST OF FIGURES

---

2.1	Base band model of a digital communication system . . . . .	14
2.2	An adaptive equaliser configuration . . . . .	14
2.3	Adaptive equaliser classification . . . . .	15
2.4	FIR model of a channel . . . . .	17
2.5	Discrete time model of a digital communication system . . . . .	18
2.6	Structure of a linear equaliser. . . . .	23
2.7	Structure of a linear decision feedback equaliser . . . . .	24
2.8	Optimal decision boundary and noise-free channel states . . . . .	26
2.9	Multilayer perceptron decision feedback equaliser . . . . .	27
2.10	Recurrent neural network equaliser . . . . .	28
3.1a	Noise-free channel states and optimal decision regions. . . . .	32
3.1b-d	Optimal decision boundary and scatter plots (ideal channel). . . . .	33
3.1e	Optimal Bayesian performance curve (ideal channel). . . . .	33
3.2	Effect of ISI on optimal Bayesian BER performance characteristics . . .	34
3.3a-c	Optimal decision boundary and scatter plots (Channel: $H_1(z)$ ) . . . . .	35
3.4	Combined effect of ISI and additive noise on optimal BER performance . . . . .	36
3.5a	Optimal Bayesian BER performances varying the delay parameter (Channel: $H_1(z)$ ). . . . .	38
3.5b-d	Decision region plots and noise-free channel states with various delays (Channel: $H_1(z)$ ) . . . . .	38
3.6a	Optimal Bayesian BER performances varying the delay parameter (Channel: $H_5(z)$ ). . . . .	39
3.6b-e	Decision region plots and noise-free channel states with different delays (Channel: $H_5(z)$ ) . . . . .	39
3.7a-f	Decision regions and noise-free channel states with different delays (Channel: $H_{11}(z)$ ). . . . .	40
3.7g	Optimal BER performance comparison for various delay parameter (Channel: $H_{11}(z)$ ) . . . . .	40
3.8	Optimal Bayesian BER performance curves varying equaliser's feedforward order (Channels: $H_1(z)$ , $H_5(z)$ , $H_{11}(z)$ and $H_6(z)$ ) . . . . .	42
3.9a-d	Effect of feedback vector on the optimal decision regions and the noise-free channel states (Channel: $H_6(z)$ ) . . . . .	45

3.10a-d	Decision region plots without feedback and for different values of delay parameter with feedback (Channel: $H_6(z)$ ).	47
3.11a-c	Decision region plots for different combinations of feedback order and delay parameter (Channel: $H_6(z)$ ).	48
3.12a-c	Optimal performance comparisons varying the delay parameter and increasing the feedback order(Channel: $H_6(z)$ ).	49
3.13a-e	Effect of feedback order and delay parameter on the optimal decision region plots (Channel: $H_{11}(z)$ ).	50
3.14a-c	Effect of feedback order and delay parameter on the decision region plots (Channel: $H_1(z)$ ).	51
3.14d	Optimal Bayesian performance comparisons (Channel: $H_1(z)$ ).	51
3.15a-b	Optimal Bayesian performance curves varying the feedback order for fixed feedforward order and delay parameter (Channels: $H_{11}(z)$ and $H_6(z)$ ).	52
3.16a-c	Optimal Bayesian Performance comparisons for different combinations of feedback order and delay parameter (Channels: $H_6(z)$ , $H_1(z)$ and $H_5(z)$ ).	54
3.16d-e	Optimal Bayesian Performance comparisons for different combinations of feedback order and delay parameter (Channels: $H_{10}(z)$ and $H_{11}(z)$ ).	55
3.17	Amplitudes of channel tap-coefficients and decision delay selection (Channel: $H_5(z)$ ).	59
3.18	Amplitudes of channel tap-coefficients and decision delay selection (Channel: $H_{13}(z)$ ).	59
3.19a-b	Optimal BER performance comparisons of symmetric channels varying delay values (Channels: $H_5(z)$ and $H_{13}(z)$ ).	60
3.20	Amplitudes of channel tap-coefficients and decision delay selection (Channel: $H_1(z)$ ).	63
3.21	Amplitudes of channel tap-coefficients and decision delay Selection (Channel: $H_4(z)$ ).	63
3.22a-b	Optimal BER performance comparisons of asymmetric channels varying delay values (Channels: $H_1(z)$ and $H_4(z)$ ).	64
3.23	Amplitudes of channel tap-coefficients and decision delay selection (Channel: $H_9(z)$ ).	66
3.24	Amplitudes of channel tap-coefficients and decision delay selection (Channel: $H_{11}(z)$ ).	66
3.25a-b	Optimal BER performance comparisons of asymmetric channels varying delay values (Channels: $H_9(z)$ and $H_{11}(z)$ ).	67
3.26	Channel transmitted symbol sequence matrix illustrating the	

relationship between $d$ and $n_b$ . . . . .	68
3.27a-b Optimal BER performance comparisons varying equaliser's feedforward order (Channels: $H_5(z)$ and $H_6(z)$ ). . . . .	71
3.27c-d Optimal BER performance comparisons varying equaliser's feedforward order (Channels: $H_1(z)$ and $H_{11}(z)$ ). . . . .	72
3.28 Effect of increasing equaliser's feedforward order on BER performance loss incurred for a given SNR change (Channel: $H_5(z)$ ). . . . .	73
4.1 Hierarchical knowledge based FNN equaliser structure . . . . .	77
4.2 Orthogonal basis function based FNN equaliser structure. . . . .	80
4.2.1 An example of proposed OBFNN structure. . . . .	82
4.2.2 Expanded view of the OBFNN structure. . . . .	82
4.2.3 Error distribution through a multiplier unit . . . . .	82
4.3 Transform domain based FNN equaliser structure. . . . .	87
4.4 Schematic block diagram of a fuzzy logic controller. . . . .	93
4.5 Block diagram of fuzzy logic approach for tuning sigmoid slope . . . . .	93
4.6a-c Sigmoid activation function at different time index. . . . .	95
4.7 Simulation model of an adaptive equaliser. . . . .	96
4.8a-e Structures of proposed FNN based equalisers . . . . .	98
4.9 BER performance comparison of proposed FNN based equalisers for channel: $H_1(z)$ . . . . .	104
4.10 BER performance comparison of proposed FNN based equalisers for channel: $H_2(z)$ . . . . .	104
4.11 BER performance comparison of proposed FNN based equalisers for channel: $H_3(z)$ . . . . .	105
4.12 BER performance comparison of proposed FNN based equalisers for channel: $H_4(z)$ . . . . .	105
4.13 BER performance comparison of proposed FNN based equalisers for channel: $H_5(z)$ . . . . .	106
4.14. BER performance comparison of proposed FNN based equalisers for channel: $H_6(z)$ . . . . .	106
4.15 BER performance comparison of proposed FNN based equalisers for channel: $H_7(z)$ . . . . .	107
4.16 BER performance comparison of proposed FNN based equalisers for channel: $H_8(z)$ . . . . .	107
4.17 BER performance comparison of proposed FNN based equalisers for channel: $H_9(z)$ . . . . .	108
4.18 BER performance comparison of proposed FNN based equalisers for channel: $H_{10}(z)$ . . . . .	108

4.19	BER performance comparison of proposed FNN based equalisers for channel: $H_{11}(z)$ . . . . .	109
4.20.	BER performance comparison of proposed FNN based equalisers for channel: $H_{12}(z)$ . . . . .	109
4.21	BER performance comparison of proposed FNN based equalisers for channel: $H_{14}(z)$ . . . . .	110
4.22	BER performance comparison of proposed FNN based equalisers for channel: $H_{15}(z)$ . . . . .	110
4.23a-b	Comparison of HKFNN equaliser with CFNN w.r.t. training samples . .	112
4.24a-b	Comparison of OBFNN equaliser with CFNN w.r.t. training samples. .	113
4.25a-b	Comparison of TDFNN equaliser with CFNN w.r.t. training samples. .	114
4.26a-b	Comparison of FZTUNFNN equaliser with CFNN w.r.t. training samples . . . . .	115
5.1	Cascaded configurations . . . . .	118
5.2	FNN-RNN cascaded equaliser structure. . . . .	120
5.2.1	An example of proposed FRCS equaliser . . . . .	123
5.3	Hierarchical knowledge reinforced FNN-RNN cascaded equaliser structure. . . . .	127
5.4	RNN-FNN cascaded equaliser structure. . . . .	130
5.5	RNN-Transform cascaded equaliser structure . . . . .	133
5.6	Fuzzy logic approach for tuning sigmoid slope . . . . .	137
5.7a-e	Structures of proposed RNN based equalisers . . . . .	139
5.8	BER performance comparison of RNN based cascaded equalisers for channel: $H_1(z)$ . . . . .	144
5.9	BER performance comparison of RNN based cascaded equalisers for channel: $H_2(z)$ . . . . .	144
5.10	BER performance comparison of RNN based cascaded equalisers for channel: $H_3(z)$ . . . . .	145
5.11	BER performance comparison of RNN based cascaded equalisers for channel: $H_5(z)$ . . . . .	145
5.12	BER performance comparison of RNN based cascaded equalisers for channel: $H_6(z)$ . . . . .	146
5.13	BER performance comparison of RNN based cascaded equalisers for channel: $H_7(z)$ . . . . .	146
5.14	BER performance comparison of RNN based cascaded equalisers for channel: $H_8(z)$ . . . . .	147
5.15	BER performance comparison of RNN based cascaded equalisers for channel: $H_9(z)$ . . . . .	147
5.16	BER performance comparison of RNN based cascaded equalisers	



	for channel: $H_{11}(z)$ .....	148
5.17	BER performance comparison of RNN based cascaded equalisers for channel: $H_{14}(z)$ .....	148
5.18a-b	Comparison of FRCS equaliser with CRNN w.r.t training samples . . . .	150
5.19a-b	Comparison of HKFRCS equaliser with CRNN w.r.t. training samples. .	151
5.20a-b	Comparison of RFCS equaliser with CRNN w.r.t. training samples . . . .	152
5.21a-b	Comparison of RTCS equaliser with CRNN w.r.t. training samples . . . .	153
5.22a-b	Comparison of FZTUNRNN equaliser with CRNN w.r.t. training samples .....	154
C.1	A fuzzy controller structure .....	177

---

## LIST OF TABLES

---

2.1	Channel states calculation. . . . .	21
3.1	Transmitted symbol sequences and received sample vectors . . . . .	44
3.2	Optimal combination of parameters of a Bayesian DFE . . . . .	53
3.3	Optimal values of parameters for equalisers (with decision feedback) . . . . .	69
4.1	Fuzzy rule table . . . . .	93
4.2	Structural configuration comparison of the proposed FNN based equalisers with conventional FNN. . . . .	97
4.3	Learning parameters of proposed FNN based equaliser structures . . . . .	99
5.1	Structural configuration comparison of the proposed RNN based equalisers with conventional RNN . . . . .	140
5.2	Learning parameters of proposed RNN based equaliser structures . . . . .	141
6.1	Performance analysis of proposed HKFNN equaliser. . . . .	158
6.2	Performance analysis of proposed OBFNN equaliser. . . . .	159
6.3	Performance analysis of proposed TDFNN equaliser. . . . .	160
6.4	Performance analysis of proposed FZTUNFNN equaliser . . . . .	160
6.5	Performance analysis of proposed FRCS equaliser. . . . .	161
6.6	Performance analysis of proposed HKFRCS equaliser. . . . .	162
6.7	Performance analysis of proposed RFCS equaliser. . . . .	162
6.8	Performance analysis of proposed RTCS equaliser . . . . .	162
6.9	Performance analysis of proposed FZTUNRNN equaliser. . . . .	163

---

## ACRONYMS & ABBREVIATIONS

---

ANN	artificial neural network
AWGN	additive white gaussian noise
BER	bit error rate
BP	back propagation
COA	center of area
CFNN	conventional feedforward neural network
CRNN	conventional recurrent neural network
DCT	discrete cosine transform
DCS	digital communication system
DFE	decision feedback equaliser
DHT	discrete Hartley transform
DST	discrete sine transform
DSP	digital signal processing
EA	evolutionary algorithm
FAF	fuzzy adaptive filter
FIR	finite impulse response
FLC	fuzzy logic controller
FNN	feedforward neural network
FPGA	field programmable gate array
FRCS	FNN-RNN cascaded structure
FZTUNFNN	fuzzy tuned feedforward neural network
FZTUNRNN	fuzzy tuned recurrent neural network
HKFNN	hierarchical knowledge reinforced feedforward neural network
HKFRCS	hierarchical knowledge reinforced FNN-RNN cascaded structure
IDCT	inverse discrete cosine transform
i.i.d.	independent identically distributed
IIR	infinite impulse response
ISI	inter symbol interference

LMS	least mean squares
LTE	linear transversal equaliser
LN	large negative
LP	large positive
MAP	maximum a-posteriori probability
MBER	minimum bit error rate
MLP	multilayer perceptron
MLPDFE	multilayer perceptron decision feedback equaliser
MLSE	maximum likelihood sequence estimation
MMSE	minimum mean square error
MSE	mean square error
MN	medium negative
MP	medium positive
NN	neural network
OBF	orthogonal basis function
OBFNN	orthogonal basis function based feedforward neural network
PAM	pulse amplitude modulation
pdf	probability density function
PID	proportional integral derivative
QAM	quadrature amplitude modulation
RBF	radial basis function
RFCS	RNN-FNN cascaded structure
RLS	recursive least squares
RNN	recurrent neural network
RDT	real-valued discrete transform
RTRL	real time recurrent learning
SNR	signal to noise ratio
SVM	support vector machine
SN	small negative
SP	small positive
TDFNN	transform domain based feedforward neural network
TDL	tapped delay line
VLSI	very large scale integration

w.r.t.	with respect to
ZE	zero
ZF	zero forcing

---

# NOMENCLATURES

---

$a_i$	tap coefficients of channel
$cn$	center of membership function
$c_j(n)$	internal activity of a neuron in the $j^{\text{th}}$ layer at time index $n$
$C(z)$	inverse channel response
$d$	decision delay
$d_o(n)$	desired response at time index $n$
$d_{max}$	maximum delay
$d_{min}$	minimum delay
$d_{opt}$	optimal value of decision delay
$E$	expectation operator
$e(n)$	error at time index $n$
$err_{rnn-node}(n)$	error at the node of RNN module at time index $n$
$err_{fnn-node}(n)$	error at the node of FNN module at time index $n$
$e_{\mathcal{N}}(n)$	error at the output of normalization block at time index $n$
$e_{\mathcal{T}}(n)$	error at the output of transform block at time index $n$
$f$	an arbitrary variable
$F$	sigmoidal activation function in the neuron
$G$	Bayesian decision function
$\mathbf{H}$	channel matrix
$H(z)$	$z$ -transform of impulse response of equaliser
$i$	an arbitrary variable
$j$	layer index in the proposed FNN based equalisers
$J(n)$	cost function
$k$	an arbitrary variable
$l$	an arbitrary variable
$m$	feedforward order of the equaliser
$n$	time index
$n_a$	channel order
$n_b$	feedback order of the equaliser

$n_{b(opt)}$	optimal length of feedback vector
$n_f$	states of feedback order
$n_s$	noise-free channel states
$ncol$	column span of the transmitted symbol sequence matrix
$nf$	number of nodes in the FNN module of cascaded structures
$nr$	number of nodes in the RNN module of cascaded structures
$nn$	output layer index in FNN based structures
$nx$	number of external inputs to the equaliser structure
$N(n)$	additive white Gaussian noise at time index $n$
$P_i(n)$	Bayesian decision variable
$p_j^{(i)}$	<i>a-priori</i> probability
$P_{kl}^j$	sensitivity parameter, a triply indexed set of variables
$P^*(n)$	minimum error probability
$p_N$	probability density function of noise
$r(n)$	received scalar sample at time index $n$
$\hat{r}(n)$	noise-free received scalar sample at time index $n$
$\hat{\mathbf{r}}(n)$	noise-free received signal vector at time index $n$
$\mathbb{R}^m$	$m$ dimensional space
<b>S</b>	transmitted symbol sequence matrix
$s(n)$	transmitted sample at time index $n$
$\mathbf{s}(n)$	transmitted symbol vector at time index $n$
$s(n-d)$	delayed value of transmitted symbol
$\hat{s}(n-d)$	estimated sample at time index $n$ with decision delay $d$
$\mathbf{s}_{f,j}$	feedback state vector
$\mathcal{T}$	transform matrix
$tap_{max}$	maximum amplitude of channel tap coefficient
$th$	threshold value at a neuron
$\mathbf{u}(n)$	vector input to RNN module at time index $n$
$\mathbf{w}$	feedforward weight vector
$\mathbf{w}_b$	feedback weight vector
$\mathbf{x}(n)$	input vector to the equaliser at time index $n$
$y(n)$	output of equaliser at time index $n$
$\mathbf{z}(n)$	vector output from FNN module at time index $n$
$z_{\mathcal{T}}(n)$	transformed signal at time index $n$

$z_N(n)$	normalized signal at time index $n$
$\theta$	learning-rate parameter of weights at output end of cascaded structures
$\delta(n)$	local gradients (error terms) at each node of neural network
$\delta_{\text{fnn-node}}(n)$	local gradients (error terms) in FNN module
$\delta_{\text{rnn-node}}(n)$	local gradients (error terms) in RNN module
$\phi$	slope of the sigmoid function
$\eta$	learning-rate parameter for adaptation of weights of FNN based structure
$\beta$	learning-rate parameter for adaptation of thresholds of FNN based structure
$\lambda$	learning-rate parameter of weight adaptation in RNN module of cascaded structures
$\alpha$	momentum parameter
$\gamma$	power scaling parameter
$\Delta \delta(n)$	rate of change of error term at a node of FNN
$\Delta \delta_{\text{rnn-node}}(n)$	rate of change of error term at a node of RNN
$\Delta \phi(n)$	correction in slope of sigmoidal activation function
$\sigma_N^2$	channel noise variance
$\sigma_S^2$	transmitted signal power
$\sigma_i$	spread of fuzzy membership function
$\delta_{kj}$	Kronecker delta
$\chi$	an arbitrary constant
$\varepsilon$	a small constant
A	a set of the neurons of RNN module
B	a set of the neurons of FNN module
C	a set of the visible neurons in RNN module
$\mathbf{R}_{m,d}$	a set of channel states
$\mathbf{R}_{m,d,j}$	a subset of channel states
$\mathcal{B}(n)$	power of the transmitted signal



---

# CHAPTER 1

## Introduction

---

In today's world, digital transmission has a tremendous impact on the human civilization. There has been a sea change in modern day living and the credit goes to the development in digital communication technology. With expanding communication networks, as we move towards a more information centric world and the demand for very high speed efficient data transmission over physical communication channels increases, communication system engineers face ever-increasing challenges in utilising the available bandwidth more efficiently so that new services can be handled in a flexible way. The objective of any digital communication system is to convey information, with the minimum possible introduction of error. Some typical forms of transmission media are open-wire lines, coaxial cables, microwave radio, optical fibers, satellite links etc. These media differ, essentially, in the volume of data per unit time which they can transmit. This data rate is limited by the noise and distortion introduced in the communication channel.

The distortions (phase-delay variations) introduced in the communication channel cause the transmitted symbols to spread and overlap over successive time intervals, resulting in a phenomenon, known as Inter Symbol Interference (ISI). In addition to ISI, the transmitted symbols are subjected to other impairments such as thermal noise, impulse noise and non-linear distortions arising from the modulation and demodulation process, cross talk interference, the use of amplifiers and converters etc. All the signal processing techniques used at the receiver end to combat distortions introduced due to channel impairments and recover the transmitted symbols accurately, are referred to as equalisation schemes.

A common means of overcoming this problem has thus been to introduce an inverse filter into the receiver to equalise the channel. This simplistic view offers a satisfactory solution provided the channel transfer function is known and its inverse is

convergent. However, such an inverse will be unstable in non-minimum phase channels (zeros outside the unit circle in the  $z$ -plane). Further, the designer does suffer from the disadvantage of having no a-priori knowledge of the channel transfer function and this function will be time-varying when the channel conditions are not stationary. For these reasons, in realistic situations equalisers are commonly adaptive in nature; that is, they automatically adjust their parameters when subjected to some external stimuli. Adaptive equalisers are characterised in general by their structures, the learning algorithms and the use of training sequences. Bandwidth efficient data communication requires the use of adaptive equalisers. Adaptive equalisation algorithms need to exhibit some form of learning and this learning property is naturally found in artificial neural networks. Hence, many avenues have been established in the application of neural networks to adaptive equalisation problems.

By viewing equalisation as a pattern classification problem in which the optimal decision boundary is highly nonlinear, the solution offered by a linear equaliser is inherently sub-optimal. This drawback also has motivated the development of efficient non-linear equalisers for optimising the performance. Artificial neural networks are parallel distributed structures in which many simple interconnected elements (neurons) simultaneously process information and adapt themselves to learn from past patterns. Attractive properties of neural networks relevant to the equalisation problem are adaptive capability, finite memory, the ability to form nonlinear decision boundaries and efficient hardware implementation. Neural networks based equalisers have been applied to this field in recent past achieving better performance than conventional methods. This thesis work mainly deals with the development of novel adaptive equalisers on the framework of feedforward neural network and recurrent neural network topologies which have outperformed their existing counterparts. The primary objective of the proposed work is to design neural equalisers on reduced structural framework keeping in mind the real-time implementation issue, as a reduced size network means a lesser computational cost and a more economical hardware implementation. Suitable modifications in the popular Back-Propagation and Real-Time-Recurrent-Learning algorithms also have been incorporated for faster adaptation of the proposed neural equaliser structure parameters. Selection of key design parameters of the equaliser is given much importance for optimising the performance and some empirical relationships also have been developed purely from induction.

This chapter begins with an exposition of the principal motivation behind the work undertaken in the thesis followed by a brief literature survey on equalisation in general and nonlinear equalisers in particular as discussed in Section 1.2. The contributions made in this thesis work have been outlined in Section 1.3. At the end, the layout of the thesis is presented in Section 1.4.

## 1.1 Motivation for Work

The adaptive equalisers have gone through many stages of development and refinement in the last few decades since their inception in late 1960[1]. In the recent past a lot of researches have been carried out on equalisation of data communication channels using a rich variety of Artificial Neural Network (ANN) techniques. These include the development of many novel architectures and efficient training algorithms. The underlying reasons for it are scientific curiosity and a desire to provide alternative engineering solutions to the problems. Different ANN architectures such as multilayer perceptron (MLP), radial basis function (RBF) and recurrent neural network (RNN) for constructing adaptive equalisers have been suggested in the literatures [2,3,4,5,6].

The channel equalisation is viewed as a classification problem [7,8,9,10], where the equaliser attempts to classify the input vector into a number of transmitted symbols. The optimal solution for symbol-decision equaliser derived using Bayes Decision Theory [11,12] is an inherently nonlinear problem. Therefore, structures which incorporate some degree of nonlinear decision making ability must be considered to achieve fully or near optimal performance. In the light of above, the linear transversal equaliser (LTE) does not achieve the full potential of equalisation process as the decision boundary is necessarily hyper planar. Though for minimum phase channels, the two classes in the observation space (2-PAM signalling) are linearly separable, the solution is not optimal. Also in case of non-minimum phase channels, a linear equaliser with zero delay is incapable of reconstructing the input signals even in a noise-free case. Further, employing non-zero delays, although the channel states are linearly separable, it can not realise the optimal boundary and the performance still remains sub-optimal due to linear classification by the decision boundary. Thus, the channel equalisation is basically a nonlinear problem regardless of whether a channel is minimum phase or non-minimum phase [13,14]. So, nonlinear structures are essential which can form decision boundaries beyond the capabilities of LTE. Taking this into consideration, ANN offers a much better solution to channel equalisation problem due to its pattern classification capability [15,16,17]. The prime advantage of using neural networks for adaptive equalisation is their capability to model any non-linear decision boundaries [14,18,19]. Further, structure selection for an ANN equaliser has always been a point of concern because a small size and hence a less complex structure is much easier to implement in real-time using VLSI circuits, DSP and FPGA [20,21,22] chips etc. Use of efficient structures having compact design will help in varieties of present day applications like mobile communication system [23], optical recording [24], magnetic hard disk storage [25] etc.

Considering all these issues and future requirement, this research work is motivated towards development of new nonlinear equaliser structures in the neural network paradigm which are superior in performance compared to conventional ones. The architectures of the proposed structures are based on Feedforward Neural Network (FNN) topology [6] having configurations of reduced structural complexity. It is obvious that in a reduced structural framework some performance loss may occur and hence a compromise is to be made between the complexity and performance while designing new equalisers. Decision feedback equalisation (DFE) is a powerful technique used in digital communication systems to eliminate ISI without enhancement of noise by using past decisions to subtract out a portion of the ISI. The multilayer perceptron based DFE offers a superior performance as a channel equaliser to that of conventional DFE, especially in high noise conditions [26,27,28]. This technique motivated the research work to utilise decision feedback in all the proposed FNN based equaliser structures. Amongst all the equalisers with symbol decision structures, the Adaptive Bayesian DFE [29] provides the best theoretical performance reducing the structural complexity and hence assessing its performance is considered to be valuable. The proper selection of DFE structure parameters namely delay order, the feedforward order and feedback order is crucial, as each one plays a significant role in the performance of an equaliser. With a motivation to optimise the performances of all the proposed FNN based configurations, the selection of decision feedback parameters is made with reference to the parameters obtained for the optimal Bayesian equaliser.

Development of equalisers of reduced structural complexity further motivated to work in another avenue of research, i.e. in the area of Recurrent Neural Network (RNN). RNN configurations are inherently of low complexity structures which exhibit highly nonlinear characteristics. In particular, RNNs are in some respect very similar to DFEs because in that outputs are fed back to the classifier to assist in subsequent decisions [30,31], however unlike DFEs RNNs may store additional information about the past signals in the form of an internal state. A simple two unit RNN is sufficient to model many communication channels encountered in practice. The RNN is used for the adaptive equalisation of linear and nonlinear channels and it is reported that the RNN equalisers have much improved performance in comparison to traditional linear equalisers [32,33,34].

## 1.2 Literature survey

Nyquist's telegraph transmission theory [35] in 1928 laid the foundation for digital communication over band-limited analogue channels. The research in channel equalisation in the 1960s was centered around the basic theory and structures of zero-forcing transversal or tapped-delay-line equalisers [36,37,38]. In 1960, Widrow [39]

presented a least- mean-square (LMS) algorithm which revolutionised the adaptive filtering schemes for the last couple of decades. 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 too 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 development of the maximum likelihood sequence estimation (MLSE)[40] using the Viterbi algorithm [41] in 1970's. But this application had a limitation in the sense that when the length of ISI was long, the structural complexity of equaliser had a dramatic increase. Another form of nonlinear equaliser which appeared around the same time was the infinite impulse response (IIR) form of linear adaptive equaliser, where the equaliser employed feedback [42] and was termed decision feedback equaliser (DFE). Other works carried out in this field in 1970's and 1980's were the development of fast convergence and/or computationally efficient algorithms like the recursive least square (RLS) algorithm [39], Kalman Filters [43] and RLS lattice algorithm [44]. Fractionally spaced equalisers (FSE) [45] were also developed during this period. A review of the development of equalisers till 1985 is available in [46].

In the late 1980's, the beginning of development in the field of ANN brought a new dimension to all spheres of research. Gibson et al. first applied the MLP structure to the channel equalisation problem [13,14,18]. These new forms of equalisers were computationally efficient than MLSE and could provide superior performance compared to the conventional adaptive equalisers with adaptive filters. Traditionally, the problem of equalisation has been considered equivalent to the inversion of the transmission channel. A different and innovative approach considered equalisation as a classification problem. The optimal solution for a symbol-by-symbol equaliser derived using Bayes classification theory is highly non-linear and hence application of non-linear structures is essential to achieve fully or near optimal performance. Further, the Bayesian solution for the symbol decision structure with decision feedback highlighted the importance of decision feedback technique with a clear geometric explanation [29]. A multilayer perceptron based DFE was proposed which also provided better bit error rate (BER) performance compared to a conventional DFE, because of its ability to form complex decision regions with non-linear boundaries [26,27].

Another form of ANN called Radial Basis Function (RBF) [6] gained much importance and thereafter equalisers constructed using RBF networks were reported [47,48]. Wang and Mendel applied RLS fuzzy filter and LMS fuzzy filter to nonlinear channel equalisation problems achieving performance level close to optimal equaliser [49]. Subsequently new training algorithms and efficient equaliser structures

using ANNs, RBFs and FAFs were developed [50,51,52]. Many possible approaches of using neural network for equalisation have been developed in the last few years. Kirkland et al. [53] applied feedforward neural networks to equalise the digital microwave radio channel in the presence of multipath fading. Peng et al. [54] modified the nonlinear activation function of the classical multilayer perceptron in order to take into account signals typically encountered, namely, PAM and QAM. Kechriotis et al.[32] applied fully recurrent neural networks trained with real-time-recurrent-learning algorithm (RTRL) [55] to the equalisation of non-minimum phase, partial response, and nonlinear channels. Chang et al. [56] introduced a neural-based decision feedback equaliser to perform equalisation of indoor radio channels. A wavelet NN [57, 58] trained with the recursive least squares (RLS) algorithm was used to equalise a non-linear transmission channel. Al-Mashouq et al. [59] used a FNN to perform equalisation and decoding in presence of severe ISI conditions; which outperforms classical structures formed by cascading a linear equaliser and a decoder. All these works show that neural networks can be successfully applied to the problem of channel equalisation.

Training schemes to optimise minimum Bit Error Rate (BER) of neural network based equalisers using fuzzy decision learning have also been developed [60]. Algorithms for training ANN equalisers to achieve MLSE performance with minimum BER criterion involving conditional distributed learning [61], Hopfield networks with mean field annealing [62], cellular neural networks with hardware annealing [63,64] have shown better equaliser performance. A number of efficient neural equalisers using single layer architecture with polynomial perceptron [12,65], functional link perceptron [66,67,68], polynomial lattice [69] have been developed. The lattice-based MLPDFE outperformed both the LMSDFE and MLPDFE in both time-invariant and time-varying channels [70]. Evolutionary algorithm provided a new optimisation technique for the solution of channel equalisation problem. The effectiveness of using an Evolutionary Algorithm (EA) for equalisation of a non-minimum phase channel using a feedforward MLP is highlighted [71]. The emerging machine learning technique called Support vector machines (SVMs) is proposed as a method for performing non-linear equalisation in communication systems [72,73]. In research work cited in [74] a strategy is proposed for designing the DFE using SVM, which is found to be computationally efficient and can be of great help in data storage system and slow time-varying communication links. However, the learning algorithm in the SVM needs to solve a quadratic programming problem and the optimisation method is somewhat computationally intensive. Recent research works have derived some adaptive linear and decision feedback minimum-BER (MBER) equalisers [75,76,77] which consider the BER to be a true performance indicator for equalisation instead of MSE criterion. A DFE using recurrent neural networks trained with

Kalman filter is developed with features of fast convergence and good performance using relatively short training symbols [78]. Recently, structure and property of MMSE linear and DFEs under realisability constraints have been analysed [79,80]. The computational complexity of optimal symbol-by-symbol equaliser using Bayesian solution can be reduced by signal space partitioning technique, but the number of hyperplanes can not be controlled. A new algorithm has been proposed to overcome such problem [81,82]. The recent advances in the field of nonlinear equalisation are centered on the design of efficient equaliser structures, development new approaches and faster training algorithms and proper selection of equaliser parameters for optimising its performance [83,84,85,86,87]. Also designing low complexity network for easier implementation has always been a challenging task amongst communication system designers and is quite an encouraging area of investigation.

### 1.3 Thesis contribution

The major focus of the present work is aimed at developing efficient neural equalisers with reduced structure configuration so as to make these attractive for easy implementation in real-time applications. All the proposed structures have been designed either on Feedforward Neural Network (FNN) or Recurrent Neural Network (RNN) framework. The widely used algorithms like Back-Propagation for FNN based structures and Real-Time-Recurrent-Learning Algorithm for RNN based structures could not be directly applied for training the proposed structures and hence suitable modifications have been included into the existing algorithms while estimating the local gradient of errors at different nodes looking into the respective structural paradigms. Further, the proposed neural equalisers have resulted in encouraging BER performance in comparison to their conventional counterparts being trained with much less training samples. The prime objective of the present research is to optimize the BER performance of the proposed equalisers with reduced structural configurations. It is seen that the key parameters for equaliser design like feedforward order ' $m$ ', feedback order ' $n_b$ ' and decision delay ' $d$ ' influence the performance significantly and hence selecting an optimum combination of these parameters is of great concern. In order to choose such a combination of these parameters, exhaustive studies have been carried out and the performance of the optimal symbol-by-symbol equaliser is evaluated under a broad range of parameter variations which has led the research to derive certain empirical relations for parameter selection in both feedback and without feedback conditions.

Development of novel equaliser structures in the FNN domain are based on the following techniques:

- Hierarchical knowledge reinforcement –

This new technique can be viewed as an enhancement in the knowledge base at the nodes of an existing multilayer feedforward configuration for arriving at a near optimal solution. In terms of hierarchy, nodes in the final layer occupy the highest level in the process of decision making and hence these experts are fed with more information to enrich their knowledge base. The popular Back-Propagation (BP) algorithm has been suitably modified for error back propagation at each node of the proposed structure for training.

- Orthogonal basis function expansion-

The orthogonal basis function (OBF) expansion technique is motivated by a genetic evolutionary concept of self breeding. Here, the decision at a node termed as expert opinion of a generation, undergoes an *orthogonal expansion* in two dimensions. One of the outputs possessing the knowledge base for that generation participates in taking the final decision; while the other one is allowed to pass on the information further to generate the expert opinion for the next generation and the process continues. Finally, a collective judgment based on the expert opinions evolved from decisions of individual generations gives a more rational and heuristic solution. Propagation of output error backwards and calculation of local gradients at each node become a difficult task as the OBF block is positioned in between the neurons of different layers. In order to circumvent such situation, a new technique has been evolved so that the BP algorithm can be applied.

- Transform domain approach-

Further, a hybrid configuration has also been presented where a *discrete cosine transform* (DCT) block with normalisation is embedded within the framework of a conventional FNN structure. Such a cascaded network representing a heterogeneous configuration proves to be efficient by learning faster and also performing better in comparison to a conventional FNN structure. The BP algorithm has been applied to adapt the weights in the proposed neural structure, but certain modifications in the algorithm has been incorporated taking into account the structural changes when compared with a conventional one.

- Sigmoid slope tuning by fuzzy logic controller approach-

In this thesis attempt has been made to improve the performance of conventional FNN equaliser with a reduced structure by adapting the slope of the sigmoidal activation function using *fuzzy logic control technique*. While the existing BP algorithm takes control in updating the network weights, the fuzzy controller approach adjusts the



slope of the sigmoidal activation function of all the nodes in the network based on the available error information making the proposed structure more adaptable. Also, the new equaliser can learn faster giving a satisfactory BER performance.

While in the FNN domain much emphasis has been given to the design of equaliser structures possessing low structural complexity, RNN topology too has been considered because even a reduced network structure in RNN domain can show better performance due to its inherent self-feedback configuration. Basically in this research work, cascading technique is utilised to evolve new topologies with RNN as the main module.

Development of novel cascaded equaliser structures in the RNN domain are based on the following techniques:

- FNN-RNN cascading-

The inputs to RNN module is directly fed from the outputs of FNN module and hence already pre-processed. This configuration being a hybrid one, the weight adaptation of different modules becomes a challenging task as no such direct algorithm exists for such task. The updation of the weights of the RNN module can be carried out straightforward using the RTRL algorithm and based on the output error. As the RTRL algorithm does not provide any explicit estimate about the local gradients at the RNN nodes, problem is encountered in updating weights of the FNN module directly. This bottleneck has motivated the present research in pursuing a new strategy named as *equivalence approach* to estimate pseudo local gradients at all the nodes of the RNN module and hence the weight adaptation of the FNN module is done by applying the BP algorithm.

- Hierarchical knowledge based FNN-RNN cascading-

An efficient equaliser structure is designed employing the concept of hierarchical knowledge reinforcement to the proposed FNN-RNN cascaded network. It is expected that by providing more information at the final processing layer in the RNN module its knowledge base is enriched and hence the performance improves.

- RNN-FNN cascading-

Another variant of the cascaded architecture, which is identical to the first hybrid structure mentioned, has been proposed with the only exception that the FNN and RNN modules are swapped. It can be inferred that such structure has an intermediate decision feed-back mechanism embedded into the network configuration. The weights of the FNN module at the output end can be updated using existing BP algorithm directly. After the estimation of the errors at the nodes of the RNN module is done, the RTRL algorithm is applied to update the weights of the RNN module.

- RNN-Transform cascading-

Further, in the previous cascading structure, replacing the FNN module by a *discrete cosine transform*(DCT) block with normalisation, a new structure is presented. However, such configuration has inherent constraints of direct application of BP algorithm to update the connection weights due to the positioning of transform block. In order to overcome such a scenario, a novel strategy has been deployed to propagate the output error back through the network.

- Fuzzy tuned sigmoid slope of RNN nodes

The concept of fuzzy logic control technique has been applied to tune the slope of sigmoidal activation functions of the RNN nodes to make the structure more adaptive keeping the structural complexity same as that of a pure RNN one.

Further in the present work, a detailed study on various factors influencing the BER performance of the Bayesian equaliser has been undertaken because such type provides the optimum performance for symbol-by-symbol type of equalisers. It is observed here that its Bayesian equaliser performance is influenced by additive noise level, decision delay ' $d$ ', feedforward order ' $m$ ' and feedback order ' $n_b$ '. Some of the important observations in this present study are as follows.

- The probability of those sample points of one class crossing the optimal decision boundary and entering to the wrong class increases depending on the noise severity, causing more and more misclassification.
- The number of noise-free channel states close to the optimal decision boundary varies with decision delay significantly. Also the channel states in close proximity to the optimal decision boundary are vulnerable and have undesirable effect on the optimal BER performance, i.e., when additive noise present in the communication system is more, a higher probability of misclassification is expected.
- The structural complexity increases with larger equaliser feedforward order ' $m$ ', which necessitates restriction of  $m$  to a certain order.
- It is observed that inclusion of the decision feedback increases the minimum distance between the two classes of channel states and hence the separation of noise-free channel states from the optimal decision boundary is more in comparison to that obtained without feedback. Hence the probability of received samples crossing the optimal decision boundary reduces considerably which in turn improves performance.
- The selection of proper combination of parameters like  $m$ ,  $n_b$  and  $d$  is essential to achieve optimal BER performance.

From the above important observations achieved by parameter variations, certain empirical relations have been derived in this present research work for choosing those key design parameters directly by examining the channel characteristics. Hence the parameters for satisfactory BER performance for equalisers (without feedback and with decision feedback) can be evaluated only by a simple examination of the type of channel model and its tap coefficients. This methodology has really opened up a new horizon for parameter selection problem in an efficient way, without going for exhaustive graphical study. The approach suggested in this present work also has been logically interpreted and verified from the various simulation results. A trade-off between structural complexity and performance loss has been well taken care of in choosing the feedforward order ' $m$ ', which has been restricted to the channel order ' $n_a$ '. This interpretation brings a new dimension to parameter selection issue in equalisers with and without decision feedback.

In summary, the proposed work relates to designing of novel equaliser structures employing both FNN and RNN topologies along with suitable modifications in the existing training algorithms and selection of key parameters in equaliser design. So far as this work is concerned, the main emphasis has been given to design equaliser configurations on a reduced structural framework. All the proposed equalisers have resulted faster learning and encouraging BER performances for various linear and nonlinear communication channel models, but the gains obtained are entirely channel dependent as observed from the exhaustive simulation studies in the present work.

## 1.4 Thesis layout

The rest of the thesis is organised as follows.

Chapter 2 presents the background of channel equalisation along with the optimal Bayesian equaliser structure. Most commonly referred linear and nonlinear equaliser structures and their training algorithms have been explained. Artificial Neural Network based structures are focussed here as these form the basis for development of novel neural equaliser structures and training algorithms pertaining to the proposed work.

Chapter 3 provides a detailed study on various factors influencing the BER performance of the optimal Bayesian equaliser. It is observed here that the decision function is dependent on decision delay ' $d$ ', equaliser feedforward order ' $m$ ' and the additive noise level. It is also observed that including decision feedback results significant improvement in the performance of a symbol-by-symbol Bayesian equaliser. Further the design of a DFE structure depends upon the proper selection of parameters like feedforward order, feedback order and decision delay for optimising

the BER performance. Certain empirical relations have been derived and logically explained in this research work based on the fundamental concepts of equaliser design. These critical design parameters can be chosen directly by examining the channel tap coefficients following the new methodology presented.

Chapter 4 is devoted to the design of efficient equaliser structures based on Feedforward Neural Network (FNN) topology and development of appropriate training algorithms to adapt the network weights. Various innovative techniques like hierarchical knowledge reinforcement, genetic evolutionary concept, transform domain based approach, sigmoid slope tuning using fuzzy logic concept etc. are incorporated into an FNN framework. Results have been presented for various simulation studies in real channels (both linear and nonlinear) to validate the efficacy of the proposed FNN structures.

Chapter 5 emphasises on designing of new cascaded equaliser structures in the RNN domain where RNN block being an integral module, other modules like FNN block, transform block etc. have been appended with it to supplement the decision for enhancing the performance. The cascading technique employed to evolve new topologies is based on various combinations of these blocks. In order to prove the efficacy of the proposed equaliser structures, equalisation of various channel models have been investigated and their performance improvements over the conventional ones are illustrated.

Chapter 6 summarises the thesis work undertaken, discusses its limitations and points out the possible directions for further research.

---

## CHAPTER 2

# Background

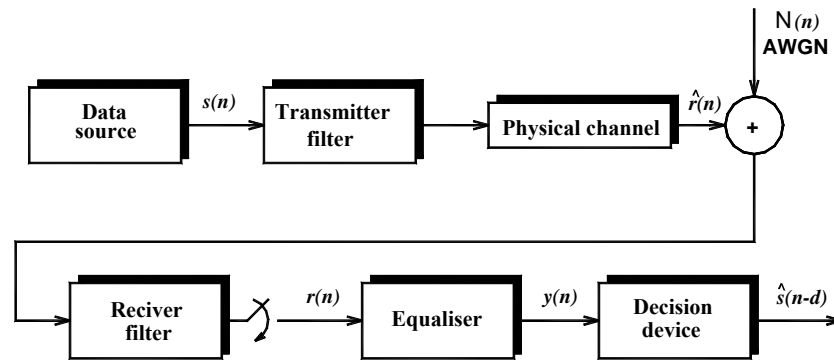
---

This thesis discusses the development of adaptive channel equalisers for communication channels using feedforward and recurrent neural networks. In order to establish the context and motivation for this research work undertaken clearly and coherently, it is necessary to discuss the background of channel equalisation. Some basic equaliser structures and fundamental concepts are presented here, which form the basis for designing equalisers as discussed in the following chapters.

Section 2.1 introduces the basic principle of the channel equaliser including classification. Section 2.2 explains the general finite impulse response (FIR) filter model for ISI channels. The optimal Bayesian equaliser alongwith channel states, decision function and effect of decision feedback is described in Section 2.3. An overview of symbol-by-symbol linear equaliser and decision feedback equaliser are discussed in Section 2.4. The two basic nonlinear equaliser structures (MLPDFE and RNE) are introduced in Section 2.5. Lastly, the concluding remarks are given in Section 2.6.

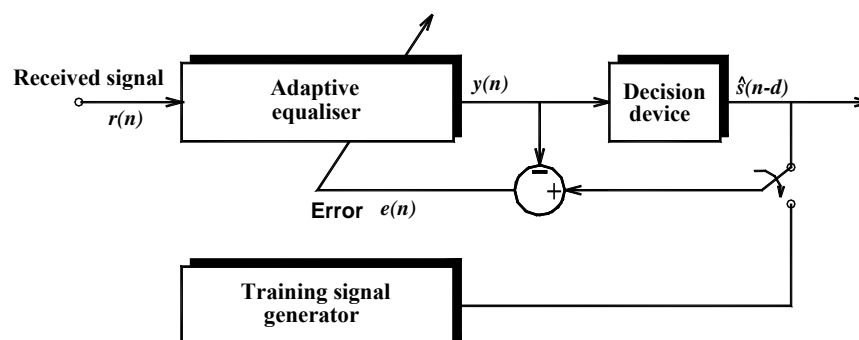
### 2.1 The channel equaliser

In an ideal communication channel, the received information is identical to that of the transmitted signal. However, this is not the case for real communication channels, where signal at the receiver gets distorted in both amplitude and phase. This distortion causes the transmitted symbols to spread and overlap over successive time intervals, which is known as Inter Symbol Interference (ISI). The time dispersion is a serious limitation in attempting to achieve a high transmission rate through a particular band-limited channel. Equalisation techniques are employed at the receiving end to compensate for such distortions and reconstruct the transmitted signal faithfully.



**Figure 2.1: Baseband model of a digital communication system**

The block diagram of baseband model of a digital communication system (DCS) is depicted in Figure 2.1. Communication systems are studied in the base band frequency to avoid the complexity associated with the analysis of various subsystems within a DCS. The data source constitutes the signal generation system that originates the information to be transmitted. The efficient use of the available bandwidth is achieved through the transmitter filter, also called the modulating filter. 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 baseband transmitted signal. This demodulated signal is processed by the receiver filter, also called the receiver demodulating filter, which should be ideally matched to the transmitter filter and channel. The equaliser in the receiver removes the distortion introduced due to the channel impairments. The decision device provides the estimate of the transmitted signal. During transmission of high speed data over a band-limited channel, its frequency response is usually not known with sufficient precision to design an optimum match or matched filter. The equaliser is therefore, should be adaptive in nature to take care of the variations in the characteristics of the channel.



**Figure 2.2: An adaptive equaliser configuration**

The configuration of an adaptive equaliser is shown in Figure 2.2, where an adaptive algorithm is applied to recursively update the equaliser configuration based on the observed channel output. There are two periods of operation of the equaliser,

one is the training period and the other one is the decision directed period. During the training period a known sequence is transmitted and a synchronized version of this signal is generated in the receiver, which is assumed to be the desired response. By comparing this with equaliser output the error signal is computed, based on which the adaptive algorithm works. After the training, the equaliser is switched to the decision directed mode, where the equaliser can update its parameter based on the past detected samples.

### 2.1.1 Adaptive equaliser classification

This section provides adaptive equaliser classification as presented in Figure 2.3. In general the family of adaptive equalisers is either supervised or unsupervised. The channel distortions introduced into the signal can be conveniently removed by sending a training or pilot signal periodically during the transmission of information. A replica of this pilot signal is available at the receiver which uses this to update its parameters during the training period. These types are *supervised equalisers*. However, in certain communication systems like digital television and digital radio, there is hardly any scope for the use of a training signal. In such situations the equaliser needs some form of unsupervised or self-recovery method to update its parameters so as to provide near optimal performance. These are called *blind equalisers*. This thesis investigates *supervised equalisers* in general.

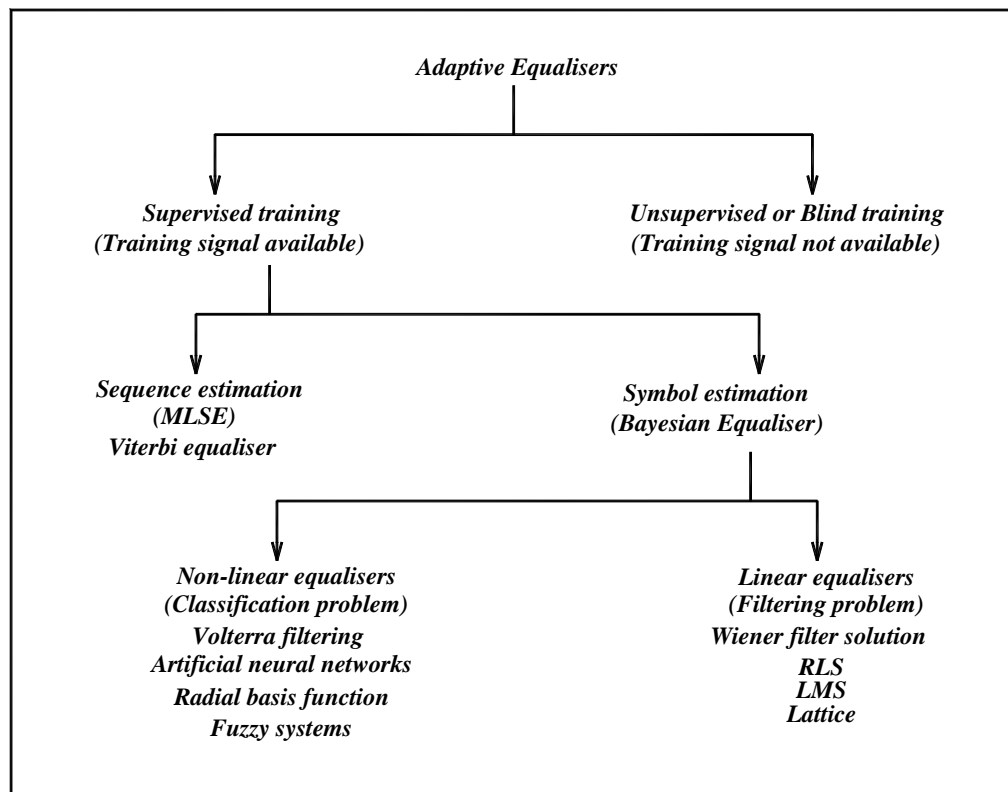


Figure 2.3: Adaptive equaliser classification

The process of supervised equalisation can be achieved in two forms. These are *sequence estimation and symbol-by-symbol estimation*. The former one uses the sequence of past received samples to estimate the transmitted symbol for which it is considered as an infinite memory equaliser and is termed *Maximum-Likelihood-Sequence-Estimation* (MLSE) [40]. The MLSE can be implemented with the Viterbi Algorithm [41]. An infinite memory sequence estimator provides the best bit error ratio (BER) performance for equalisation of time invariant channels but its complexity increases for longer channel length and while tracking a time-varying channel. The *symbol-by-symbol equaliser* on 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 [11]. Hence this optimum finite memory equaliser is also called the *Bayesian equaliser* [29]. The Bayesian equaliser provides the lower performance bound for symbol-by-symbol equaliser in terms of probability of error or BER. The sub-optimal ones are of two types, linear and non-linear. The linear adaptive equaliser is a linear FIR adaptive filter [88,89] trained with an adaptive algorithm like the LMS, RLS or lattice algorithm. During the process of training, these linear equalisers optimise certain performance criterion like minimum mean square error (MMSE). Linear equalisers trained with MMSE criteria provide the *Wiener filter* [88,89] solution. Further, if decision feedback is employed, the linear equaliser provides a decision function received samples along with previously detected samples.

Recent advances in signal processing techniques have provided a rich variety of *nonlinear equalisers*, which are capable of providing the optimum performance. Some of the equalisers developed with these techniques are based on *Volterra filters, MLP, RNN, RBF networks and fuzzy filters*. A review of some of these equalisation techniques can be seen in [18,31,47,52]. The nonlinear equalisers treat equalisation as a pattern classification process instead of inverse filtering adopted in linear ones. 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 *optimal Bayesian equaliser's* performance in terms of BER. The following sections analyse some of the commonly referred linear and nonlinear equalisers in detail.

## 2.2 FIR model of a channel

An ideal physical propagation channel should behave like an ideal low pass filter with fixed amplitude and linear phase characteristics. However in reality all physical channels deviate from this behavior. When signals are transmitted over a



channel, both distortion and additive noise are introduced into it. The transmitted symbols persist beyond the time interval allocated for the transmission and hence subsequent symbols interfere, causing Inter Symbol Interference (ISI). This phenomenon increases as the data rate compression is increased within a fixed bandwidth channel. It is common to model a propagation channel by a digital finite impulse response(FIR) filter shown in Figure 2.4, with taps chosen at the signal's sampling interval and coefficients chosen to accurately model the channel impulse response [37,38].

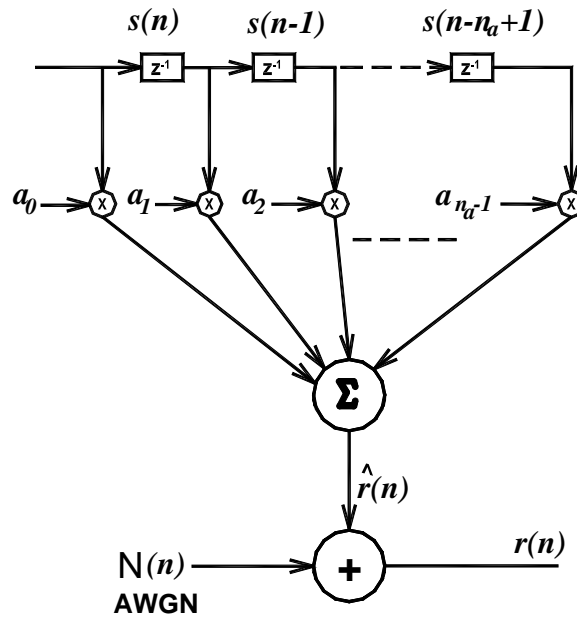


Figure 2.4: FIR model of a channel

The channel impulse response in the  $z$ -domain can be represented by

$$H(z) = \sum_{i=0}^{n_a-1} a_i z^{-i} = a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots \quad (2.1)$$

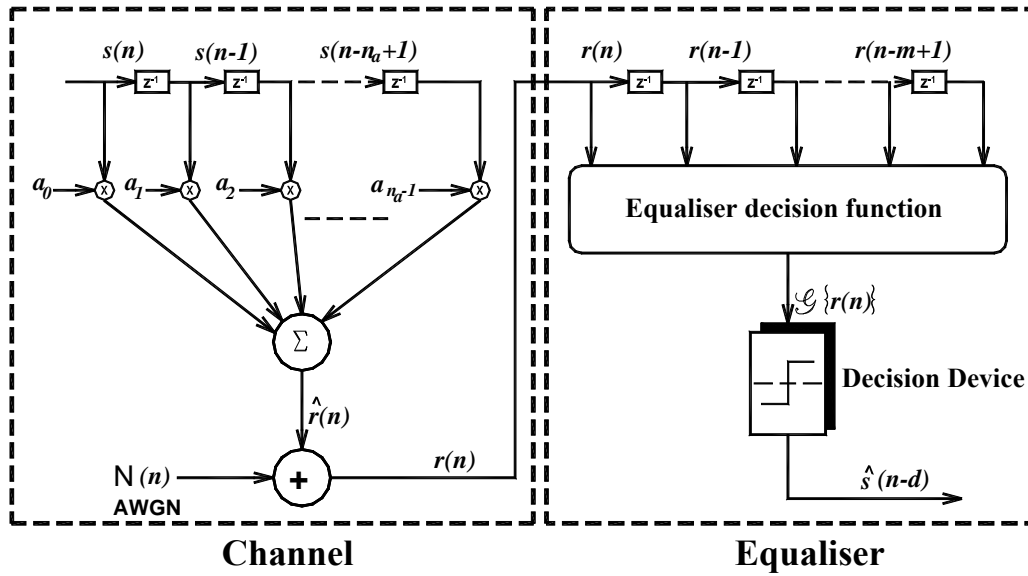
where  $n_a$  represents the length of the channel impulse response (channel order) and the channel provides dispersion up to  $n_a$  samples. The coefficients  $a_i$  represent the strength of the dispersion. The output from FIR modelled channel is described as

$$\begin{aligned} r(n) &= \hat{r}(n) + N(n) \\ &= \sum_{i=0}^{n_a-1} a_i s(n-i) + N(n) \end{aligned} \quad (2.2)$$

where  $r(n)$  is the channel observed output (input to the equaliser). It is given by the sum of the noise free channel output  $\hat{r}(n)$ , which is formed by convolution of the transmitted sequence  $s(n)$  with the channel taps  $a_i$ ,  $0 \leq i \leq n_a-1$  and AWGN  $N(n)$ .

### 2.3 Optimal symbol-by-symbol equaliser: Bayesian equaliser

The optimal symbol-by-symbol equaliser is termed as Bayesian equaliser. To derive the equaliser decision function the discrete time model of the baseband communication system is presented in Figure 2.5.



**Figure 2.5: Discrete time model of a digital communication system**

The equaliser uses an input vector  $\mathbf{r}(n) \in \mathbb{R}^m$ , the  $m$  dimensional space where the term  $m$  is the feedforward order of the equaliser. The equaliser provides a decision function  $G\{\mathbf{r}(n)\}$  based on the input vector which is passed through a decision device to provide the estimate of transmitted signal  $\hat{s}(n-d)$ , 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  $s(n)$  is drawn from an independent identically distributed (i.i.d) sequence comprising of  $\{\pm 1\}$  symbols. The noise source is Additive White Gaussian Noise (AWGN) characterised by zero mean and a variance of  $\sigma_N^2$ .

The equaliser performance is described by the probability of misclassification w.r.t. Signal to Noise Ratio (SNR). The SNR is defined as,

$$\text{SNR} = \frac{\mathbf{E}[\hat{r}(n)^2]}{\mathbf{E}[N(n)^2]}$$

$$= \frac{\sigma_s^2 \sum_{i=0}^{n_a-1} a_i^2}{\sigma_N^2} \quad (2.3)$$

where,  $\mathbf{E}$  is the Expectation operator,  $\sigma_s^2$  represents the transmitted signal power and  $\sum_{i=0}^{n_a-1} a_i^2$  is the channel power. With the assumption that the signal is drawn from an i.i.d. sequence of  $\{\pm 1\}$ , the signal power becomes  $\sigma_s^2 = 1$ . Hence, the SNR can be represented as,

$$\text{SNR} = 10 \log_{10}(1 / \sigma_N^2) \text{ dB} \quad (2.4)$$

The equaliser uses the received signal vector  $\mathbf{r}(n) = [r(n), r(n-1), \dots, r(n-m+1)]^T \in \mathbb{R}^m$  to estimate the delayed transmitted symbol  $s(n-d)$ . The decision device at the equaliser output uses a  $\text{sgn}(x)$  function given by

$$\text{sgn}(x) = \begin{cases} +1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases} \quad (2.5)$$

Hence, the estimate of the transmitted signal given by the equaliser is

$$\hat{s}(n-d) = \text{sgn}(\mathbf{G}\{\mathbf{r}(n)\}) = \begin{cases} +1 & \text{if } \mathbf{G}\{\mathbf{r}(n)\} \geq 0 \\ -1 & \text{if } \mathbf{G}\{\mathbf{r}(n)\} < 0 \end{cases} \quad (2.6)$$

The performance of an equaliser can be evaluated as follows. For bit error rate (BER) calculation if the equaliser is tested with statistically independent random data sequence of  $10^7$  channel samples then an error value  $e_i$  is generated in the following manner.

$$e_i = \begin{cases} 0 & \text{if } \hat{s}(n-d) = s(n-d) \\ 1 & \text{if } \hat{s}(n-d) \neq s(n-d) \end{cases} \quad (2.7)$$

Then the BER is evaluated in decimal logarithm as

$$\text{BER} = \log_{10}\left(\sum_{i=1}^{10^7} e_i / 10^7\right) \quad (2.8)$$

The process of equalisation discussed here can be viewed as a classification process in which the equaliser partitions the input space  $\mathbf{r}(n) \in \mathbb{R}^m$ , into two regions corresponding to each of the transmitted sequence  $+1$  /  $-1$  [14,29,90]. The loci of points which separate these two regions is termed as the *decision boundary*. If the received signal vector is perturbed sufficiently to cross the decision boundary due to the presence of AWGN, misclassifications result. To minimise the probability of misclassifications for a given received signal vector  $\mathbf{r}(n)$ , the transmitted symbol should be estimated based on  $s(n) \in \{\pm 1\}$  having a maximum *a-posteriori* probability

(MAP) [11,91]. The partition which provides the minimum probability of misclassification is termed as optimal (Bayesian) decision boundary.

### 2.3.1 Channel states

The concept of channel states is introduced here. The equaliser input vector has been defined as  $\mathbf{r}(n) = [r(n), r(n-1), \dots, r(n-m+1)]^T \in \mathbb{R}^m$ , the  $m$  dimensional observation space. The vector  $\hat{\mathbf{r}}(n)$  is the noise-free received signal vector  $\hat{\mathbf{r}}(n) = [\hat{r}(n), \hat{r}(n-1), \dots, \hat{r}(n-m+1)]^T$ . Each of these possible noise-free received signal vectors constitutes a channel state. The channel states are determined by the transmitted symbol vector

$$\mathbf{s}(n) = [s(n), s(n-1), \dots, s(n-m-n_a+2)]^T \in \mathbb{R}^{m+n_a-1} \quad (2.9)$$

Here  $\hat{\mathbf{r}}(n)$  can be represented as  $\hat{\mathbf{r}}(n) = \mathbf{H}[\mathbf{s}(n)]$ , where matrix  $\mathbf{H} \in \mathbb{R}^{m \times (m+n_a-1)}$  is the channel matrix which can be expressed as

$$\mathbf{H} = \begin{bmatrix} a_0 & a_1 & L & a_{n_a-1} & 0 & L & 0 & L & 0 \\ 0 & a_0 & L & a_{n_a-2} & a_{n_a-1} & L & 0 & L & 0 \\ M & M & M & O & O & O & M & M & M \\ 0 & 0 & L & L & L & L & a_0 & L & a_{n_a-1} \end{bmatrix} \quad (2.10)$$

Since the channel input sequence  $\mathbf{s}(n)$  has  $n_s = 2^{m+n_a-1}$  combinations, the noise-free channel output vector  $\hat{\mathbf{r}}(n)$  has  $n_s$  states, which are constructed with  $n_s$  sequences of  $\mathbf{s}(n)$ . The set of these states, denoted as  $\mathbf{R}_{m,d}$  can be partitioned into two subsets according to the value of transmitted symbols  $s(n-d)$ , i.e.

$$\mathbf{R}_{m,d} = \bigcup_{1 \leq i \leq 2} \mathbf{R}_{m,d}^{(i)} \quad (2.11)$$

$$\text{where, } \mathbf{R}_{m,d}^{(i)} = \{\hat{\mathbf{r}}(n) \mid s(n-d) = s_i\}, \quad 1 \leq i \leq 2 \quad (2.12)$$

Here, the positive channel states for  $s_i = +1$  are denoted as  $\mathbf{R}_{m,d}^1$  where as the negative channel states are  $\mathbf{R}_{m,d}^2$  for  $s_i = -1$ . The number of states in each  $\mathbf{R}_{m,d}^{(i)}$  for  $1 \leq i \leq 2$  is given by

$$n_s^{(i)} = n_s / 2 \quad (2.13)$$

#### **Example:**

An example is considered to show the channel states. The channel considered here is a non-minimum phase channel represented by its  $z$ -transform  $H_4(z) = 0.5 + 1.0 z^{-1}$ . The equaliser feedforward order considered here is  $m = 2$ . The number of noise-free

channel states are  $n_s=8$  ( $2^{m+n_a-1}$ ) and are presented in Table 2.1. Their location with reference to  $\hat{\mathbf{r}}(n)$  is decided taking the scalar components  $[\hat{\mathbf{r}}(n), \hat{\mathbf{r}}(n-1)]^T$ .

No.	Transmitted symbol sequence			$\hat{\mathbf{r}}(n)$	
	$s(n)$	$s(n-1)$	$s(n-2)$	$\hat{\mathbf{r}}(n)$	$\hat{\mathbf{r}}(n-1)$
1	1	1	1	1.5	1.5
2	1	1	-1	1.5	-0.5
3	1	-1	1	-0.5	0.5
4	1	-1	-1	-0.5	-1.5
5	-1	1	1	0.5	1.5
6	-1	1	-1	0.5	-0.5
7	-1	-1	1	-1.5	0.5
8	-1	-1	-1	-1.5	-1.5

**Table 2.1: Channel states calculation for channel  $H_4(z) = 0.5 + 1.0z^{-1}$  with  $m = 2$**

### 2.3.2. Bayesian equaliser decision function

The presence of additive noise (AWGN) makes the channel observation vector  $\mathbf{r}(n)$  a random process, having a conditional Gaussian density function centered at each noise free received vector  $\hat{\mathbf{r}}(n)$ . Hence, each channel state  $\hat{\mathbf{r}}(n)$  is a conditional mean vector of  $\mathbf{r}(n)$  given  $s(n)$ . Bayes decision theory [29] provides the optimal solution to the general decision problem and is applicable here. The two Bayesian decision variables for classifying into two regions corresponding to each of the transmitted sequence +1/-1 are computed as

$$P_i(n) = \sum_{j=1}^{n_s^{(i)}} p_j^{(i)} p_{\mathbf{N}}(\mathbf{r}(n) - \mathbf{r}_j^{(i)}), \quad 1 \leq i \leq 2 \quad (2.14)$$

where  $\mathbf{r}_j^{(i)} \in \mathbf{R}_{m,d}^{(i)}$  defined in Equation (2.12),  $p_j^{(i)}$  are the *a-priori* probabilities of  $\mathbf{r}_j^{(i)}$  and  $p_{\mathbf{N}}(\cdot)$  is the probability density function (pdf) of  $\mathbf{N}(n)$ . Each  $P_i(n)$  is the conditional pdf of  $\mathbf{r}(n)$  given  $s(n-d) = s_i$ . Since the entire channel states can be assumed to be equiprobable, all the  $p_j^{(i)}$  are equal and the noise distribution is assumed to be Gaussian, Equation (2.14) can be explicitly expressed as

$$P_i(n) = \sum_{j=1}^{n_s^{(i)}} \chi \cdot \exp\left(-\|\mathbf{r}(n) - \mathbf{r}_j^{(i)}\|^2 / 2\sigma_{\mathbf{N}}^2\right), \quad 1 \leq i \leq 2 \quad (2.15)$$

where  $\chi$  is  $p_j^{(i)}(2\pi\sigma_N^2)^{-m/2}$  multiplied by an arbitrary positive constant and  $\|\cdot\|$  constitutes the Euclidean distance.

The minimum-error-probability decision is defined by

$$\hat{s}(n-d) = s_i^* \quad \text{if } P_i^*(n) = \max\{P_i(n), \quad 1 \leq i \leq 2\} \quad (2.16)$$

The Bayesian decision procedure effectively partitions the  $m$ -dimensional observation space into two decision regions. Equation (2.16) can be rearranged as

$$\hat{s}(n-d) = \text{sgn}(\mathbf{G}\{r(n)\}) = \text{sgn}\{P_2(n) - P_1(n)\} \quad (2.17)$$

where  $\text{sgn}(\cdot)$  is the signum function and  $\mathbf{G}(\cdot)$  can be referred to as the *Bayesian decision function*. The decision function is nonlinear and is completely specified in terms of the channel states and the noise characteristics. Further, the set

$$\{r(n) | \mathbf{G}\{r(n)\} = 0\} \quad (2.18)$$

defines the optimal decision boundary, which is a hypersurface in the observation space. It also means that the optimal decision boundary exists when the two class probabilities are equal for a 2-PAM input signalling scheme.

### 2.3.3 Bayesian equaliser with decision feedback

Inclusion of decision feedback further reduces the number of channel states and it also increases the minimum distance between the two classes of channel states. The significant improvement in performance offered by decision feedback can be explained by the following mathematical reasoning.

The feedback order  $n_b$  of the equaliser [29] employed to mitigate ISI from previously detected symbols is given by

$$n_b = n_a + m - 2 - d \quad (2.19)$$

The feedback vector  $\hat{s}_f(n-d)$  has  $n_f = 2^{n_b}$  states. These feedback states are denoted as  $s_{f,j}$ ,  $1 \leq j \leq n_f$ . A subset of the channel states  $\mathbf{R}_{m,d}^{(i)}$  defined in Equation (2.12) can further be partitioned into  $n_f$  subsets according to the feedback state

$$\mathbf{R}_{m,d}^{(i)} = \bigcup_{1 \leq j \leq n_f} \mathbf{R}_{m,d,j}^{(i)} \quad (2.20)$$

with

$$\mathbf{R}_{m,d,j}^{(i)} = \{\hat{r}(n) | s(n-d) = s_i \text{ I } \hat{s}_f(n-d) = s_{f,j}\}, \quad 1 \leq j \leq n_f \quad (2.21)$$

The number of states in each  $\mathbf{R}_{m,d,j}^{(i)}$  is  $n_{s,j}^{(i)} = n_s^{(i)} / n_f$ . Under the assumption of correct feedback vector  $\hat{\mathbf{s}}_f(n-d)$ , the two Bayesian decision variables given  $\hat{\mathbf{s}}_f(n-d) = s_{f,j}$  are

$$P_i(n | \hat{\mathbf{s}}_f(n-d) = s_{f,j}) = \sum_{l=1}^{n_{s,j}^{(i)}} \chi_l \cdot \exp(-\|\mathbf{r}(n) - \mathbf{r}_l^{(i)}\|^2 / 2\sigma_N^2), \quad 1 \leq i \leq 2 \quad (2.22)$$

where  $\mathbf{r}_l^{(i)} \in \mathbf{R}_{m,d,j}^{(i)}$

The conditional Bayesian decision is defined as

$$\hat{\mathbf{s}}(n-d) = s_i^* \quad \text{if} \quad P_i^*(n) = \max\{P_i(n | \hat{\mathbf{s}}_f(n-d) = s_{f,j}), \quad 1 \leq i \leq 2\} \quad (2.23)$$

The feedback vector is used to reduce the number of channel states needed in decision making. Without feedback, all the  $n_s$  channel states are required to compute the two decision variables. As a result of feedback, only a fractional number of these states  $n_s / n_f = 2^{d+1}$  are needed to compute the decision variables. Hence, there is a reduction in computational complexity due to inclusion of decision feedback technique.

## 2.4 Symbol-by-symbol linear equaliser

The structure of a linear equaliser is presented in Figure 2.6.

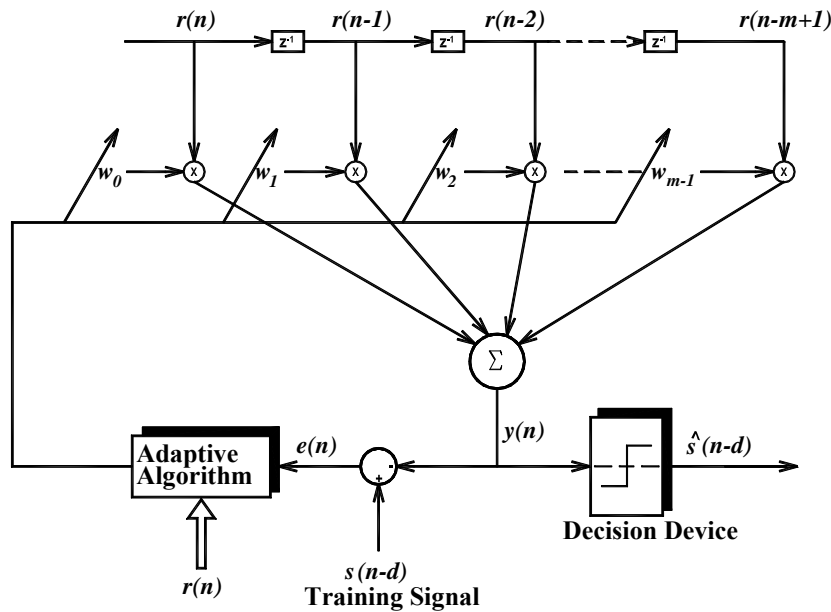


Figure 2.6: Structure of a linear equaliser

The equaliser consists of a tapped delay line (TDL) which receives the receiver sampled input vector  $\mathbf{r}(n) = [r(n), r(n-1), \dots, r(n-m+1)]^T$  and provides an output  $y(n)$  which is the convolution sum of input vector  $\mathbf{r}(n)$  with the weight vector  $\mathbf{w}$ . The output is computed once per symbol and can be represented as

$$y(n) = \sum_{i=0}^{m-1} w_i r(n-i) \quad (2.24)$$

The weight vector  $\mathbf{w}$  optimises one of the performance criteria like zero forcing (ZF) or MMSE criteria [37,38]. The decision device present at the output of the equaliser, provides the transmitted signal constellation. The MMSE criteria provide equaliser tap coefficients to minimise the mean square error at the equaliser output before the decision device. This condition can be represented as

$$J(n) = \mathbf{E} |e(n)|^2 \quad (2.25)$$

$$e(n) = s(n-d) - y(n) \quad (2.26)$$

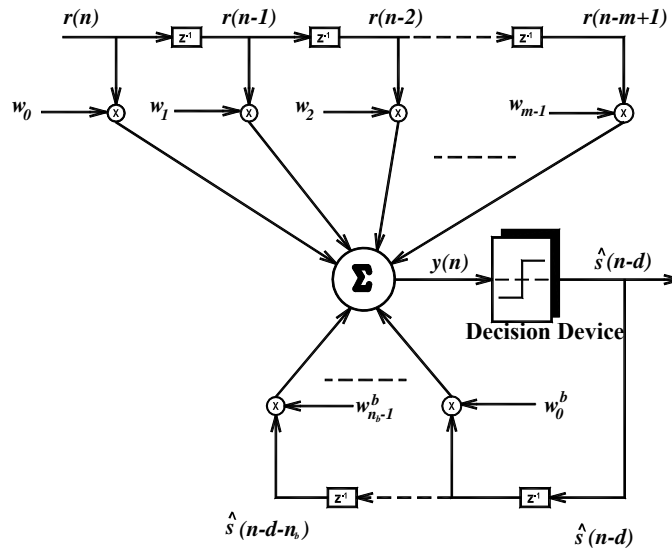
where  $e(n)$  is the error associated with the equaliser output  $y(n)$ . Adaptive algorithms like LMS and RLS can be used to recursively update the equaliser weights during the training period. The convergence properties and the performance of linear equalisation have been well documented in the literature [37,38]

Linear equalisers with higher weight dimensions can improve accuracy. However, higher dimensions leave the equaliser susceptible to noisy samples and such structures will take a long time to converge [46]. Thus LTE suffers performance degradation when the communication channel causes severe ISI distortion. When the channel has a deep spectral null in its bandwidth, linear equalisation performs poorly since it places a high gain at the frequency of the null, thus enhancing the additive noise. Under such conditions decision feedback equalisation [46] can be employed to overcome these limitations.

#### 2.4.1 Decision feedback equaliser

A basic structure of Decision Feedback Equaliser is presented in Figure 2.7.





**Figure 2.7: Structure of a linear decision feedback equaliser**

This equaliser is characterised by its feedforward order  $m$  and the feedback order  $n_b$ . The equaliser uses  $m$  feedforward samples and  $n_b$  feedback samples from the previously detected samples.

The signal vector associated with feedback weight vector  $\mathbf{w}_b = [w_0^b, w_1^b, \dots, w_{n_b-1}^b]^T$  is given by  $\hat{\mathbf{s}}(n) = [\hat{s}(n-d-1), \hat{s}(n-d-2), \dots, \hat{s}(n-d-n_b)]^T$ . Considering that the DFE is updated with a recursive LMS algorithm, the feedforward and feedback filter weights can be jointly adapted using a common error signal  $e(n)$ . The feedback section in the equaliser helps to remove the ISI contribution from the estimated symbols and hence the DFE provides better performance than a conventional feedforward linear equaliser under severe ISI distortion.

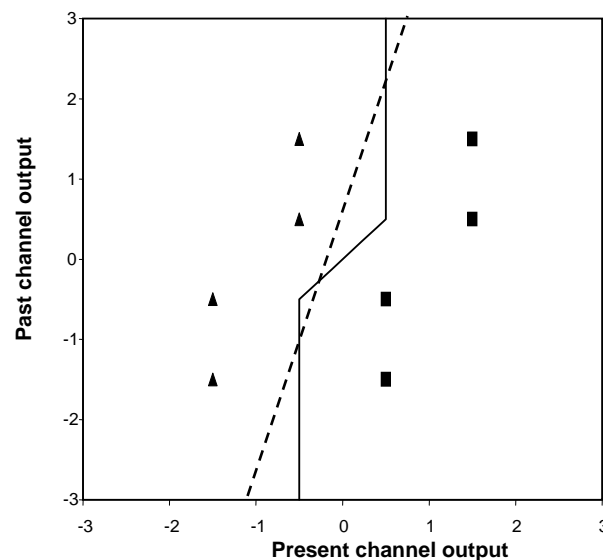
The linear equaliser can successfully reconstruct the transmitted sequence only if the channel is minimum phase. The non-minimum phase channels can be equalized by linear equalisers only if some delay  $d$  is introduced. This is illustrated by examples of minimum phase and non-minimum phase channels [12,13,14]. Thus, in order to design structures to equalise non-minimum phase channels, more sophisticated architectures are needed.

**Example:**

A minimum phase channel is considered as an example whose transfer function is given by

$$H_1(z) = 1 + 0.5z^{-1} \quad (2.27)$$

By viewing the concept of the linear equaliser as a classification process, the observation space is formed by using two successive samples of the channel output. The linear equaliser partitions the input observation space by a simple hyperplane as shown in Figure 2.8. It gives a complete satisfactory solution in the absence of noise. However, if noise is added then this linear division is no longer optimum as it is observed that perfect observation vectors (noise-free channel states) lie at unequal distances from the hyper plane. Further, if we apply MAP criterion, it is observed that the optimal decision boundary is highly non-linear as shown in Figure 2.8 and deviates markedly from any decision boundary which can be formed by a linear equaliser. Therefore solution offered by any linear equaliser is inherently sub-optimal and this important drawback motivated the development of several nonlinear architectures capable of realising the optimal decision boundaries.



**Figure 2.8:** Optimal decision boundary and noise-free channel states corresponding to a  $\{+1\}$  transmission given by symbols (■) and points corresponding to a  $\{-1\}$  transmission given by symbols (▲) for the channel with impulse response  $H_1(z)$ ,  $m=2$ ,  $d=0$ .

## 2.5 Symbol-by-symbol adaptive nonlinear equalisers

All non-linear equalisers treat equalisation as a non-linear 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 discussed in the following section are based on the Artificial Neural Network (ANN). The Feedforward Neural Network (FNN) and the Recurrent Neural Network (RNN) are explained in detail as the proposed equaliser structures are built on these frame works. Some of the other nonlinear equalisers are the RBF network, [51], recurrent RBF [92],

Volterra filters [93], the functional link networks [66], adaptive fuzzy filters [94], etc. Amongst the FNN structures, the most commonly and widely used is the multilayer perceptron (MLP). The MLP architecture consists of a number of processing neurons organised in layers and is capable of performing complex, nonlinear mappings between the input and output. Gibson et al. [13] showed that such an equaliser can provide the non-linear decision boundary associated with the MAP equaliser. Further, its performance can be enhanced by incorporating decision feedback approach. It is shown that the MLP decision feedback equaliser (DFE) trained in a supervised manner using the Back Propagation (BP) algorithm gives a significant improvement in performance as documented in the literature [26].

### 2.5.1 A multilayer perceptron decision feedback equaliser : MLPDFE

In equaliser application the input to the multilayer perceptron structure is presented through a set of tapped delay lines and the output layers has a single neuron. A MLPDFE structure [26] is shown in Figure 2.9, consisting of a feedforward filter and a feedback filter. The input to the feedforward filter is the sequence of noisy received signal samples  $r(n)$ . The input to the feedback filter is the estimated signal  $\hat{s}(n-d)$  from the decision device, where  $d$  is the delay introduced. The equaliser structure can be trained in a supervised manner using the BP algorithm [06].

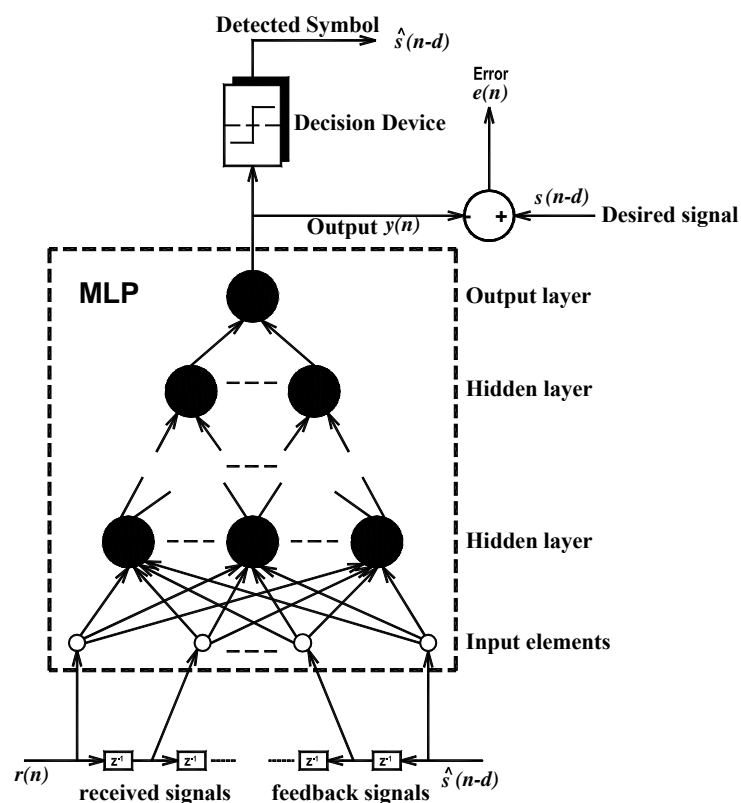


Figure 2.9: Multilayer perceptron decision feedback equaliser

At time index  $n$ , the  $m \times 1$  received signal vector  $\mathbf{r}(n)=[r(n), r(n-1), \dots, r(n-m+1)]$  and  $n_b \times 1$  decision signal vector  $[\hat{s}(n-d-1), \hat{s}(n-d-2), \dots, \hat{s}(n-d-n_b)]$  are fed into the decision feedback equaliser. The signal at the input layer of the decision feedback equaliser can be represented by a  $(m + n_b) \times 1$  vector as

$$\mathbf{x}(n) = [r(n), r(n-1), \dots, r(n-m+1); \hat{s}(n-d-1), \hat{s}(n-d-2), \dots, \hat{s}(n-d-n_b)]^T \quad (2.28)$$

The final estimated output signal  $y(n)$  at time index  $n$ , can be calculated as follows

$$y(n) = F_o \left( \sum_{k=1}^{N_2} w_{ko}^{(3)}(n) F_k \left( \sum_{j=1}^{N_1} w_{jk}^{(2)}(n) F_j \left( \sum_{i=0}^{m-1} w_{ij}^{(1)}(n) r(n-i) + \sum_{p=1}^{n_b} w_{pj}^{b(1)}(n) \hat{s}(n-d-p) + th_j^{(1)}(n) \right) + th_k^{(2)}(n) \right) + th_o^{(3)} \right) \quad (2.29)$$

where all  $F$ 's denote sigmoidal activation functions in the neurons.  $N_1$  and  $N_2$  are the number of neurons in the two hidden layers respectively. The output of the nonlinear detector can be defined as

$$\hat{s}(n-d) = \begin{cases} 1 & \text{if } y(n) \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (2.30)$$

The  $w$  (weights) and  $th$  (threshold levels) in Equation (2.29) are values specified by the training algorithm, so that after training is completed the equaliser will self-adapt to the changes in the channel characteristics occurring during transmission (decision directed mode). The BP learning algorithm [06] for training multilayer perceptron network is discussed in detail in Appendix A.

### 2.5.2 Recurrent neural network equaliser (RNE)

Recurrent Neural Networks (RNNs) are highly nonlinear. The network incorporates feedback mechanism of its own and as a result their architectures become inherently dynamic. Actually RNNs model non-linear IIR filters and can accurately realise the inverse of finite memory channels using relatively small number of neurons. The use of Recurrent Neural Network Equalisers (RNE) for adaptive equalisation of linear and non-linear channels has been proposed in [32]. Kechriotis et al. showed that simple RNE structures having small sizes can be successfully applied to equalisation problems. The block diagram of a communication system that employs a RNN based adaptive equaliser is shown in Figure 2.10.

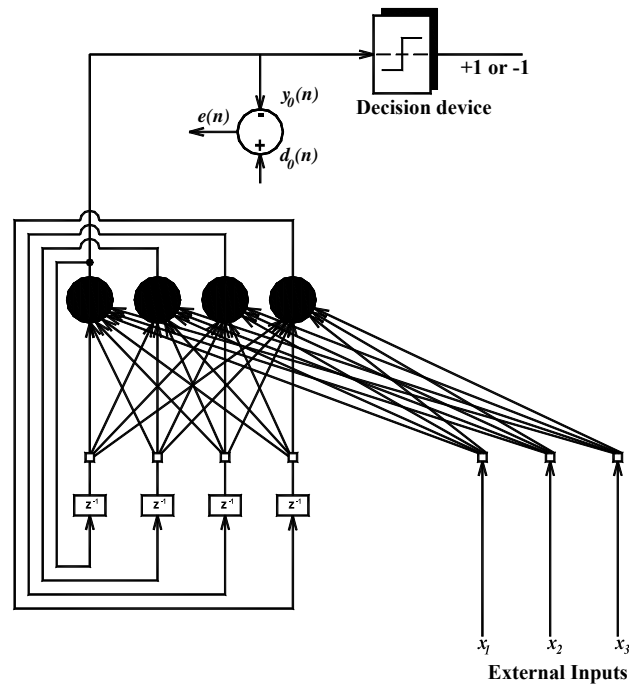


Figure 2.10: Recurrent neural network equaliser

The most widely used Real-Time-Recurrent-Learning (RTRL) algorithm, proposed by Williams and Zipser [55], is used to update the weights of the RNN by computing the gradient of the squared error with respect to the weights of the equaliser. The details of the RTRL algorithm are discussed in detail in Appendix B.

## 2.6 Conclusion

In this chapter the background of channel equalisation along with the optimal Bayesian Equaliser structure have been discussed elaborately. The symbol-by-symbol linear equaliser has been explained along with its drawbacks. Its performance is inherently suboptimal as the optimal decision boundary is nonlinear in nature. It is concluded that by incorporating a degree of nonlinearity in the design of an equaliser, it is possible to produce a structure which can achieve near optimal performance. All non-linear equalisers treat equalisation as a pattern classification problem and from this angle neural networks offer attractive solutions as they are capable of providing nonlinear decision boundaries. Feedforward neural network and recurrent neural network have been the main focus of this chapter, as all the proposed neural equaliser structures and their respective training algorithms have been designed on these platforms. The effectiveness of proposed structures in terms of BER performance in comparison with conventional ones has been validated in the subsequent chapters.

---

## CHAPTER 3

# Factors Influencing Equaliser's Performance and Parameter Selection

---

The symbol-by-symbol detection approach to equalisation applies the channel output samples to a decision classifier that separates the symbols into their respective classes. The main objective is the separation of the received symbols in the output signal space with the minimum probability of misclassification. The optimal solution for the symbol-by-symbol equaliser structure has already been discussed in the previous chapter using Bayes decision theory, which views equalisation exclusively as a two state classification problem assuming 2-PAM signalling scheme [14,29,90]. A detailed study on various factors influencing the BER performance of the optimal Bayesian equaliser has been undertaken in this research work. It is observed here that the Bayesian decision function is greatly dependent on decision delay ' $d$ ', equaliser feedforward order ' $m$ ' and additive noise level [29]. Also from the design point of view, the proper selection of ' $m$ ' and ' $d$ ' plays a significant role in the optimal BER performance, which is evident from in-depth computer simulation study carried out for various real channel models (both minimum phase and non-minimum phase).

Further, the performance of a symbol-by-symbol Bayesian equaliser improves after incorporating decision feedback technique. Simulation studies carried out in the present research work have also proved that if the noise-free channel states belonging to different classes are either coincident or very close to the optimal decision boundary, then symbol-by-symbol Bayesian equalisers with decision feedback are capable of classifying the transmitted symbols correctly, while equalisers without feedback fail to achieve any acceptable performance even if the feedforward order is increased to a higher value. This advantage offered by DFE structure has necessitated the selection of proper values of structure parameters like feedforward order ' $m$ ', feedback order ' $n_b$ ' and decision delay ' $d$ ' for optimising the BER performance of an equaliser[29,86,95]. Considering this aspect to be of great importance, certain

empirical relations based on fundamental concepts have been developed and logically interpreted in this present work for selection of these critical design parameters by directly examining the channel coefficients.

In the present simulation work, the bit error rate (BER) of optimal Bayesian equaliser is computed based on a realisation of  $10^5$  channel samples and averaged over 20 independent realisations. Each realisation has a different random input sequence (binary message). Equalisation of various linear and nonlinear channels with 2-PAM signalling input only is considered for all comparative performance analysis. The performance measure is presented through the plot of scatter diagrams, decision regions and BER curves.

### **3.1 Factors influencing the performance of optimal symbol-by-symbol (Bayesian) equaliser**

The Bayesian decision boundary is affected by the decision delay parameter and the noise statistics, which are to be discussed in the subsequent sections. Further, the role of decision feedback in improving the BER performance is studied. Decision regions for equaliser's feedforward order ' $m$ ' = 2 only has been shown in order to provide some geometric insight into the equalisation process in a two-dimensional observation space as the graphic display is difficult to realise in higher dimensions.

#### **3.1.1 Additive noise level**

The performance of an equaliser is greatly dependent on the additive noise level irrespective of the influence of the ISI. However, it has been observed that if the effect of ISI is completely ignored then at certain additive noise level, the observed samples belonging to one class (either class 1 or 2 i.e. for +1 or -1 transmitted symbols respectively) in a two state classification, can invade the decision boundary and migrate to the territory of the other class causing misclassification. It particularly occurs at severe noise conditions and hence under no circumstances this effect can be mitigated. It clearly signifies that for a given feedforward order of the equaliser, compensation for missed bits is not satisfactory at higher level of noise. This phenomenon has been explained in detail.

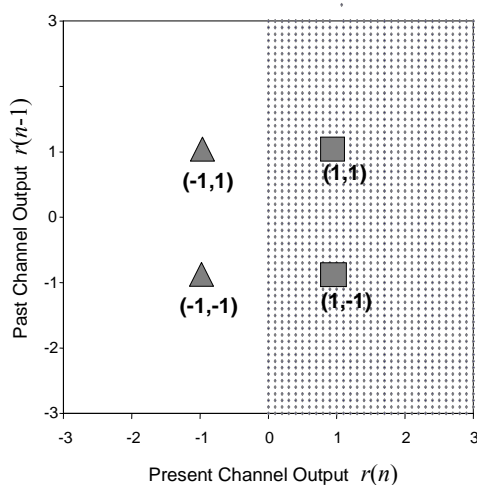
In the beginning the effect of additive noise on the BER performance while equalising an ideal channel (without ISI effect) is analysed. Such a channel is characterised by unit impulse response and its transfer function is defined as

$$H(z)=1 \tag{3.1}$$

Figure 3.1a shows the noise-free channel states and the corresponding optimal decision region plots for this channel. Here the decision delay ' $d$ ' has been chosen as  $d = 0$ . The optimal decision regions also provide an insight into the margin between the two classes (1 and 2) in the classification, where the margin is defined as the minimum distance of the channel states from the decision boundary.

Figures 3.1b-d represent the two-dimensional scatter plots of the observed channel output vectors belonging to the two classes for various additive noise levels (i.e., SNR=6dB, SNR=12dB, and SNR=20dB). The observed channel output points will be distributed as a sum of 2-dimensional Gaussian distributions with spread determined by the variance of noise and centered on the noise-free channel states.

It is clearly evident that at low noise level (SNR=20dB) the radius of spread of the observation clusters is squeezed; hence no received sample points can migrate from one class to another. But depending on the noise severity, the probability of those sample points of one class crossing the optimal decision boundary and entering to the wrong class increases, causing more and more misclassification. This interesting phenomenon is corroborated in the BER performance characteristic given in Figure 3.1e, where at point 'A' (SNR = 6dB), the BER level is more in comparison to point 'B' (SNR = 12dB).



**Figure 3.1:** (a) Noise-free channel states and optimal decision regions{-ve symbols represented by triangles and +ve symbols represented by rectangles}(Ideal Channel:  $H(z) = 1$ )



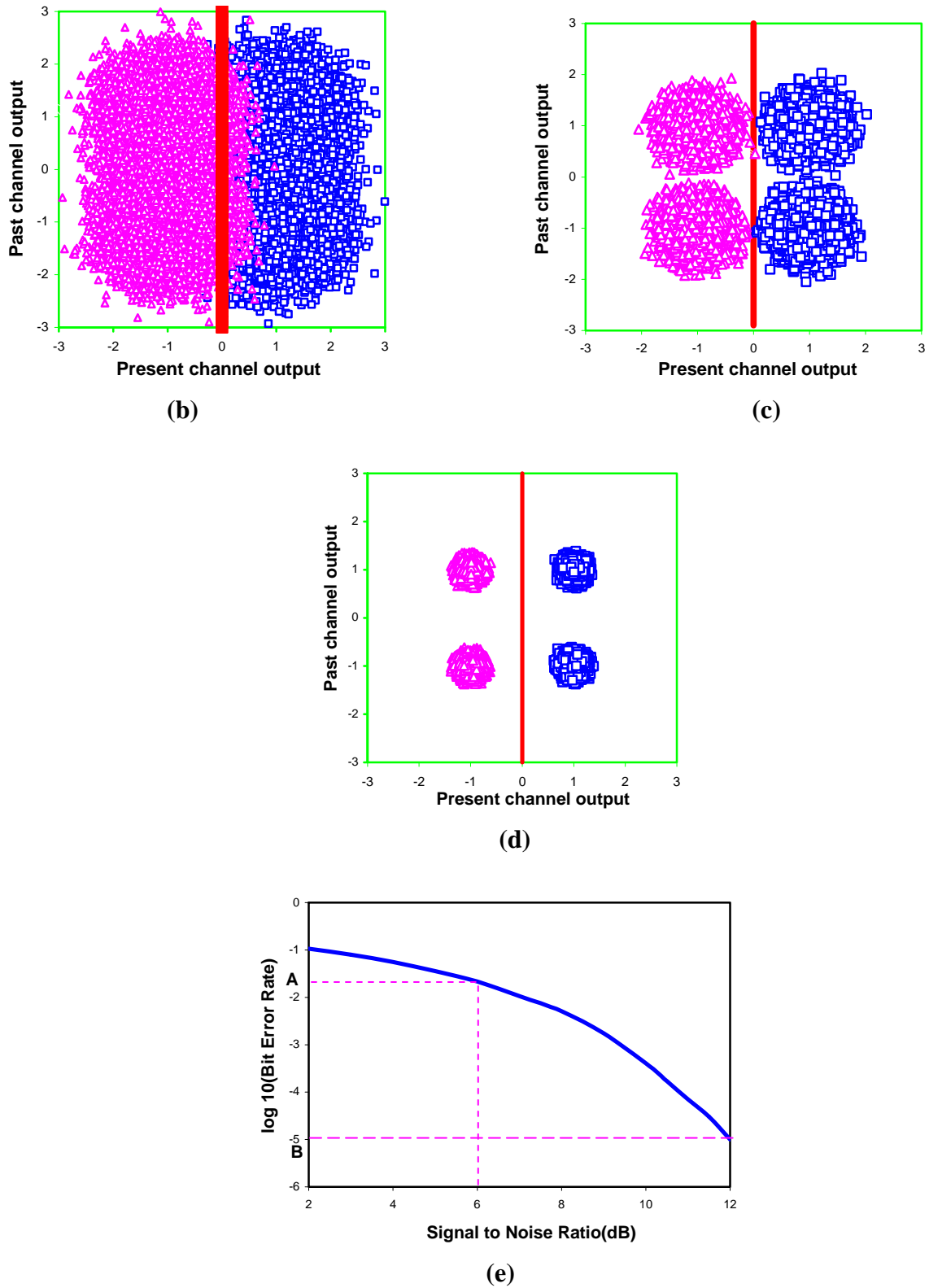


Figure 3.1 : Optimal decision boundary and scatter plots (b) SNR=6dB, (c) SNR=12dB, (d) SNR=20dB {-ve symbols represented by triangles (pink) and +ve symbols represented by squares(blue)} and (e) Optimal Bayesian performance curve ( Channel :  $H(z)=1$ )

The influence of ISI in a channel degrades the equaliser's BER performance, as evident from the comparative analysis given in Figure 3.2. The various channel models used here are defined as

$$H_1(z)=1+0.5z^{-1} \tag{3.2}$$

$$H_2(z)=1+0.7z^{-1} \tag{3.3}$$

$$H_3(z)=1+0.95z^{-1} \tag{3.4}$$

Further, the importance of additive noise level in a channel with ISI effect is discussed. Figures 3.3a-c illustrate in a qualitative manner the 2-dimensional scatter plots and optimal decision boundary for different additive noise levels (i.e., SNR=6dB, SNR=10dB, and SNR=20dB) for a two-tap channel model  $H_1(z)$ . It is inferred from these plots that the margin between the two classes reduces depending upon the severity of noise level and also the optimal decision boundary becomes nonlinear. This effect of additive noise level remains unchanged irrespective of the presence of ISI.

If the additive noise level is high (SNR < 7dB), the influence of ISI on BER performance is minimal as evident from Figure 3.2. This is because the noise amplitude is already strong enough to push the received sample points of a given class in such a way that these sample points can easily cross the optimal decision boundary and enter into the territory of the other class. And thus the ISI factor can not degrade the BER performance further, as misclassifications of received sample points have already been occurred. However, at additive noise level i.e., SNR=12dB, the effect of ISI is predominant over additive noise level as seen in Figure 3.2. This happens because the separation distance (margin) from the decision boundary reduces significantly in comparison with that without ISI due to the nonlinearity introduced by the channel characteristics as observed in Figure 3.3c. So the chances of misclassification of the received sample points are more due to crossover of the decision boundary by them.

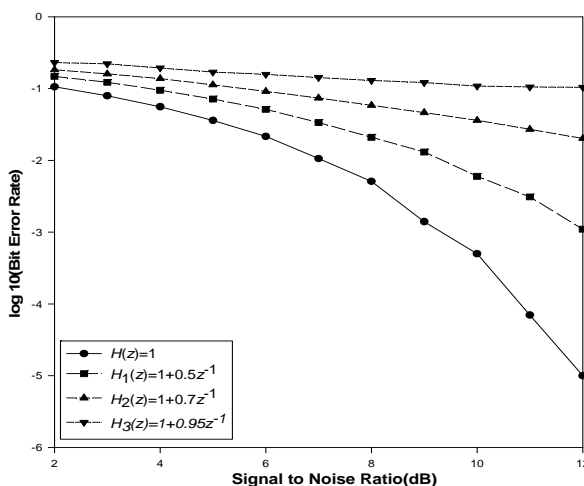


Figure 3.2: Effect of ISI on optimal Bayesian BER performance characteristics

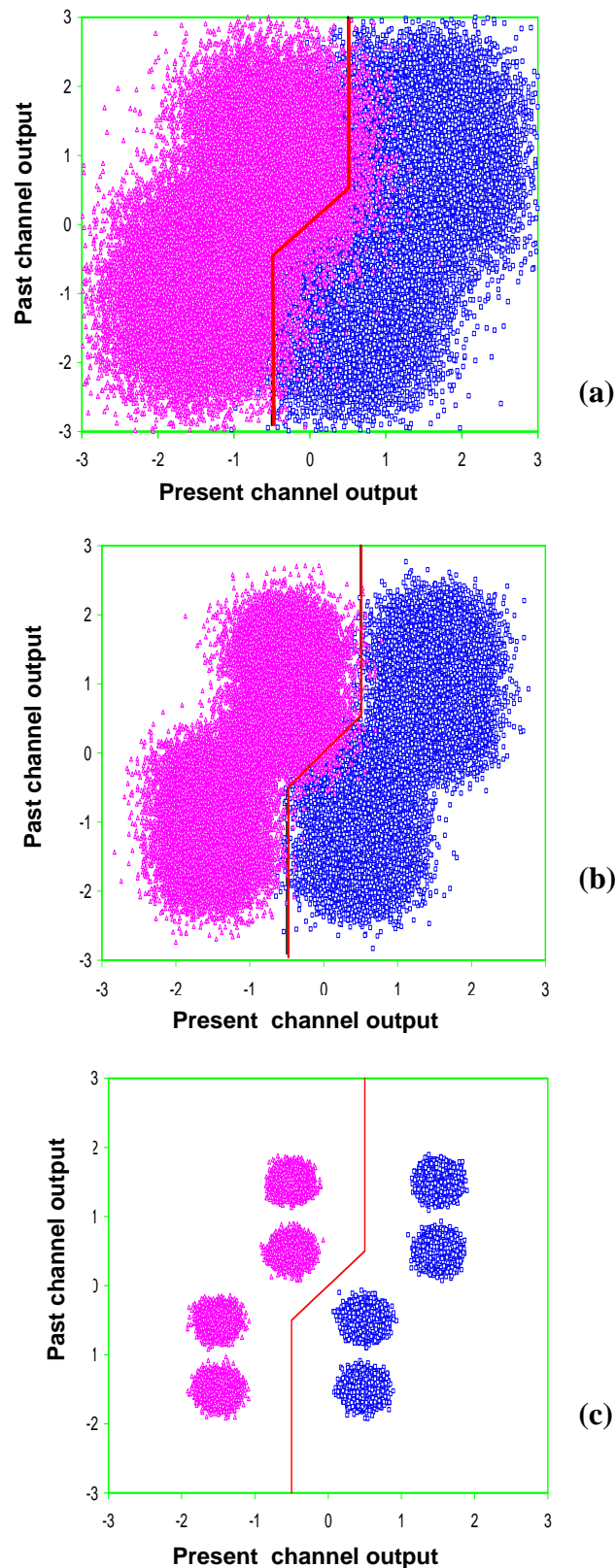
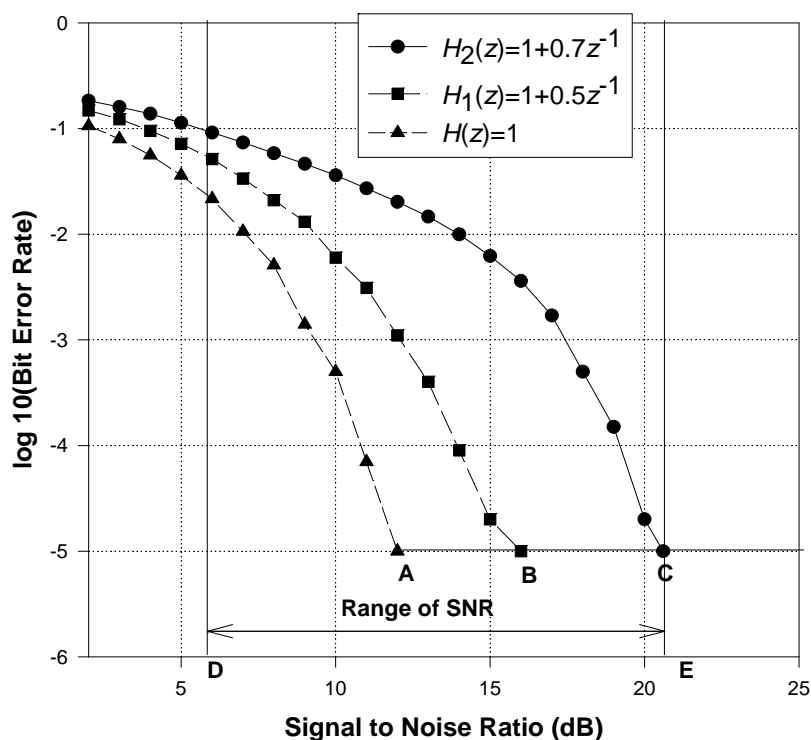


Figure 3.3: Optimal decision boundary and scatter plots for (a) SNR=6 dB,(b)SNR=10 dB, (c) SNR=20 dB with delay  $d=0$  (Channel:  $H_1(z) = 1+0.5z^{-1}$  {-ve symbols represented by triangles(pink) and +ve symbols represented by squares(blue)})

Again a careful examination of Figure 3.2 reveals that a situation may exist when the effect of ISI is not at all noticeable for a certain level of additive noise and an expected level of the error probability has to be achieved. This situation has been discussed more elaborately in Figure 3.4.



**Figure 3.4: Combined effect of ISI and additive noise on optimal BER performance**

The optimal BER plot for channel model with transfer function  $H_2(z)$  intersects a pre-fixed error probability level of  $10^{-5}$  at point 'C' and corresponding to this point the SNR level is found to be 20.6 dB. Similarly, the BER performance plot of channel without ISI defined by  $H(z)$ , intersects with the given error probability level at point 'A' (SNR=12dB). Thus it can be concluded that if the additive noise level is below the point 'D' (SNR =6dB), the BER performance characteristics with all the channels considered here are identical; hence the effect of ISI on performance remains insignificant in high noise conditions. Also beyond the point 'E'(20.6 dB), the ISI has no prominent effect on the BER performance characteristics i.e., at low noise conditions. Thus it is concluded that both the additive noise and ISI factor in a channel influence the equaliser's BER performance appreciably only over a restricted range of SNR (i.e., for realistic SNR levels).

### 3.1.2 Decision delay

The decision delay parameter is of paramount importance in adaptive channel equalisation process. Decision delay order strongly influences the location of channel states and thus the need for either linear or nonlinear equaliser structure arises. For a fixed equaliser feedforward order ' $m$ ', proper selection of delay ' $d$ ' is extremely crucial to achieve optimum performance. The effect of the various possible decision delays on optimal BER plots has been illustrated and analysed thoroughly considering for various types of channel models. The following equation yields all the possible values of delay order parameter ' $d$ ' values for optimal performance as cited in [29], which has been taken into consideration in the following simulation study.

$$d \leq m + n_a - 2 \quad (3.5)$$

The first example taken here is a two-tap minimum phase channel characterised by

$$H_1(z) = 1 + 0.5z^{-1} \quad (3.6)$$

where a Bayesian equaliser, with feedforward order  $m=2$  and decision delay  $d=0$ , yields better performance as shown in Figure 3.5a compared to  $d=1$  or 2. Corresponding optimal decision regions plots varying the delay values are depicted in Figures 3.5b-d for SNR = 10dB.

The second example considered here is a three-tap non-minimum phase channel defined by

$$H_5(z) = 0.3482 + 0.8704 z^{-1} + 0.3482 z^{-2} \quad (3.7)$$

It is observed in Figure 3.6a, the optimal BER performances with  $d = 1$  and 2 are much superior in comparison to either  $d=0$  or 3 for a fixed equaliser feedforward order  $m = 2$ . Corresponding decision regions for SNR = 10dB are plotted in Figures 3.6b-e support the performance analysis for different ' $d$ ' values.

Further, an example of a five-tap channel described by the following transfer function is given by

$$H_{11}(z) = -0.2052 - 0.5131z^{-1} + 0.7183 z^{-2} + 0.3695 z^{-3} + 0.2052 z^{-4} \quad (3.8)$$

Various values of ' $d$ ' for a fixed equaliser order  $m=2$  are considered for performance analysis. It is observed in the decision region plot for  $d=2$  shown in Figure 3.7c that the degree of non-linearity of the optimal decision boundary is milder and this helps the channel states belonging to two classes to be classified almost in a linear manner. For  $d=1$  and 3, shown in Figures 3.7b and 3.7d, the channel states belonging to two classes are non-linearly separable. But for  $d=0, 4$  and 5, these cannot be separated in two distinct classes as seen in Figures 3.7a, 3.7e and 3.7f. It is also observed that optimal BER performance is better at decision delay  $d=2$  in comparison with other delay values as depicted in Figure 3.7g, hence this result conforms to the corresponding decision region plot as illustrated in Figure 3.7c.

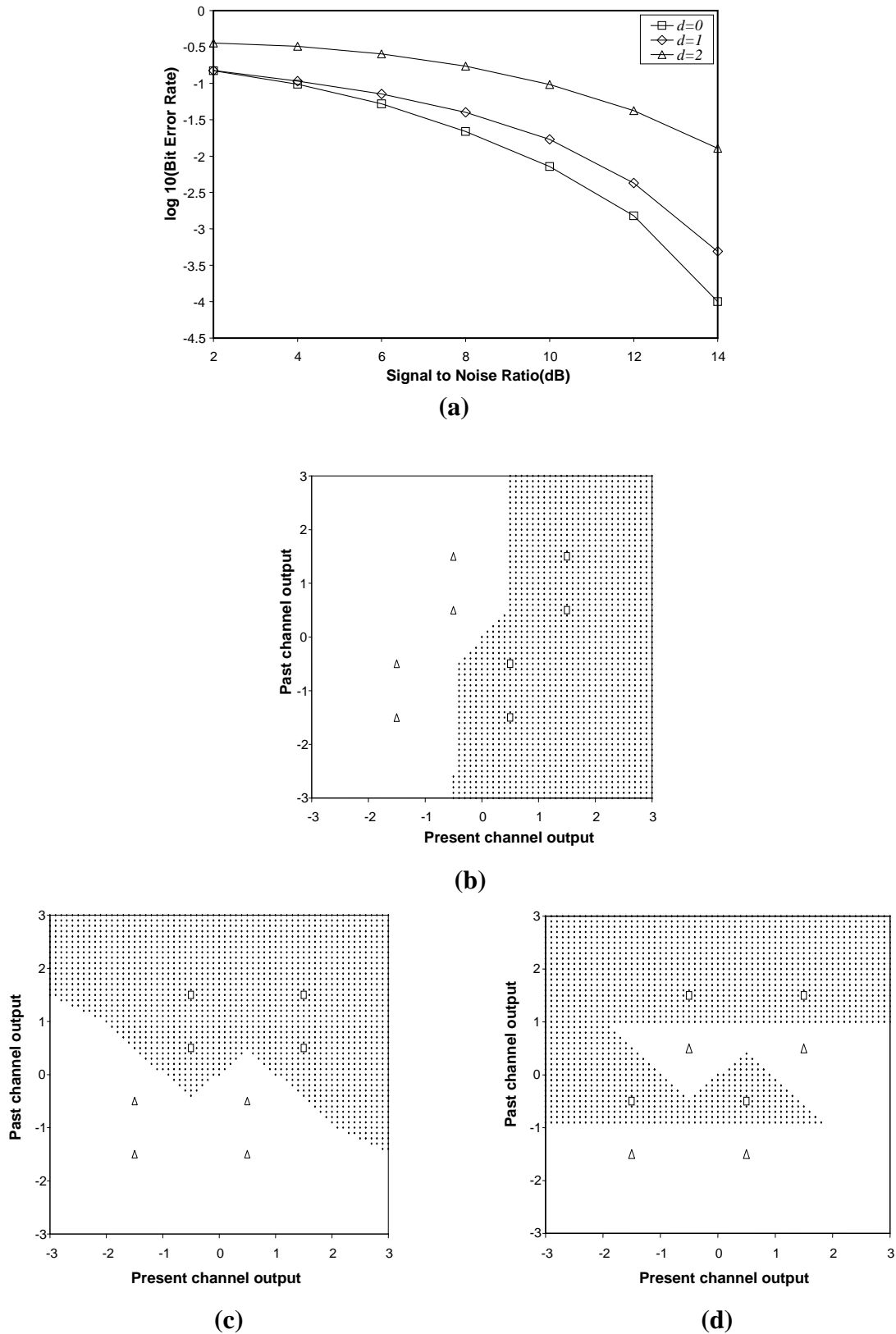


Figure 3.5 : Channel:  $H_1(z) = 1 + 0.5z^{-1}$  (a)Optimal Bayesian BER performances varying the delay parameter, Decision region plots and Noise-free channel states with various delays (b)  $d=0$ ,(c)  $d=1$ ,(d)  $d=2$  {-ve symbols represented by triangles and +ve symbols represented by rectangles}

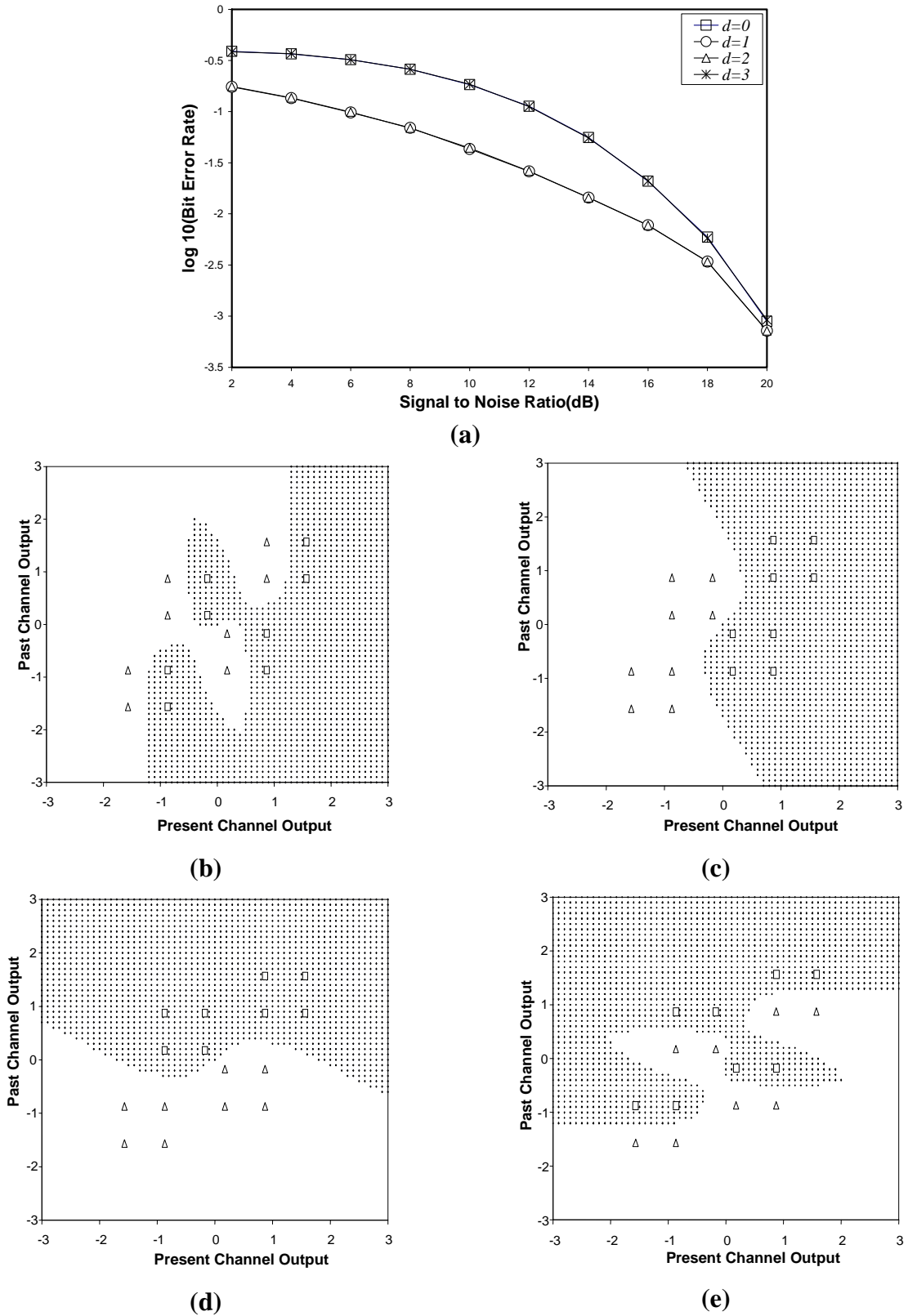


Figure 3.6 : Channel  $H_5(z) = 0.3482 + 0.8704 z^{-1} + 0.3482 z^{-2}$  (a)Optimal Bayesian BER performances varying the delay parameter, Decision region plots and noise-free channel states with different delays (b)  $d=0$ , (c)  $d=1$ , (d)  $d=2$  and (e)  $d=3$ . {-ve symbols represented by triangles and +ve symbols represented by rectangles}

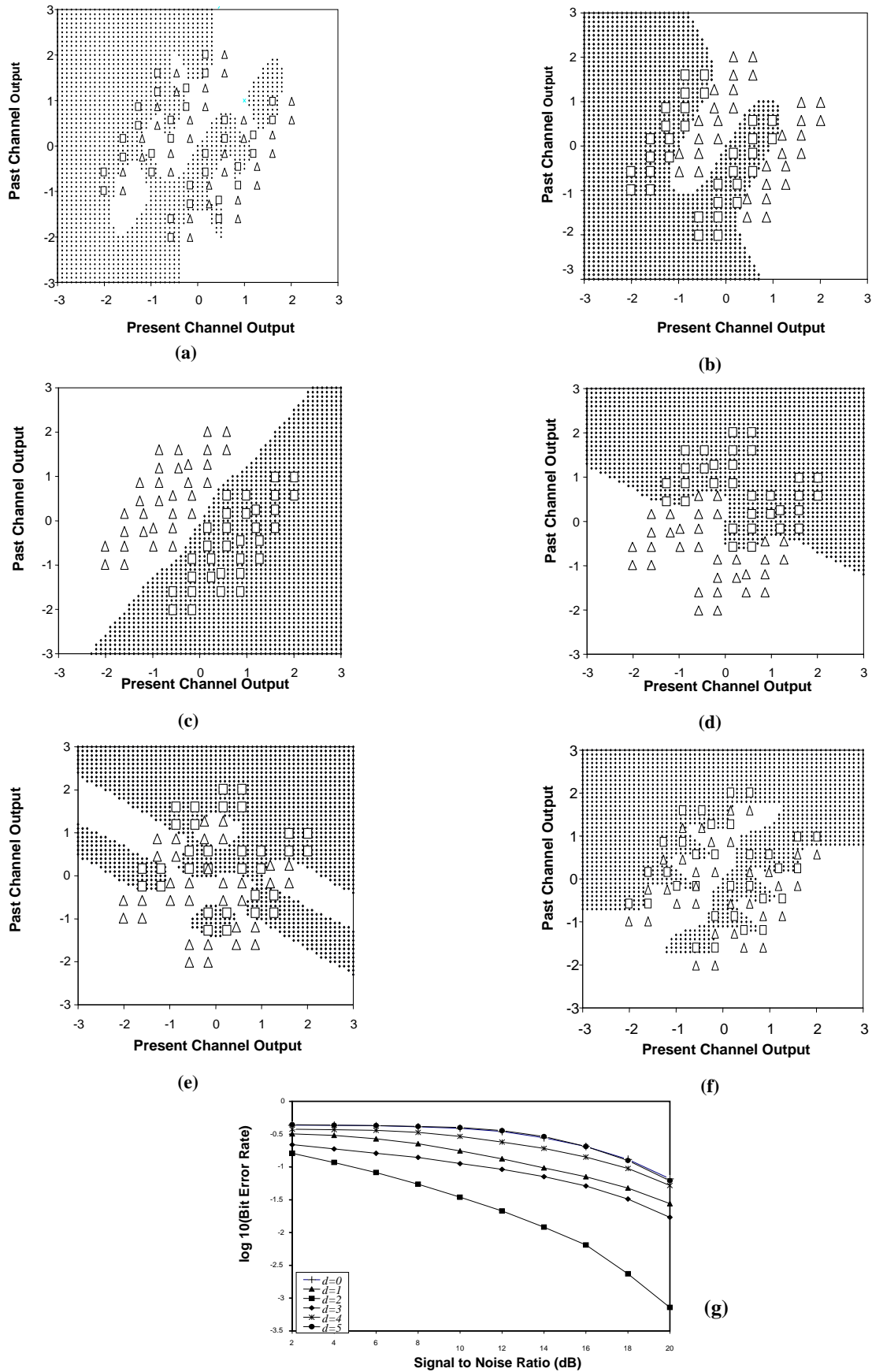


Figure 3.7 : Decision regions and noise-free channel states (a)  $d=0$  (b)  $d=1$  (c)  $d=2$  (d)  $d=3$  (e)  $d=4$  (f)  $d=5$ , and (g) optimal BER performance comparison for various delay parameters (Channel:  $H_{11}(z) = -0.2052 - 0.5131z^{-1} + 0.7183z^{-2} + 0.3695z^{-3} + 0.2052z^{-4}$ )



It is an important finding from the decision region plots that the number of noise-free channel states, close to the optimal decision boundary varies with decision delay significantly and it is less at that specific decision delay or delays for which an equaliser's BER performance is the optimal. Also, the channel states in close proximity to the optimal decision boundary have undesirable effect on the optimal BER performance, i.e., when additive noise present in the communication system is more, a higher probability of misclassification of the received sample points in the observation space is expected and performance degrades.

### **3.1.3 Equaliser order**

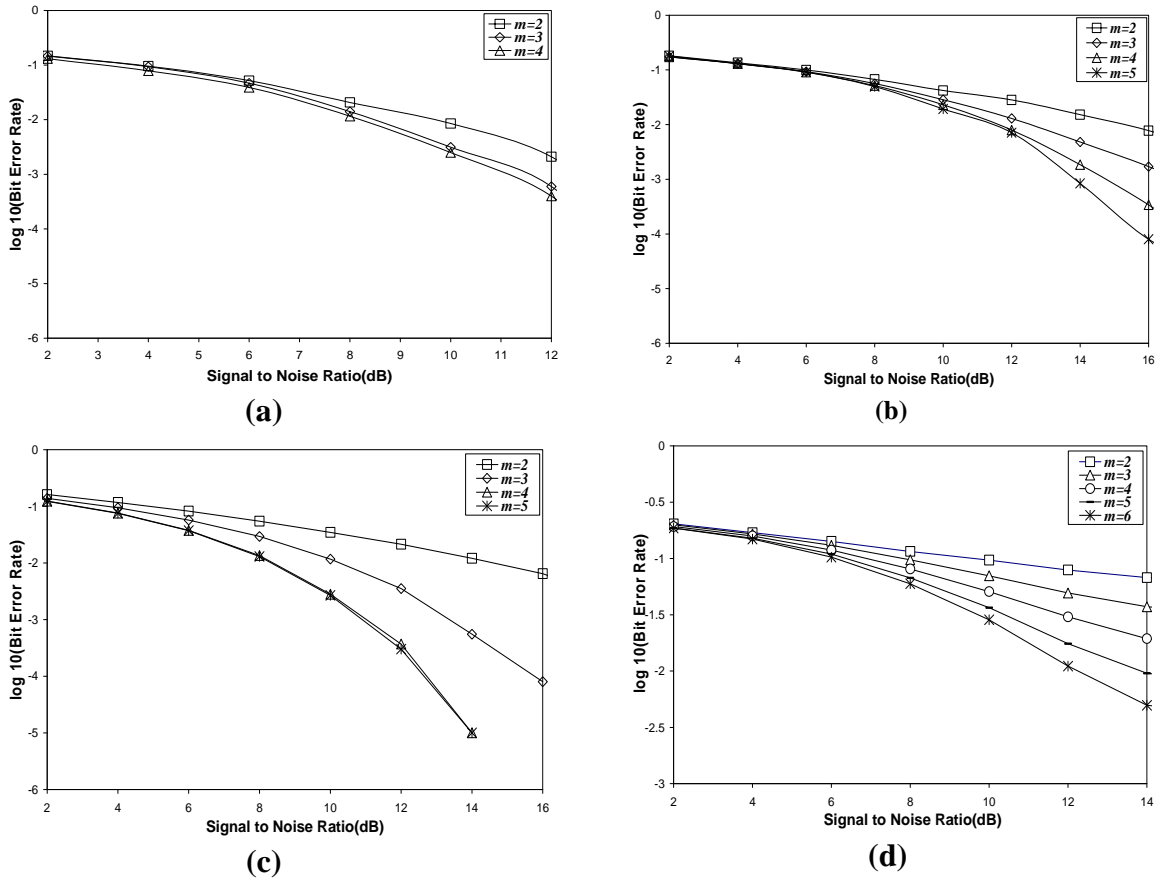
It is observed that increasing the equaliser feedforward order ' $m$ ' and properly selecting the decision delay ' $d$ ' for that specific ' $m$ ' value, result in significant improvement in performance of equaliser in general. But the structural complexity increases with larger equaliser order, which necessitates restriction of  $m$  to a certain order. Detailed study regarding this critical aspect in equaliser design is undertaken in the present work.

Figures 3.8a-c show the optimal BER performances of equalisers for three channel models by increasing the feedforward order ' $m$ '. The channel models under study are defined by transfer functions  $H_1(z)$ ,  $H_5(z)$  and  $H_{11}(z)$ . It is observed that by increasing ' $m$ ', the input observation space dimensionality of the equaliser increases too, thus reducing the chances of misclassification of the received sample points quite appreciably, which in turn results in an enhancement of equaliser's BER performance.

Further, in some typical situations it is noticed that, if some of the noise-free channel states belonging to two different classes are either coincident or very close to the decision boundary, then even increasing the equaliser's feedforward order ' $m$ ' to a higher value, satisfactory gain in performance can never be achieved. This limitation is observed in a linear 3-tap channel defined by

$$H_6(z) = 0.4084 + 0.8164 z^{-1} + 0.4084 z^{-2} \quad (3.9)$$

In this example, channel states belonging to the two different classes are coincident on the decision boundary. It is noticed from Figure 3.8d that gain in BER performance remains unsatisfactory even after increasing ' $m$ ' to 6. This poor BER performance necessitates the inclusion of the decision feedback technique, which has been discussed in the following section.



**Figures 3.8 : Optimal Bayesian BER performance curves varying equaliser's feedforward order( $m$ ) (a) channel  $H_1(z)$ , (b) channel  $H_5(z)$ , (c) channel  $H_{11}(z)$  and (d) channel  $H_6(z)$**

### 3.1.4 Effect of decision feedback

Decision feedback (either corrected or detected symbols used) improves the performance of a Bayesian equaliser and the performance loss due to error propagation, which occurs when wrongly detected symbols are passed into the feedback vector, is not significant [29]. Because of decision feedback, considerably fewer states are used in computing Bayesian decision function. Also it is observed that the decision feedback increases the minimum distance between the two classes of channel states and hence the separation of noise-free channel states from the optimal decision boundary is more in comparison to that obtained without feedback. Hence the probability of received samples crossing the optimal decision boundary reduces considerably which in turn improves performance. Previous research work has demonstrated that the decision feedback effectively merges channel states and this simplifies the equalisation process [75] by making the decision regions linearly separable. Motivated by this phenomenon, the present work further investigates geometric translation property in channel output observation space due to decision feedback considering the following examples.

All the combinations of transmitted symbol (channel input) sequences and the expected receiver sample vectors at equaliser input in each case are listed for a three-tap channel characterised by transfer function  $H_6(z)$  in Table 3.1.

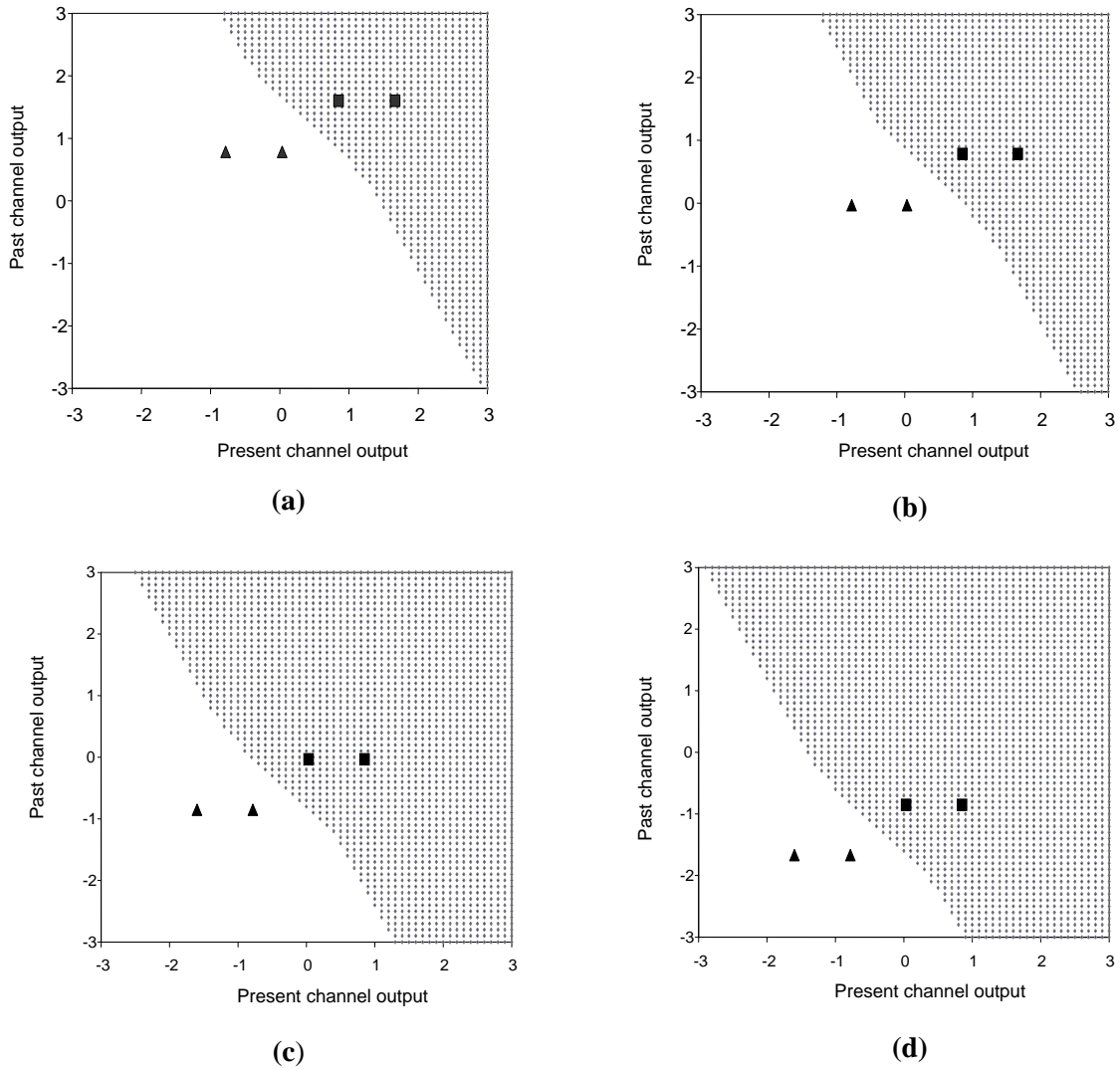
$$H_6(z)=0.4084+0.8164z^{-1}+0.4084z^{-2}$$

Here the received sample vector is the position of the symbols (channel states) in a 2-dimensional observation space. The possible combinations of transmitted symbol sequence has been represented in a matrix form (transmitted symbol sequence matrix  $\mathbf{S}$ ), which has  $2^{m+n_a-1}$  rows and  $m+n_a-1$  columns. The subset channel states for a given feedback  $[1 \ 2]^T$  are emphasized in boldface in the given Table 3.1. Here 1 and 2 denote the two classes which belong to the transmitted symbols +1 and -1 respectively.

Transmitted Symbol Sequences				Received sample vectors at equaliser input	
$s(n)$	$s(n-1)$	$s(n-2)$	$s(n-3)$	$\hat{r}(n)$	$\hat{r}(n-1)$
1	1	<b>1</b>	<b>1</b>	<b>1.6332</b>	<b>1.6332</b>
1	1	1	2	1.6332	0.8164
1	1	2	1	0.8164	0.0004
1	1	2	2	0.8164	-0.8164
2	1	<b>1</b>	<b>1</b>	<b>0.8164</b>	<b>1.6332</b>
2	1	1	2	0.8164	0.8164
2	1	2	1	-0.0004	0.0004
2	1	2	2	-0.0004	-0.8164
1	2	<b>1</b>	<b>1</b>	<b>0.0004</b>	<b>0.8164</b>
1	2	1	2	0.0004	0.0004
1	2	2	1	-0.8164	-0.8164
1	2	2	2	-0.8164	-1.6332
2	2	<b>1</b>	<b>1</b>	<b>-0.8164</b>	<b>0.8164</b>
2	2	1	2	-0.8164	-0.0004
2	2	2	1	-1.6332	-0.8164
2	2	2	2	-1.6332	-1.6332

**Table 3.1: Transmitted symbol sequences and received sample vectors at equaliser's input. Structure parameters are  $d = 1$ ,  $m = 2$ , and  $n_b = 2$ . Bold faced numbers : given feedback  $s_{f,j} = [1 \ 1]^T$  (Channel  $H_6(z) = 0.4084 + 0.8164 z^{-1} + 0.4084 z^{-2}$ )**

DFE channel states in the output observation space are solely decided by the elements of the feedback vector. The reduction in channel states for the other possible feedbacks i.e.  $[2 \ 1]^T$ ,  $[1 \ 1]^T$  and  $[2 \ 2]^T$  are shown by three coordinate translations in the respective decision region plots as depicted in Figures 3.9a-d. Basically, the decision feedback performs a space translation that maps the DFE onto an equivalent transversal equaliser in the input observation space. In the translated observation space, the subsets of channel states corresponding to the different decisions are almost linearly separable.



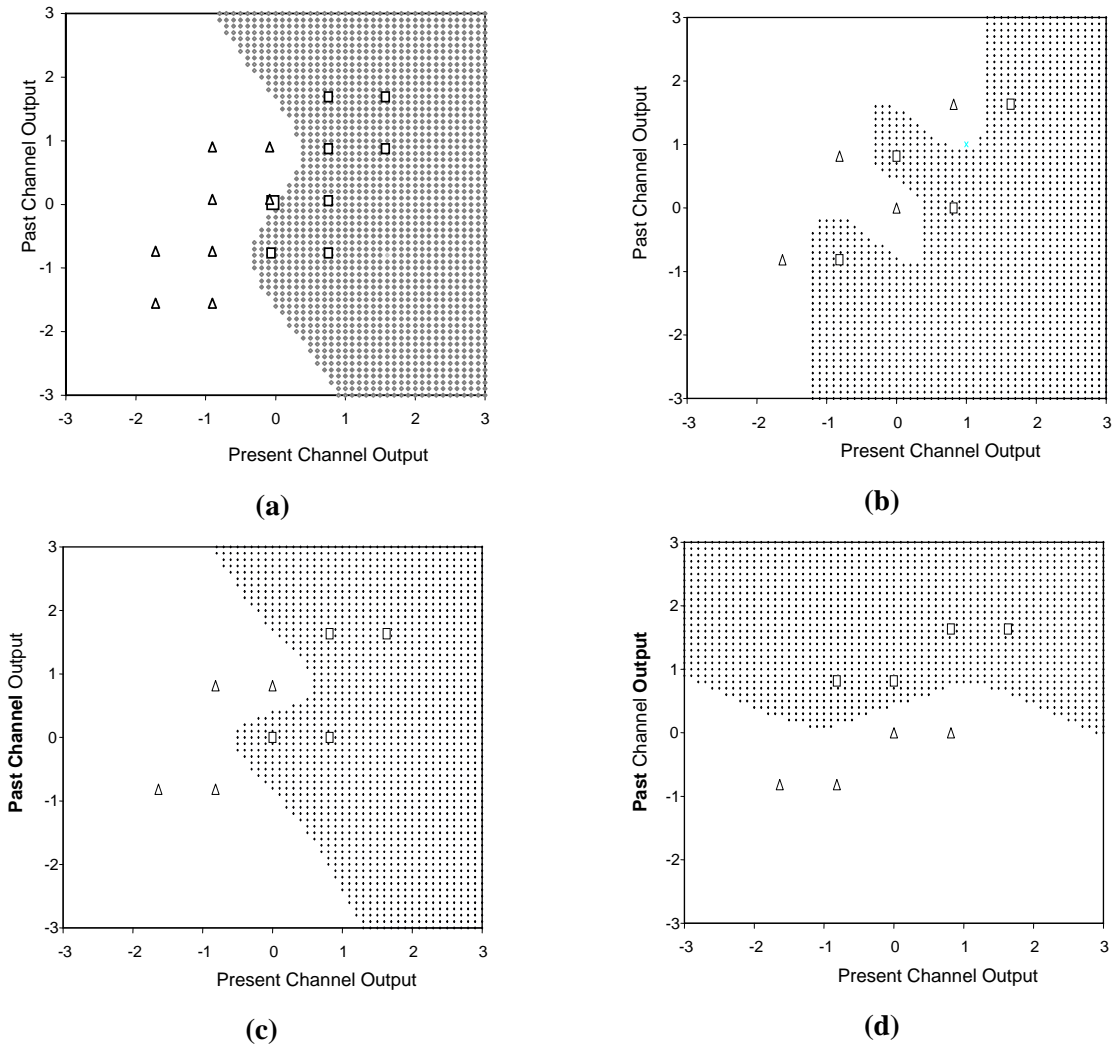
**Figure 3.9 :** Effect of feedback vector on the optimal decision regions and Noise-free channel states with  $m=2$ ,  $n_b=2$ ,  $d=1$  and SNR= 10dB (a) Feedback Vector  $[1 \ 1]^T$ , (b) Feedback Vector  $[1 \ 2]^T$ , (c) Feedback Vector  $[2 \ 1]^T$  and (d) Feedback Vector  $[2 \ 2]^T$  (Channel :  $H_6(z)=0.4084+0.8164z^{-1}+0.4084z^{-2}$ ) {-ve symbols represented by triangles and +ve symbols represented by rectangles}

The geometric properties of a DFE are further illustrated here. Occurrence of coincident channel states, which is prominently noticed in the decision region plot without feedback in Figure 3.10a, can be completely eliminated by incorporating decision feedback. It is observed that when the decision feedback is used with the feedback order  $n_b=1$  (which means that symbol from either class 1 or class 2 is feedback), the number of noise free channel states are reduced from 16 (i.e., without feedback) to 8 as depicted in Figures 3.10b. This is because the number of channel states is now reduced by an order of magnitude ( $2^{n_b}$ ) [29] and those are called virtual channel states which actively take part in decisions. Again it is observed that with a specific decision delay  $d=2$ , only four states in the vicinity of the optimal decision boundary are vulnerable in comparison to that with different decision delay values (i.e.,  $d=0$  or 1), and hence it is expected that for  $d=2$  the equaliser's performance improves significantly. Increasing the feedback order ' $n_b$ ' to 2 and 3, the number of vulnerable channel states are further reduced to 4 and 2 respectively as observed in Figure 3.11. Optimal BER comparisons utilising decision feedback ( $n_b=1$ ,  $n_b=2$  and  $n_b=3$ ) and for various  $d$  values are illustrated in Figure 3.12.

Further, it is found that both decision delay ' $d$ ' and feedback order ' $n_b$ ' are of much importance in deciding the performance of a DFE. In Figure 3.13a it has been shown that for the 5-tap channel model  $H_{11}(z)$  ( $H_{11}(z) = -0.2052 - 0.5131z^{-1} + 0.7183z^{-2} + 0.3695z^{-3} + 0.2052z^{-4}$ ), if an equaliser with feedback order  $n_b=1$  is chosen, the total number of noise-free channel states are reduced to 32 from 64 (i.e., equaliser without feedback). Decision region plots in Figures 3.13b-e show that increasing ' $n_b$ ' to 2, 3, 4 and 5, the number of channel states are further reduced to 16, 8, 4 and 2 respectively.

Optimal decision region plots in Figures 3.14a-c show the effect of varying the decision delay order ' $d$ ' for a fixed feedback order ' $n_b$ ' in another example of a 2-tap channel model  $H_1(z)$  ( $H_1(z) = 1 + 0.5z^{-1}$ ). The corresponding optimal BER plots are provided in Figure 3.14d.

A careful examination of various optimal decision region plots illustrated here reveal that reduction in the number of noise-free channel states depends on feedback order ' $n_b$ ' in a DFE structure. The decision delay order chosen decides the location of channel states and number of channel states in close proximity of the decision boundary. It is observed that decision delay further causes a rotation of the decision boundary. All these effects combine to contribute for the improvement in the BER performance due to decision feedback.



**Figure 3. 10 :** Decision region plots (a) without feedback with delay  $d = 1$ , With feedback and for different values of delay parameter (b)  $d = 0$ , (c)  $d = 1$ , (d)  $d = 2$  (Channel :  $H_6(z) = 0.4084 + 0.8164z^{-1} + 0.4084z^{-2}$ ) {-ve symbols represented by triangles and +ve symbols represented by rectangles}

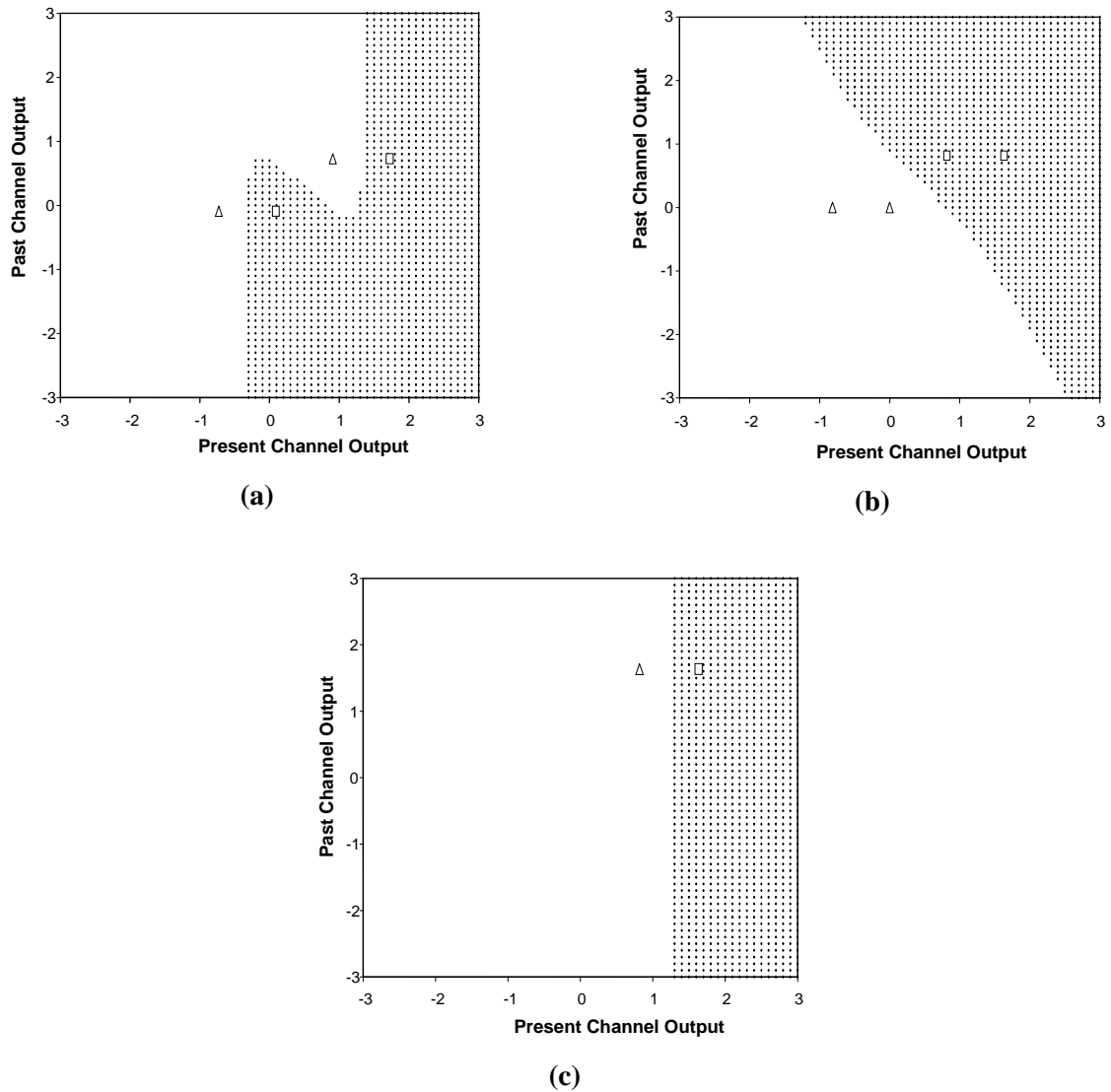
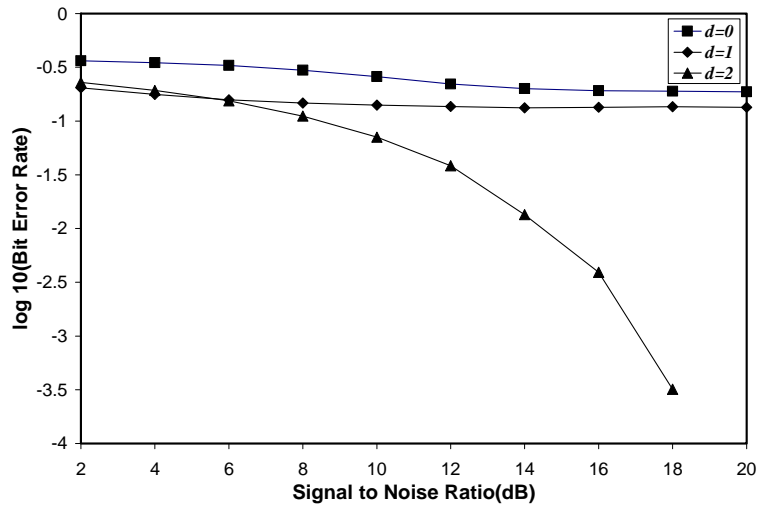
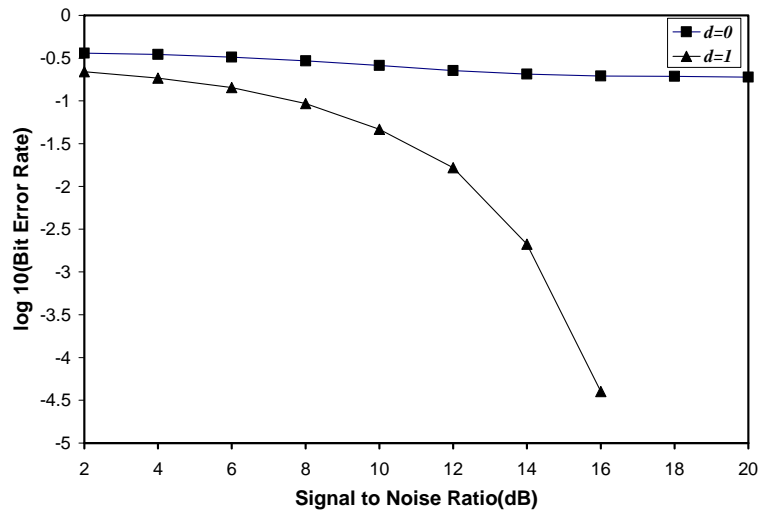


Figure 3.11 : Decision region plots for different combinations of feedback order and delay parameter (Channel  $H_6(z)$ ), (a)  $n_b=2$  and  $d=0$ , (b)  $n_b=2$  and  $d=1$  and (c)  $n_b=3$  and  $d=0$  ) {-ve symbols represented by triangles and +ve symbols represented by rectangles}

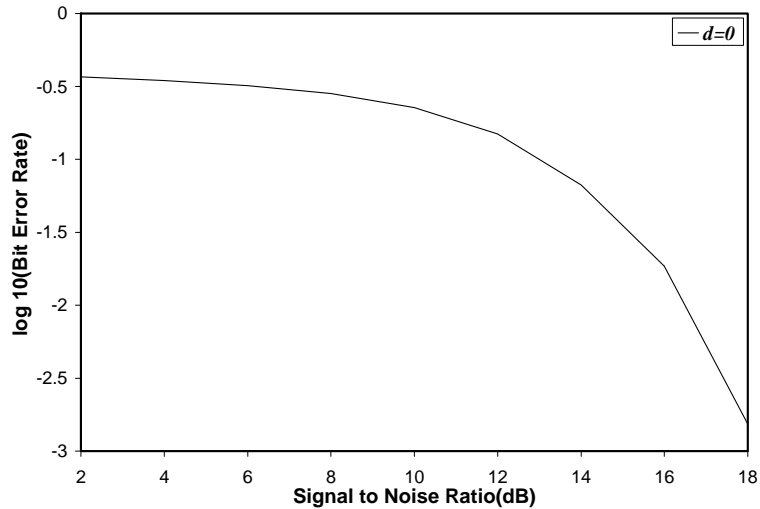




(a)



(b)



(c)

Figure 3.12 : Optimal performance comparisons varying the delay parameter and with feedback (a)  $n_b=1$ , (b)  $n_b=2$  and (c)  $n_b=3$  (Channel:  $H_6(z) = 0.4084 + 0.8164 z^{-1} + 0.4084 z^{-2}$ )

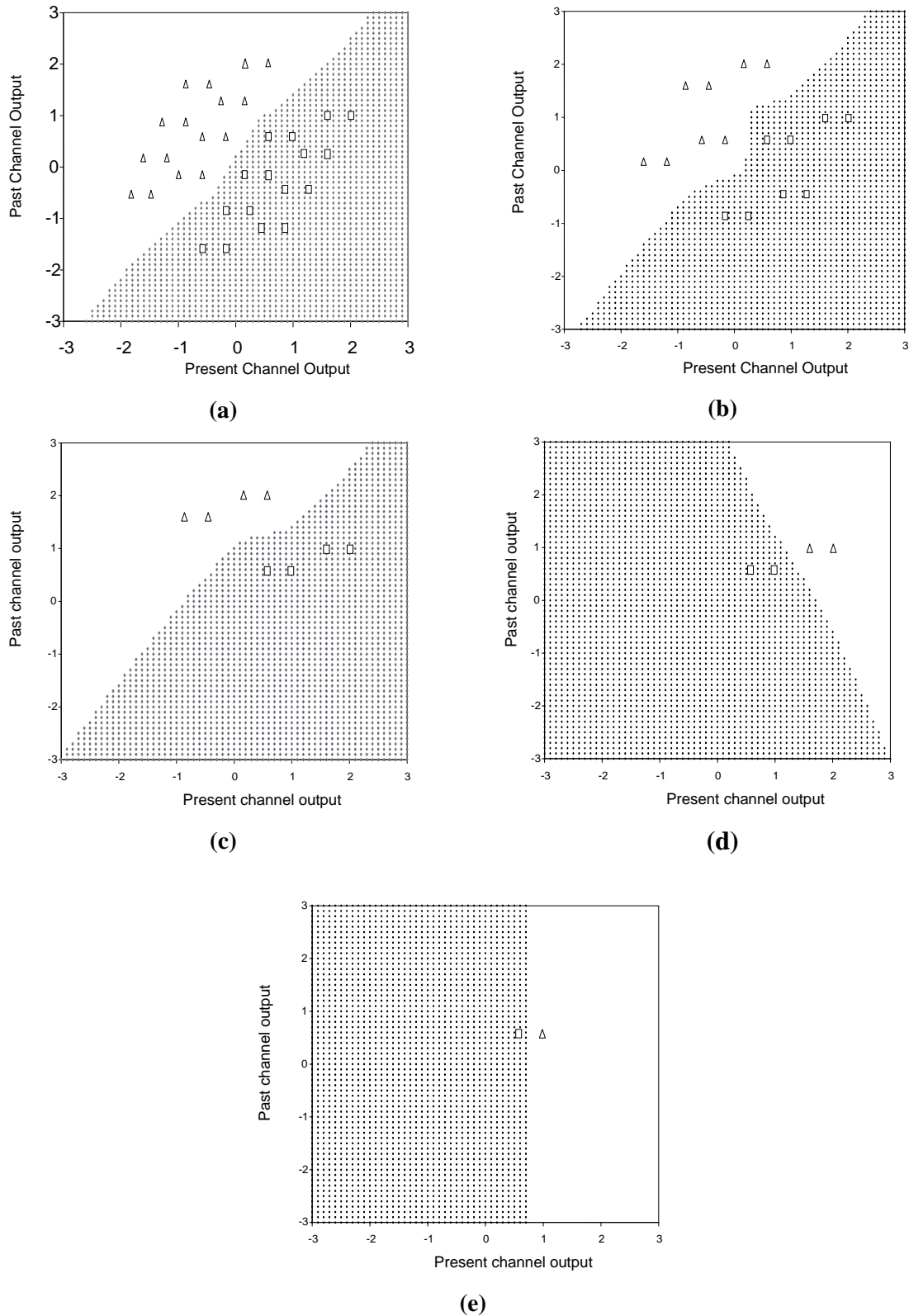


Figure 3.13 : Optimal decision region plots (a)  $n_b=1$  and  $d=2$ , (b)  $n_b=2$  and  $d=2$ , (c)  $n_b=3$  and  $d=2$ , (d)  $n_b=4$  and  $d=1$  (e)  $n_b=5$  and  $d=0$  (Channel :  $H_{11}(z) = -0.2052 - 0.5131z^{-1} + 0.7182z^{-2} + 0.3695z^{-3} + 0.2052z^{-4}$  {-ve symbols represented by triangles and +ve symbols represented by rectangles})

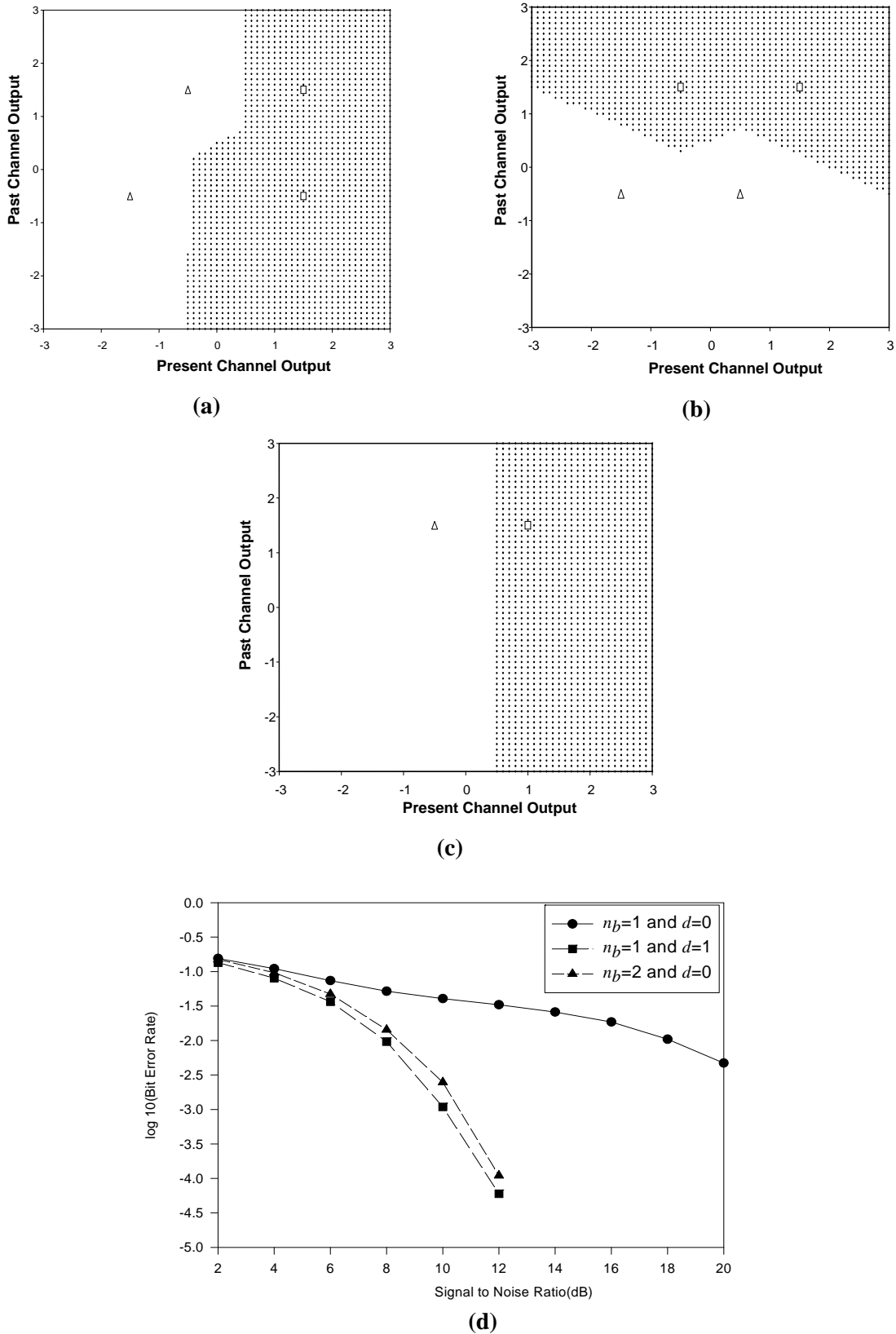


Figure 3.14 : Decision region plots (a)  $n_b=1$  and  $d=0$ , (b)  $n_b=1$  and  $d=1$ , (c)  $n_b=2$  and  $d=0$  { -ve symbols represented by triangles and +ve symbols represented by rectangles } (d) Optimal Bayesian performance comparisons for 'm'=2 (Channel :  $H_1(z)=1+0.5z^{-1}$ )

Here, an important observation can be made from Figure 3.15, which contradicts the very concept that improvement in BER performance is guaranteed simply by increasing the feedback order of a DFE for a fixed delay parameter. Thus it can be inferred here that both decision delay ' $d$ ' and feedback order ' $n_b$ ' play a vital role in deciding the optimal BER performance of a Bayesian DFE.

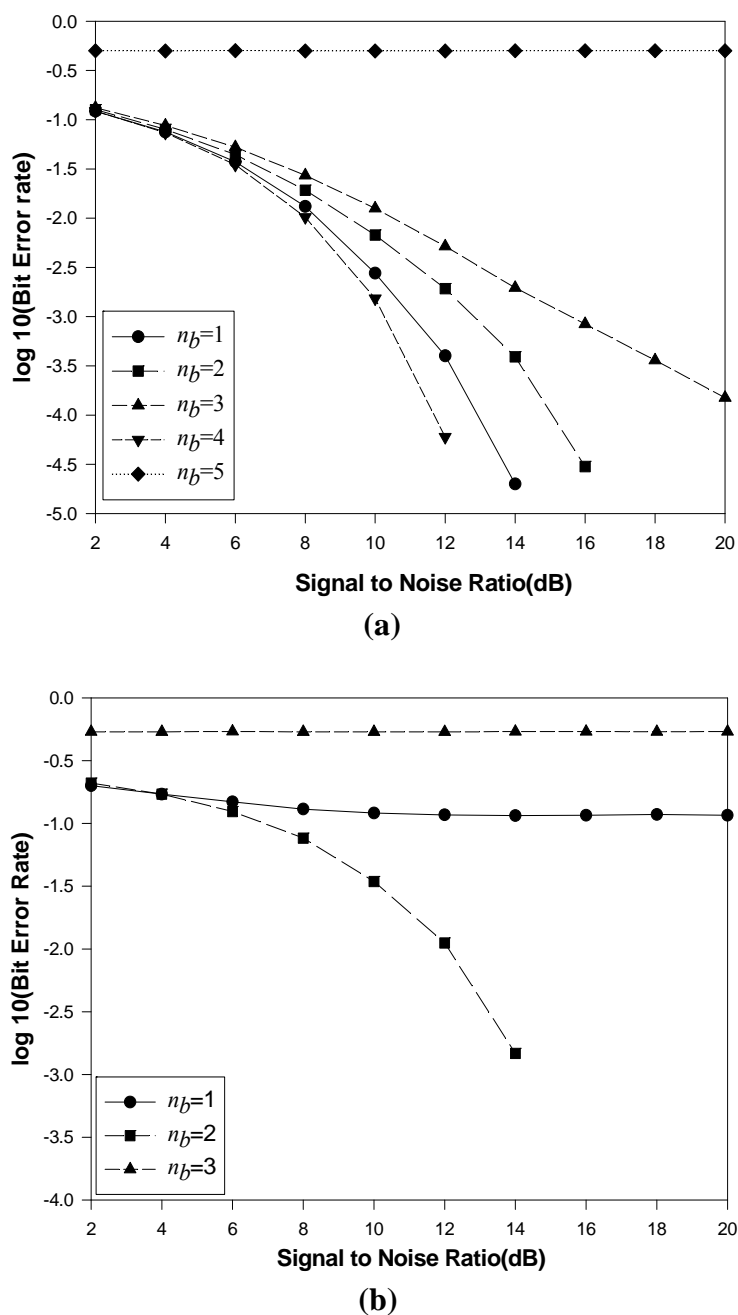


Figure 3.15: Optimal Bayesian performance curves varying the feedback order ' $n_b$ '  
 (a) Channel  $H_{11}(z)$  with  $m=5$  and  $d=4$   
 (b) Channel  $H_6(z)$  with  $m=2$  and  $d=1$

### 3.1.5 Importance of selection of 'n<sub>b</sub>' and 'd' in a DFE structure

Selection of right combination of feedback order 'n<sub>b</sub>' and decision delay 'd' for a fixed feedforward order 'm' in DFE structure plays a major role in obtaining the optimal performance. Detailed simulation study has been undertaken by exhausting all possible combinations of 'n<sub>b</sub>' and 'd' for a given 'm', based on the already established relationship [29] as given by

$$n_b = m + n_a - 2 - d \quad (3.9)$$

Figure 3.16a shows that for channel  $H_6(z)$  if equaliser feedforward order 'm' is fixed at 3, then combination of  $n_b=2$  and  $d=2$  results in improved performance in comparison to all other possible combinations of 'n<sub>b</sub>' and 'd'. Similarly for different channel models various combinations of 'n<sub>b</sub>' and 'd' for a fixed value of 'm' in the equaliser design are selected based on Equation (3.9) and the optimal BER performance comparisons curves are illustrated in Figures 3.16b-e. It has been concluded from these extensive simulation results that only a specific combination of 'n<sub>b</sub>' and 'd' for a fixed 'm' yields superior performance amongst all. These optimal combinations of 'm', 'n<sub>b</sub>' and 'd' for different channel models are summarised below in Table 3.2.

Channel transfer functions	Optimal combination of equaliser parameters
$H_6(z)=0.4084 + 0.8164z^{-1}+0.4084z^{-2}$	$m=3, n_b=2$ and $d=2$
$H_1(z)= 1+ 0.5z^{-1}$	$m=2, n_b=1$ and $d=1$
$H_5(z)=0.3482+ 0.8704 z^{-1}+ 0.3482 z^{-2}$	$m=3, n_b=2$ and $d=2$
$H_{10}(z)=0.35+0.8 z^{-1}+1 z^{-2} +0.8 z^{-3}$	$m=2, n_b=3$ and $d=1$
$H_{11}(z) = -0.2052-0.5131z^{-1}+ 0.7183 z^{-2}+0.3695 z^{-3}+0.2052 z^{-4}$	$m=5, n_b=4$ and $d=4$

**Table 3.2: Optimal combination of parameters of a Bayesian DFE**

An important observation, made from the various optimal BER plots shown in Figure 3.16a-e is that a Bayesian equaliser without decision feedback but with a proper delay order parameter can outperform those with decision feedback incorporating non-optimal combination of 'n<sub>b</sub>' and 'd' parameters. Hence, inclusion of decision feedback in Bayesian equaliser does not enhance the optimal BER performance significantly unless the combination of parameters 'm', 'n<sub>b</sub>' and 'd' is rightly chosen. For example, the Bayesian equaliser without feedback and with  $d=1$  outperforms DFEs without proper combination of feedback order and decision delay as observed in Figure 3.16c.

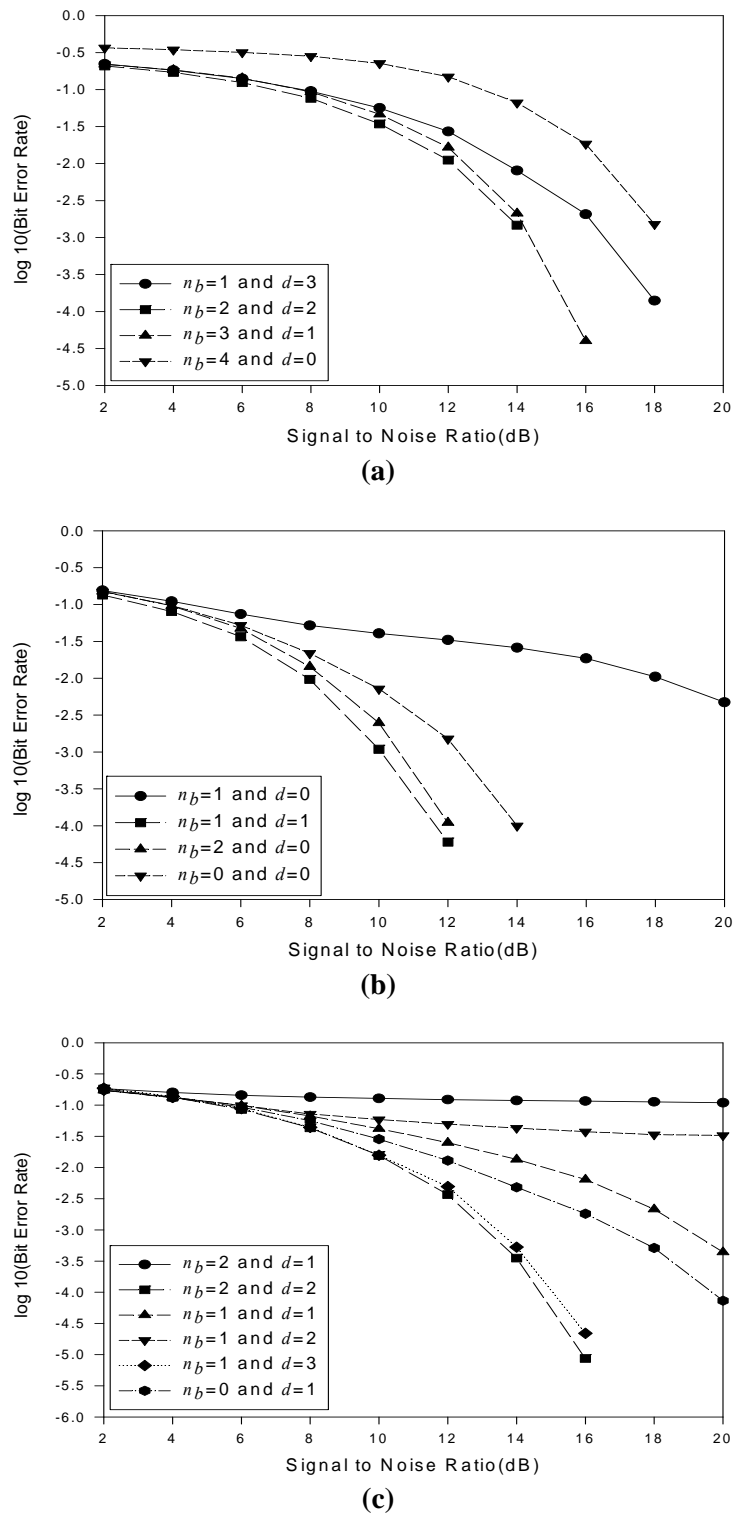
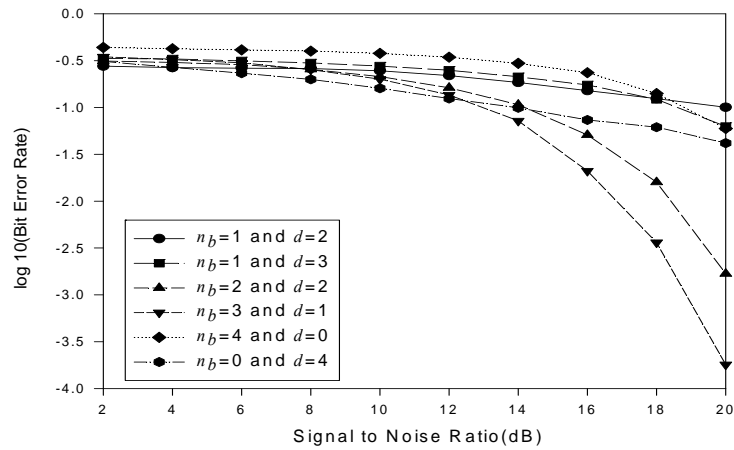
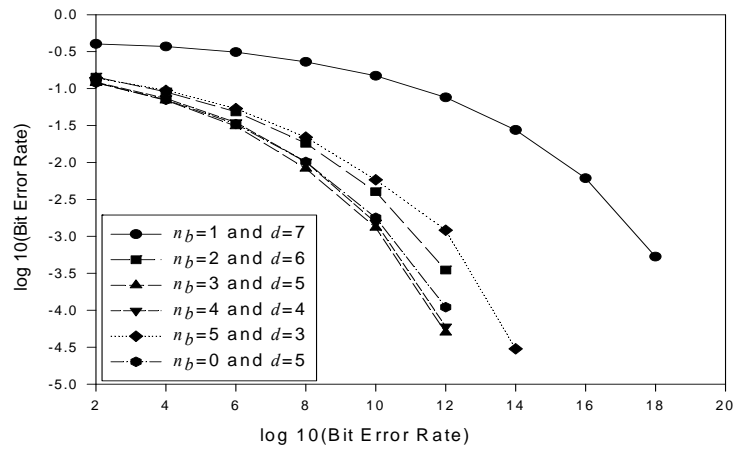


Figure 3.16 : Optimal Bayesian performance comparisons (a) Channel  $H_6(z)$  with  $m=3$ , (b) Channel  $H_1(z)$  with  $m=2$ , (c) Channel  $H_5(z)$  with  $m=3$



(d)



(e)

Figure 3.16 : Optimal Bayesian performance comparisons (d) Channel  $H_{10}(z)$  with  $m=2$  and (e) Channel  $H_{11}(z)$  with  $m=5$

### 3.2 A new approach for selection of equaliser parameters

So far as equalisers without decision feedback is concerned, in order to arrive at an optimal value of decision delay ' $d_{opt}$ ' for a given feedforward order ' $m$ ', a set of graphical plots for various possible  $d$  values is required. This process of sorting the numerous plots to pick ' $d_{opt}$ ' value is extremely cumbersome. In order to circumvent such a problem a logical explanation has been provided for establishing empirical relationships for ' $d_{opt}$ ' purely from induction, which has been graphically interpreted for the sake of brevity. Thus the new approach can be applied to parameter selection process by simply examining the channel characteristics.

Similarly, a proper selection of feedforward order ' $m$ ', decision delay ' $d$ ' and feedback order ' $n_b$ ' is a challenging task in equalisers with decision feedback as these design parameters play a crucial role in the BER performance. Some studies have already been made regarding this aspect as reported in the literatures[86,95,96]. In the present research work, attempt has been made to interpret this concept in a different way taking into account the transmitted symbol sequence matrix **S**. The column span ' $ncol$ ' of the **S** matrix is the deciding factor for the selection of optimal values of feedback order ' $n_{b(opt)}$ ' and decision delay ' $d_{opt}$ ' for a fixed feedforward order ' $m$ ' of the equaliser. Logical explanations for the new approach have been provided and interpreted also in the form of mathematical expressions.

A major breakthrough has been thus achieved in successfully evaluating the key design parameters of equalisers with and without feedback from the channel characteristics directly. The efficacy of the proposed approach has been verified by considering examples of various symmetric and asymmetric channel models.

#### 3.2.1 Equaliser without decision feedback

In the equaliser structure without feedback, the optimal value of decision delay ' $d_{opt}$ ' has been evaluated for a fixed equaliser feedforward order ' $m$ '. In the proposed approach the selection of ' $d_{opt}$ ' is decided broadly by the two types of channel i.e., whether symmetric or asymmetric. Separate analysis for each type has been carried out in the following subsections.



### 3.2.1.1 Symmetric channels

For symmetric channel a methodology has been developed for computing the optimal decision delay ' $d_{opt}$ '. The column span ' $ncol$ ' of the transmitted symbol sequence matrix  $\mathbf{S}$ , which is generally used to determine the channel states, as given in Table 3.1, satisfies the following relationship.

$$ncol = m + n_a - 1 \quad (3.10)$$

where ' $m$ ' is the equaliser's feedforward order and ' $n_a$ ' is the channel order.

As the equaliser performance depends on the energy content of the individual channel taps [29], the concept utilised here is that the energy is distributed over the entire column span ( $ncol$ ) of the  $\mathbf{S}$  matrix. While this procedure is adopted it is seen that ' $ncol$ ' generally exceeds the channel order ' $n_a$ '. Hence the distribution of energy will be such that the channel tap coefficient representing the maximum amplitude, will have a major share of the column span. Also due to the mirror-symmetry of the channel characteristics, the remaining tap coefficients will occupy the column space (one for each tap coefficient) in a symmetric fashion about the position of tap coefficient with the maximum amplitude ' $tap_{max}$ '. The allocation of column space one by one starting from both the ends of the  $\mathbf{S}$  matrix continues simultaneously till all the tap coefficients get exhausted leaving behind the  $tap_{max}$  only, to occupy the rest of the column space. This column space left for ' $tap_{max}$ ' is directly related to the evaluation of the optimal decision delay ' $d_{opt}$ ' for a given feedforward order ' $m$ '. The interpretation of this statement is that depending upon the column span left for  $tap_{max}$ , as already mentioned, the solution of ' $d_{opt}$ ' can be multi-valued. Hence any one of the ' $d_{opt}$ ' can be chosen to achieve the optimal BER performance of equaliser for a fixed feedforward order ' $m$ ' which also signifies that the BER performance for all of the ' $d_{opt}$ ' values are exactly identical.

The mathematical expression governing ' $d_{opt}$ ' values is given by

$$d_{min} \leq d_{opt} \leq d_{max} \quad (3.11)$$

where, ' $d_{min}$ ' and ' $d_{max}$ ' both are dependent upon the ' $tap_{max}$ ' and are defined as given below.

$$d_{min} = tap_{max} - 1 \quad (3.12)$$

$$d_{max} = tap_{max} + m - 2 \quad (3.13)$$

**Example 1:**

The first example of a symmetric channel considered here is represented by

$$H_5(z) = 0.3482 + 0.8704 z^{-1} + 0.3482 z^{-2}$$

In this three-tap channel  $H_5(z)$  for a feedforward order  $m=2$ , the column span is  $ncol = 4$  as given by Equation (3.10) and shown in Figure 3.17. The position of the tap coefficient with maximum amplitude is  $tap_{max} = 2$ . Hence the values of optimal decision delay decided by Equation (3.11) is

$$1 \leq d_{opt} \leq 2$$

where  $d_{min} = 1$  and  $d_{max} = 2$  are given by Equation (3.12) and Equation (3.13) respectively. The optimal BER performance of equaliser, given in Figure 3.19a for different delay values, clearly shows that the optimal values of decision delay are  $d_{opt} = 1$  or  $2$  for a feedforward order  $m=2$ . These results are exactly identical with that obtained from the mathematical expressions based on the new approach as discussed.

**Example 2:**

The second example of a symmetric channel is given by

$$H_{13}(z) = 0.227 + 0.46 z^{-1} + 0.688 z^{-2} + 0.46 z^{-3} + 0.227 z^{-4}$$

Figure 3.18 shows that if a feedforward order  $m=2$  is chosen, then the column span is  $ncol = 6$  for this five-tap channel  $H_{13}(z)$  as given by Equation (3.10). The position of the tap coefficient with maximum amplitude is  $tap_{max} = 3$ . Hence the values of optimal decision delay given by Equation (3.11) is

$$2 \leq d_{opt} \leq 3$$

where  $d_{min} = 2$  and  $d_{max} = 3$  are obtained from Equation (3.12) and Equation (3.13) respectively.

The optimal BER performance illustrated in Figure 3.19b for different delay values, shows that the optimal decision delay values are  $d_{opt} = 2$  or  $3$  for a feedforward order  $m=2$ , which are exactly identical with that obtained from the mathematical expressions directly based on the new approach for parameter selection.

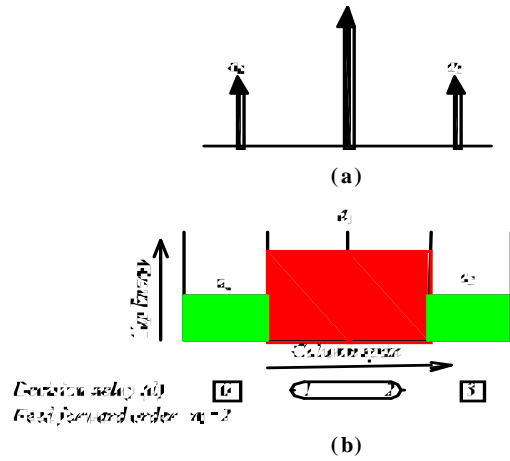


Figure 3.17: Symmetric Channel  $H_2(z)$   
(a) Impulse Response  
(b) Selection of optimal decision delay

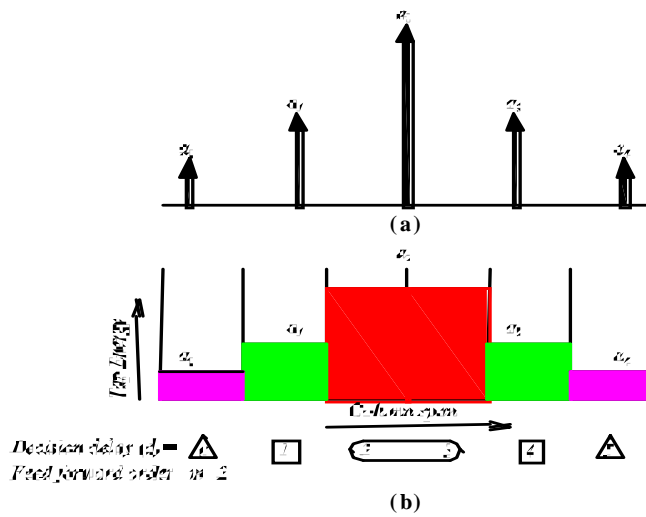
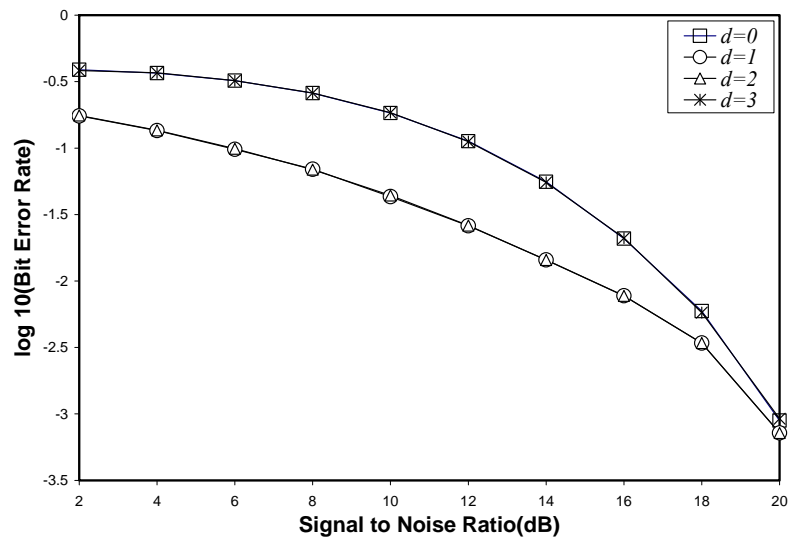
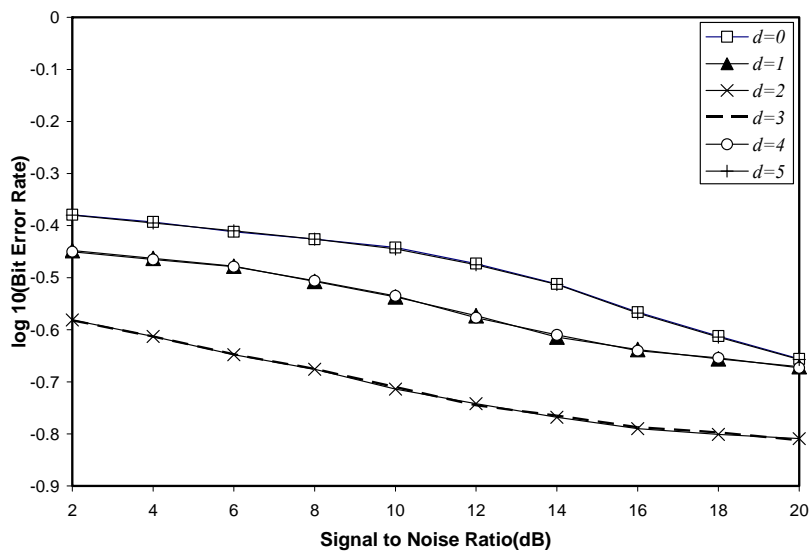


Figure 3.18: Symmetric Channel  $H_3(z)$   
(a) Impulse Response  
(b) Selection of optimal decision delay



(a)



(b)

Figure 3.19 : Optimal BER performance comparisons of Symmetric Channels varying delay values (a)  $H_5(z)$  with  $m=2$  and (b)  $H_{13}(z)$  with  $m=2$

### 3.2.1.2 Asymmetric channels

The energy distribution of an asymmetric channel is quite different from a symmetric channel as is seen in the previous section. However, the common feature between the two types of channel is the position of the channel tap coefficient representing the maximum amplitude ' $tap_{max}$ ', which dominates the BER performance. It can rather be said that ' $tap_{max}$ ' dictates the optimal value of decision delay parameter ' $d_{opt}$ ' to be selected for a fixed equaliser feedforward order ' $m$ '. Here also a significant contribution of ' $tap_{max}$ ' exists so far as the spread of the column ' $ncol$ ' of the **S** matrix is concerned. As symmetry is not maintained in such type of channel characteristics, uneven distribution of tap energy along the column span is bound to occur. Though due to mirror-symmetry exactly identical BER performances are obtained for all the ' $d_{opt}$ ' values in symmetrical channels, this does not happen in case of asymmetrical ones. But like its symmetrical counterparts the other tap coefficients do occupy the column space based on their respective positions. Further it is inferred that the BER performances are greatly influenced by the decision delay ' $d$ ' values as decided by the positions of tap coefficients in the channel characteristics and their corresponding amplitudes.

Asymmetric channels have been primarily configured into two groups depending upon whether channel order ' $n_a$ ' is even or odd. In the light of the above concept, different empirical relationships have been determined separately for those two groups which are entirely based on the position of the tap coefficient with the maximum amplitude ' $tap_{max}$ ' in the channel characteristics for a fixed feedforward order ' $m$ '. Following the new approach any one of the ' $d_{opt}$ ' values can be chosen in order to achieve optimal BER performance. The expression governing ' $d_{opt}$ ' is given by Equation (3.11).

The values of ' $d_{min}$ ' and ' $d_{max}$ ' both are dependent upon ' $tap_{max}$ ' directly and the mathematical expressions relating these parameters are given as below.

$$\left. \begin{array}{l} d_{min} = tap_{max} - 1 \\ d_{max} = tap_{max} + m - 3 \end{array} \right\} \text{ for } \begin{array}{l} tap_{max} \leq n_a/2 \quad (\text{if } n_a \text{ is even}) \\ tap_{max} \leq (n_a + 1)/2 \quad (\text{if } n_a \text{ is odd}) \end{array} \quad (3.14)$$

$$\left. \begin{array}{l} d_{min} = tap_{max} \\ d_{max} = tap_{max} + m - 2 \end{array} \right\} \text{ for } \begin{array}{l} tap_{max} > n_a/2 \quad (\text{if } n_a \text{ is even}) \\ tap_{max} > (n_a + 1)/2 \quad (\text{if } n_a \text{ is odd}) \end{array} \quad (3.15)$$

**Example 1:**

The first example of a two-tap asymmetric channel considered here is represented by

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

Figure 3.20 shows that if a feedforward order  $m=2$  is chosen, then the column span is  $ncol = 3$  following the Equation (3.10). The position of the tap coefficient with maximum amplitude  $tap_{max}= 1$  and the channel order is even i.e.,  $n_a=2$ . Hence the values of optimal decision delay given by Equation (3.11) is

$$d_{opt} = 0$$

where  $d_{min}=0$  and  $d_{max}=0$  are computed by Equation (3.14).

The BER performance of equaliser for different delay values shown in Figure 3.22a indicates that the optimal value of decision delay values is  $d_{opt}=0$  only for a feedforward order  $m=2$ , which is exactly identical with the value obtained from the mathematical expressions derived in the new approach.

**Example 2:**

The second example of a two-tap asymmetric channel is represented by

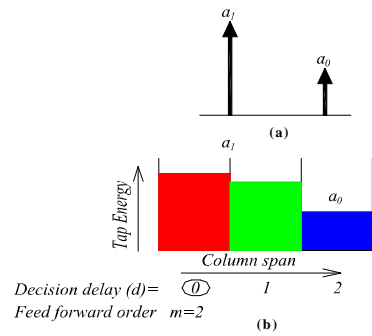
$$H_4(z)=0.5 + 1 z^{-1}$$

If a feedforward order  $m=2$  is chosen, then the column span is  $ncol = 3$  as shown in Figure 3.21. The position of the tap coefficient with maximum amplitude  $tap_{max}=2$  and the channel order is even i.e.,  $n_a=2$ . Hence the values of optimal decision delay based on Equation (3.11) is given by

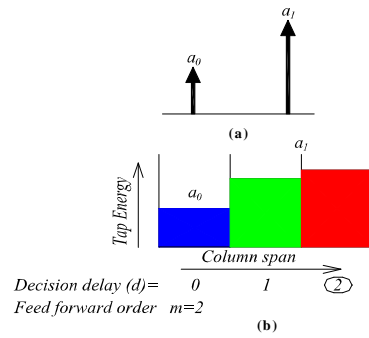
$$d_{opt} = 2$$

where  $d_{min}=2$  and  $d_{max}=2$  are obtained from Equation (3.15).

The BER performance of the equaliser for different delay values, given in Figure 3.22b, shows that the optimal decision delay value here is  $d_{opt}=2$  only for a feedforward order  $m=2$ , which is exactly identical with the value obtained from the mathematical expressions based on the new approach as discussed earlier.



**Figure 3.20: Asymmetric Channel  $H_1(z)$**   
**(a) Impulse Response**  
**(b) Selection of optimal decision delay**



**Figure 3.21: Asymmetric Channel  $H_3(z)$**   
**(a) Impulse Response**  
**(b) Selection of optimal decision delay**

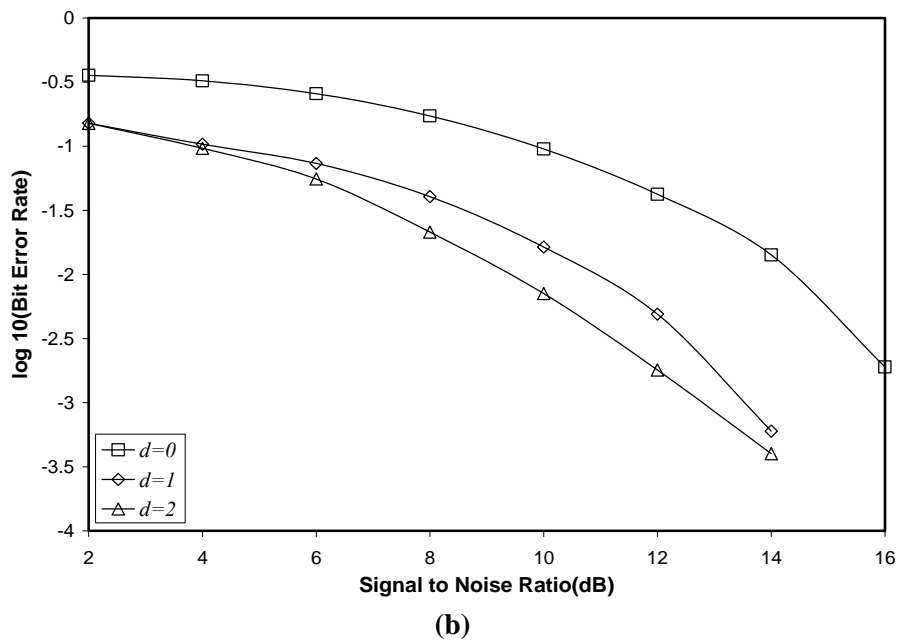
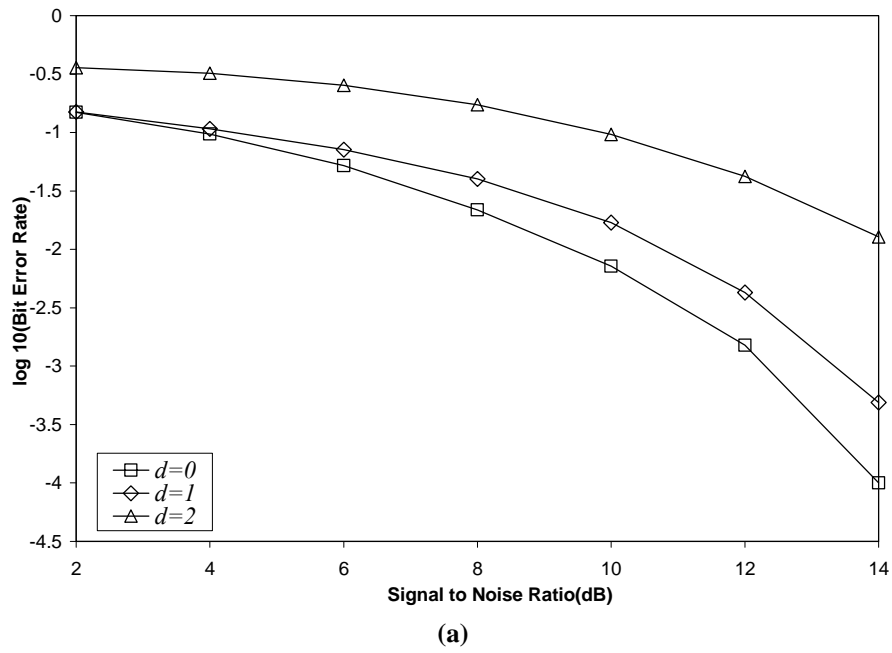


Figure 3.22: Optimal BER performance comparisons of Asymmetric Channels varying delay values (a)  $H_1(z)$  with  $m=2$  and (b)  $H_4(z)$  with  $m=2$ .



**Example 3:**

The next example of a four-tap asymmetric channel is defined by

$$H_9(z) = 0.7255 + 0.584 z^{-1} + 0.3627 z^{-2} + 0.0724 z^{-3}$$

Figure 3.23 illustrates that for a feedforward order  $m=4$ , the column span is  $ncol = 7$ . The position of the tap coefficient with maximum amplitude  $tap_{max} = 1$  and the channel order is even i.e.,  $n_a=4$ . Hence, the values of optimal decision delay given by Equation (3.11) is

$$0 \leq d_{opt} \leq 2$$

where  $d_{min} = 0$  and  $d_{max} = 2$  as computed by Equation (3.14).

The BER performance of the equaliser, given in Figure 3.25a for different delay values shows that the optimal decision delay values are  $d_{opt} = 0, 1$  or  $2$  for a feedforward order  $m=4$ . The optimal decision delay values obtained from the mathematical expressions given in the new approach are exactly the same.

**Example 4:**

The next example of a five-tap asymmetric channel considered here is defined by

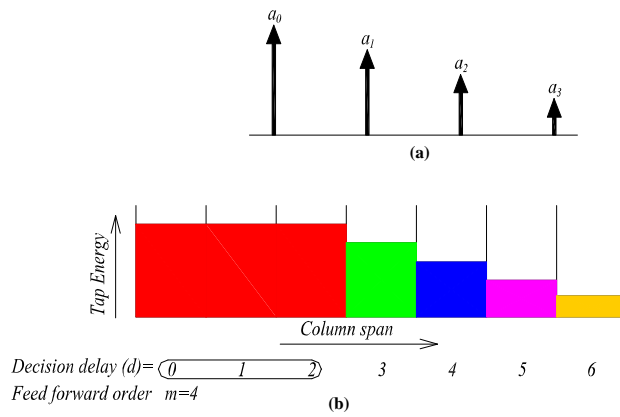
$$H_{11}(z) = -0.2052 - 0.5131z^{-1} + 0.7183 z^{-2} + 0.3695 z^{-3} + 0.2052 z^{-4}$$

As shown in Figure 3.24, for a feedforward order  $m=3$ , the column span is  $ncol = 7$ . The position of the tap coefficient with maximum amplitude  $tap_{max} = 3$  and the channel order is odd i.e.,  $n_a = 5$ . Hence the values of optimal decision delay given by Equation (3.11) is

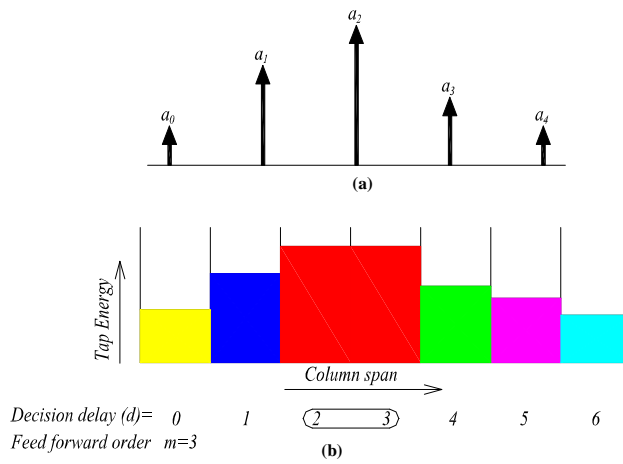
$$2 \leq d_{opt} \leq 3$$

where both  $d_{min} = 2$  and  $d_{max} = 3$  are obtained from Equation (3.14).

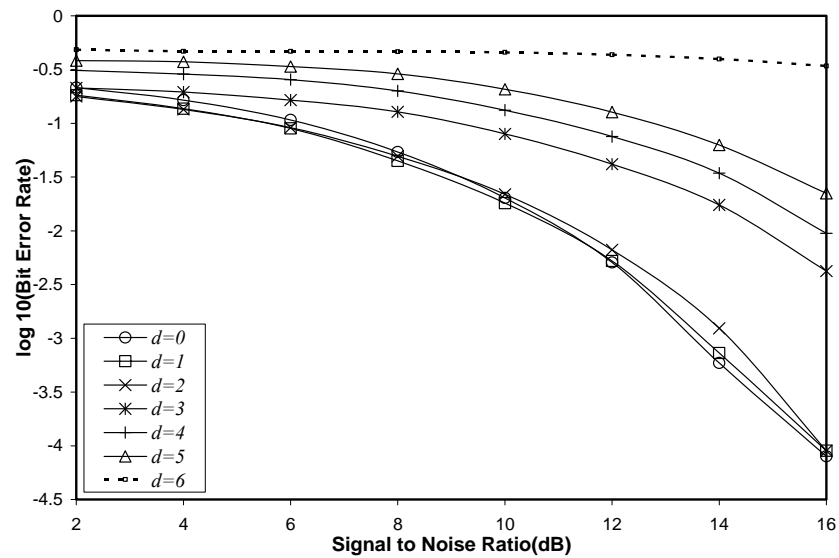
The BER performance of the equaliser, given in Figure 3.25b for different delay values shows that the optimal values of decision delay values are  $d_{opt} = 2$  or  $3$  for a feedforward order  $m=3$ , which are exactly identical with the values obtained from the mathematical expressions directly based on the new approach.



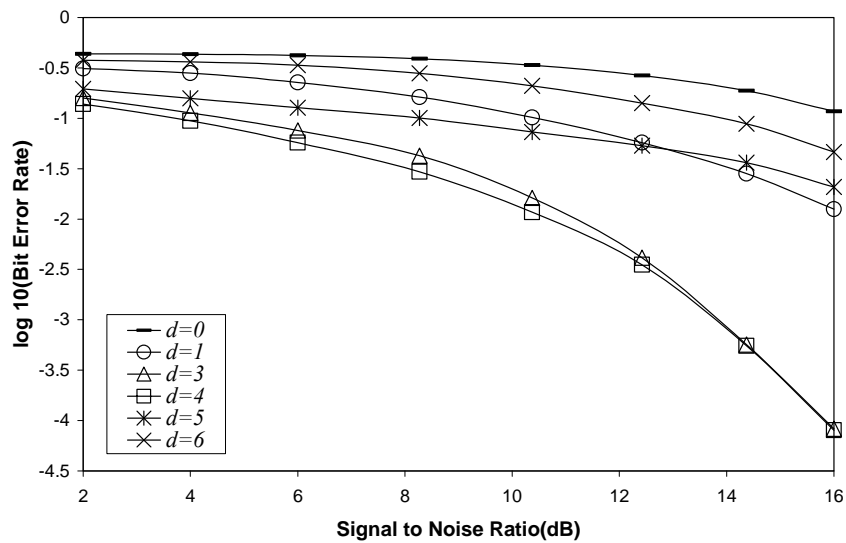
**Figure 3.23: Asymmetric Channel  $H_9(z)$**   
**(a) Impulse Response**  
**(b) Selection of optimal decision delay**



**Figure 3.24: Asymmetric Channel  $H_{11}(z)$**   
**(a) Impulse Response**  
**(b) Selection of optimal decision delay**



(a)



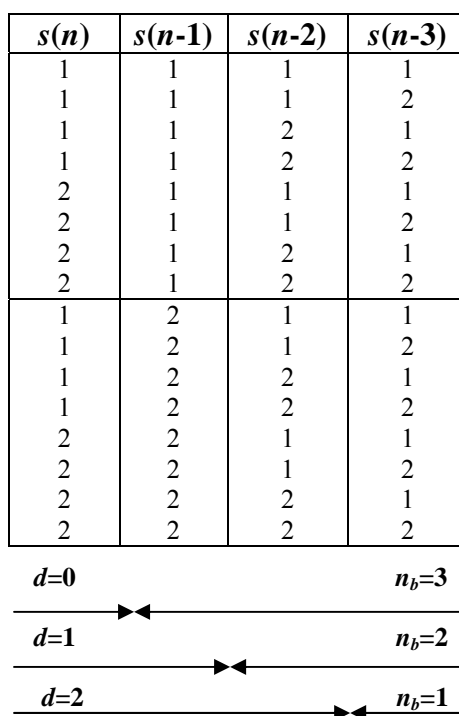
(b)

Figure 3.25: Optimal BER performance comparisons of Asymmetric Channels varying delay values (a)  $H_9(z)$  with  $m=4$  and (b)  $H_{11}(z)$  with  $m=3$ .

### 3.2.2 Equaliser with decision feedback

Attempt has also been made for selecting feedback order ' $n_b$ ' along with the decision delay parameter ' $d$ ' for equaliser structures incorporating decision feedback. The computer simulations for this study have prompted a new approach in determining the above parameters. Further it has been observed that inclusion of decision feedback makes the parameter selection procedure totally different from that without feedback structures. While in case of feedback type configurations, the column span ' $ncol$ ' of  $\mathbf{S}$  matrix plays a vital role in parameter selection procedure, the position of channel tap coefficient with maximum amplitude ' $tap_{max}$ ' is a key factor for decision delay selection for equalisers without decision feedback. In feedback type structures the value of ' $d$ ' and ' $n_b$ ' are related in such a way that they span the entire columns of  $\mathbf{S}$  matrix together as demonstrated in Figure 3.26. This aspect is further mathematically expressed as

$$ncol = d + n_b + 1 \tag{3.16}$$



**Figure 3.26 : Channel transmitted symbol sequence matrix ( $\mathbf{S}$ ) illustrating the relationship between  $d$  and  $n_b$ ,  $m=2$  (Channel:  $H_6(z)=0.4084+0.8164z^{-1}+0.4084z^{-2}$ )**

It is also found that the BER performance based on a given value of ' $n_b$ ' and the computed value ' $d$ ' using the above expression always yields a better result than any other values of ' $d$ '. A methodology has been evolved taking this concept into account to find the optimum values of ' $n_{b(opt)}$ ' and ' $d_{opt}$ '. Various BER plots for

different channels both symmetric and asymmetric types are already given in Figure 3.16 for different ' $n_b$ ' and ' $d$ ' combinations. The conclusion derived from these plots forms the basis of framing certain empirical relations between the design parameters ' $m$ ', ' $n_b$ ' and ' $d$ ' for a given channel order ' $n_a$ '. The analysis based on these plots has resulted in an interesting finding, where the optimum length of the feedback vector ' $n_{b(opt)}$ ' is always related to the channel order ' $n_a$ ' given by the following expression

$$n_{b(opt)} = n_a - 1 \quad (3.17)$$

Hence the feedback order ' $n_{b(opt)}$ ' for the condition of optimality is equal to ' $n_a - 1$ '. Once ' $n_{b(opt)}$ ' is evaluated as per Equation (3.17), the ' $d_{opt}$ ' value can now be computed using Equation (3.16) as

$$d_{opt} = ncol - n_{b(opt)} - 1 \quad (3.18)$$

The optimal combinations of parameters ' $n_{b(opt)}$ ' and ' $d_{opt}$ ' for a fixed feedforward order of equaliser for various channels using Equation (3.17) and Equation (3.18) based on the proposed approach are given in Table 3.2, which are exactly identical to the parameter values given in Table 3.3 obtained previously from BER performance plots .

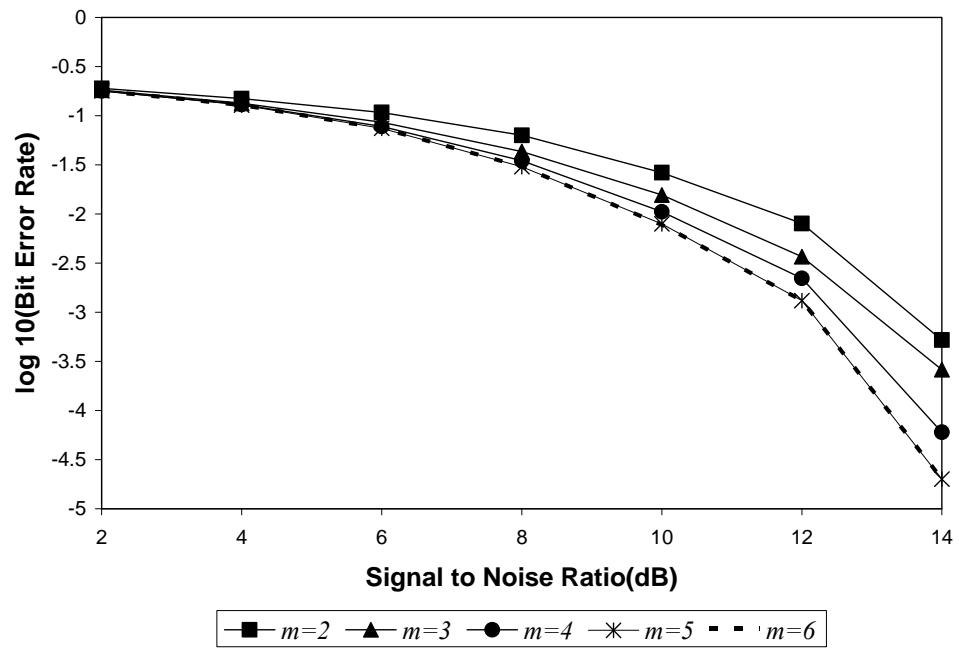
Channel transfer functions	Channel Order ( $n_a$ )	Column span ( $ncol$ )	Optimal values of equaliser parameters
$H_6(z) = 0.4084 + 0.8164z^{-1} + 0.4084z^{-2}$	3	5	$m=3, n_{b(opt)}=2$ and $d_{opt}=2$
$H_1(z) = 1 + 0.5z^{-1}$	2	3	$m=2, n_{b(opt)}=1$ and $d_{opt}=1$
$H_5(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$	3	5	$m=3, n_{b(opt)}=2$ and $d_{opt}=2$
$H_{10}(z) = 0.35 + 0.8z^{-1} + z^{-2} + 0.8z^{-3}$	4	5	$m=2, n_{b(opt)}=3$ and $d_{opt}=1$
$H_{11}(z) = -0.2052 - 0.5131z^{-1} + 0.7183z^{-2} + 0.3695z^{-3} + 0.2052z^{-4}$	5	9	$m=5, n_{b(opt)}=4$ and $d_{opt}=4$

**Table 3.3: Optimal values of parameters for equalisers (with decision feedback)**

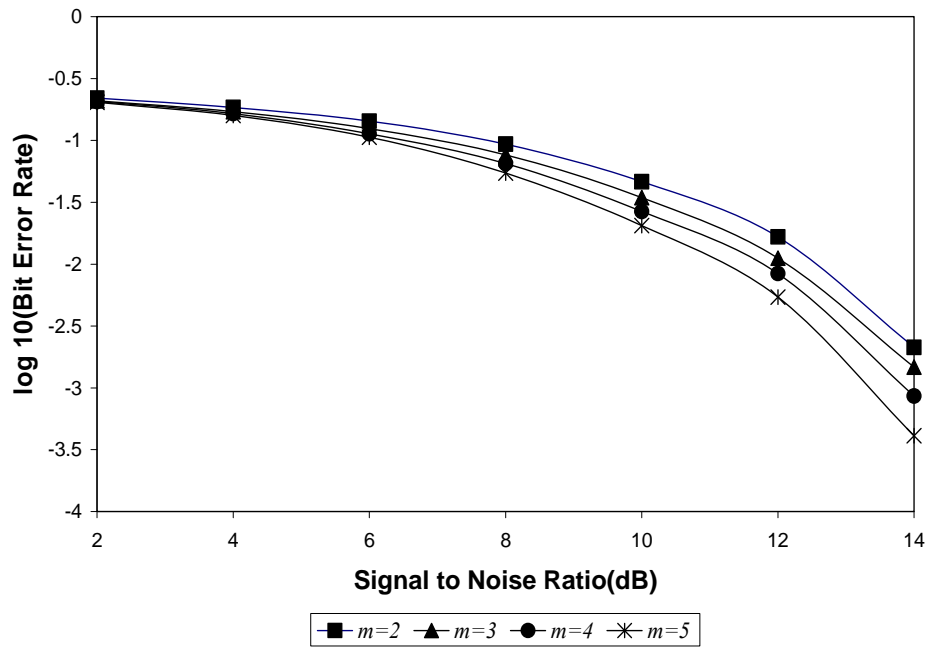
### 3.3 Selection of feedforward order ' $m$ '

All the discussions in the previous sections are based on a fixed feedforward order ' $m$ '. It is observed from the Figures 3.27a-b that for equalisers without feedback and with feedback, increasing the feedforward order ' $m$ ', the BER performance improves. Basically optimal equaliser design may need a large feedforward order. However, increasing the order of ' $m$ ' to a higher value elevates the structure size, thus computational complexity and hardware cost further rises. Also it is observed in Figure 3.27c-d that increasing the feedforward order of ' $m$ ' of the equaliser, convergence in the BER performance characteristic is achieved.

So far as this research work is concerned, the very point of emphasis is laid on selecting ' $m$ ' for designing a reduced structural configuration. And if this criterion is to be satisfied then the only possibility lies with picking up a small value for the feedforward order ' $m$ '. It is found that if a lower order of ' $m$ ' is chosen, some performance loss is accounted for with respect to BER performance. But this small sacrifice is made purposefully in order to gain structural advantage in design of equalisers, which is the prime objective of this research work. Thus a trade off between structural complexity and performance loss has been well taken care of in choosing the feedforward order ' $m$ '. Considering this aspect, the feedforward order ' $m$ ' has been restricted to the channel order ' $n_a$ ' such that the BER performance loss is not appreciable.



(a)



(b)

Figure 3.27 : Optimal BER performance comparisons varying equaliser's feedforward order( $m$ ) (a) Channel  $H_5(z)$ , (b) Channel  $H_6(z)$

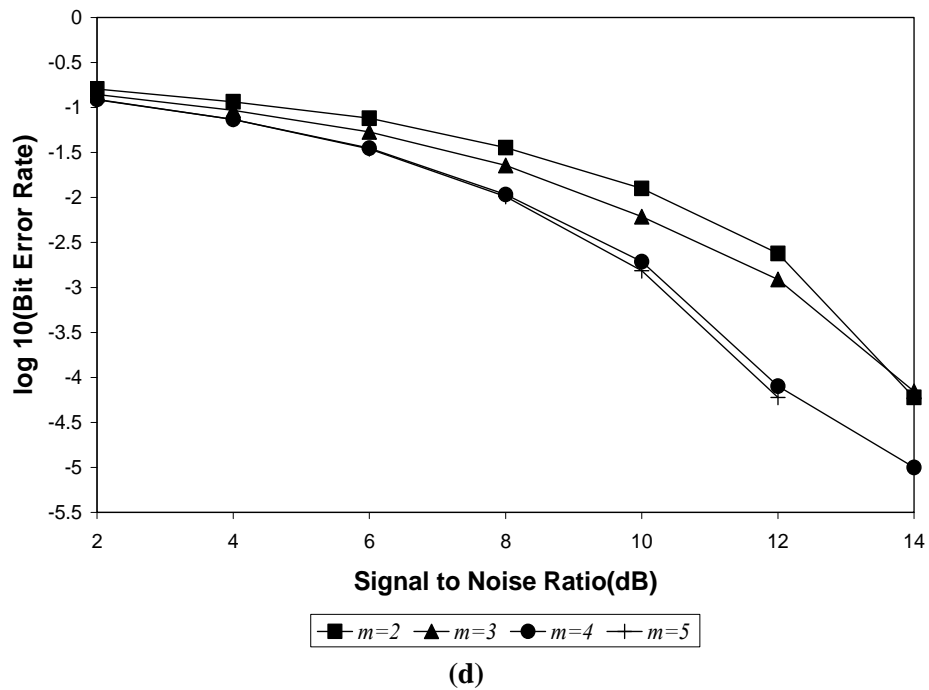
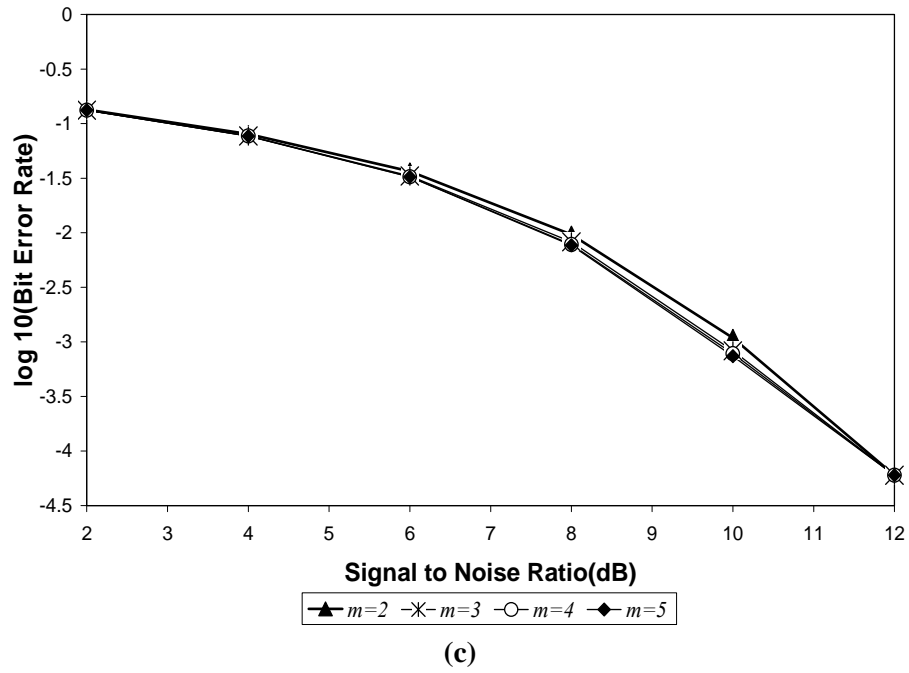


Figure 3.27 : Optimal BER performance comparisons varying equaliser's feedforward order ( $m$ ) (c) Channel  $H_1(z)$  and (d) Channel  $H_{11}(z)$



The restriction imposed on for selection of equaliser's feedforward order ' $m$ ', can be further justified from the simulation result given below. It is observed that if the change in the additive noise level is fixed, then the BER performance loss occurred is more predominant in case of equalisers with higher ' $m$ ' compared to those with lower ' $m$ '. This study is illustrated in Figure 3.28. For an increase of 2 dB in SNR value (BA), the BER performance loss in case of equaliser with  $m=3$  and  $m=5$  are CD and EF respectively, which justifies clearly another additional advantage gained in choosing a lower order feedforward order ' $m$ ' for equaliser design.

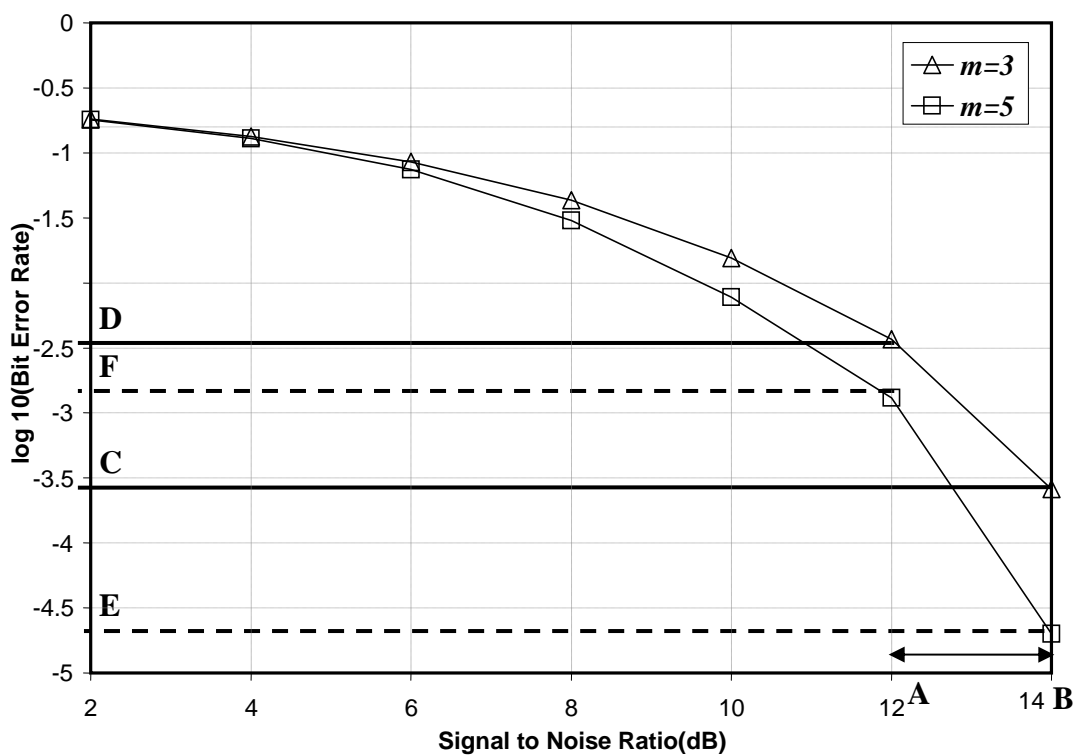


Figure 3.28: Effect of increasing equaliser's feedforward order on BER performance loss incurred for a given SNR change (Channel:  $H_5(z)$ )

### **3.4 Conclusion**

In the present work, a detail study on various factors influencing the BER performance of Bayesian equaliser has been undertaken as it provides the optimum performance for symbol-by-symbol type of equalisers. It is observed here that a Bayesian equaliser's BER performance is greatly influenced by additive noise level, decision delay ' $d$ ', feedforward order ' $m$ ' and feedback order ' $n_b$ '. It is inferred that both the additive noise level and ISI factor in a channel influence the equaliser's BER performance only for realistic SNR levels. The decision delay parameter decides the location of channel states with respect to the optimal decision boundary in the input observation space of the equaliser. In case of channels with coincident states, appreciable gain in BER performance is only achieved by incorporating decision feedback technique. Reduction in the number of noise-free channel states entirely depends on feedback order ' $n_b$ ' in a DFE structure. Selection of right combination of feedback order ' $n_b$ ' and decision delay ' $d$ ' for a fixed feedforward order ' $m$ ' in DFE structure plays a major role in obtaining the optimal performance. Further, this research work provides an insight in determining the optimal values of its various parameters from a different perspective while comparing with the previous works. The parameters for optimizing the BER performance for equalisers (without feedback and with decision feedback) can easily be evaluated only by a simple examination of the type of channel model and its tap coefficients. This methodology has really opened up a new horizon for parameter selection problem in an efficient way, without going for exhaustive graphical study. The approach suggested in this present work is conceptually different from that suggested by previous researchers [29] and has been logically interpreted from various simulation results. Hence this aspect of present study brings a new dimension to parameter selection issue in decision feedback based equaliser structures.

---

## CHAPTER 4

# Proposed FNN Based Equalisers

---

It has already been established that multilayer feedforward neural network (FNN) based equalisers have significant performance improvement over the conventional linear equalisers based on LMS or RLS adaptive algorithms [13,18,88,89]. The basic objective of the present work is primarily aimed at developing new equalisers in the FNN domain having reduced structural complexity so that these can be easily implemented in real-time. The learning algorithms developed to update the weights of different structures utilise the basic BP algorithm concept. However, with reference to the structural paradigm certain modifications have been included in the estimation of local gradient of errors at different nodes. In addition to this, a novel technique has been presented for adapting the slope parameter of the sigmoidal activation function using fuzzy controller approach for all the nodes of a conventional multilayer FNN structure without any structural modifications.

The major contribution of this chapter comprises of designing efficient equaliser structures based on feedforward neural network (FNN) topology and development of appropriate training algorithms to adapt the network weights faster. In Section 4.1, an elaborate explanation has been provided for an hierarchical knowledge reinforced feedforward neural network equaliser. Thereafter Section 4.2 deals with another variant in the FNN configuration, where an orthogonal basis function expansion technique has been employed to develop a new structure. Subsequently in Section 4.3, an analysis is carried out for a hybrid transform domain FNN equaliser structure. In Section 4.4, a novel fuzzy controller concept is introduced to tune the slope of the sigmoidal activation function of a conventional FNN based equaliser. Simulation study and the BER performance comparisons of all the proposed equalisers with reference to a conventional FNN one are presented in Section 4.5. Finally, Section 4.6 provides the summary of this chapter.

## 4.1 Hierarchical knowledge based feedforward neural network (HKFNN) equaliser

In this configuration the concept of *hierarchical knowledge* has been incorporated into an existing multilayer feedforward adaptive equaliser to augment the decisions at the nodes (expert's opinion)[97]. The expert's opinion is based on the outputs from the nodes of the previous layers and the original information (external inputs) fed to the structure. This new technique can be viewed as an enhancement in the knowledge base required for arriving at a pseudo-optimal solution. The order of cascading of nodes (experts) depends entirely upon the channel characteristics and hence the structural complexity directly relates to the problem under consideration. Thus more and more refined and matured information need to be processed from a raw database in order to correctly update the domain of knowledge for optimally characterizing a system.

The proposed neural structure shown in Figure 4.1 has only one neuron (expert) in each layer. Except the first layer, the knowledge base at other layers is strengthened by the expert opinions of the previous layers as well as the original input information. This realization is quite different from the conventional FNN, where the decision at a node solely relies upon the information generated at the nodes of the previous layer. This can be attributed to the fact that while the decision in case of conventional structure may be a biased one, the proposed method provides a well defined rational approach as it follows the basic concept of laws of natural justice. To be more specific, it can be stated that before delivering the final decision, each node (expert) analyses thoroughly the original information (external inputs to the network) along with the filtered knowledge generated by the nodes of the previous layers with an objective to eliminate any possible uncertainty that might have influenced it. The structure has been so chosen that the nodes responsible for the final decision are more reinforced with knowledge compared to the nodes at the intermediate layers. In terms of hierarchy, these nodes (experts) occupy the highest level in the process of decision making and hence these experts be fed as much information as possible to enrich their knowledge base in order to arrive at a near optimal solution. The proposed neural equaliser has to be trained first before its performance in terms of bit-error-rate is evaluated and the training algorithm is discussed in the following section.

### 4.1.1 Learning algorithm

The supervised training of the neural structure is adopted, through an error correction learning rule based on the Back Propagation algorithm, which is a generalisation of the LMS optimisation procedure. The adaptation of the neural network's synaptic weights is carried out by the method of gradient descent in the weight space which is implemented by propagating the error back from the output layer towards the input.

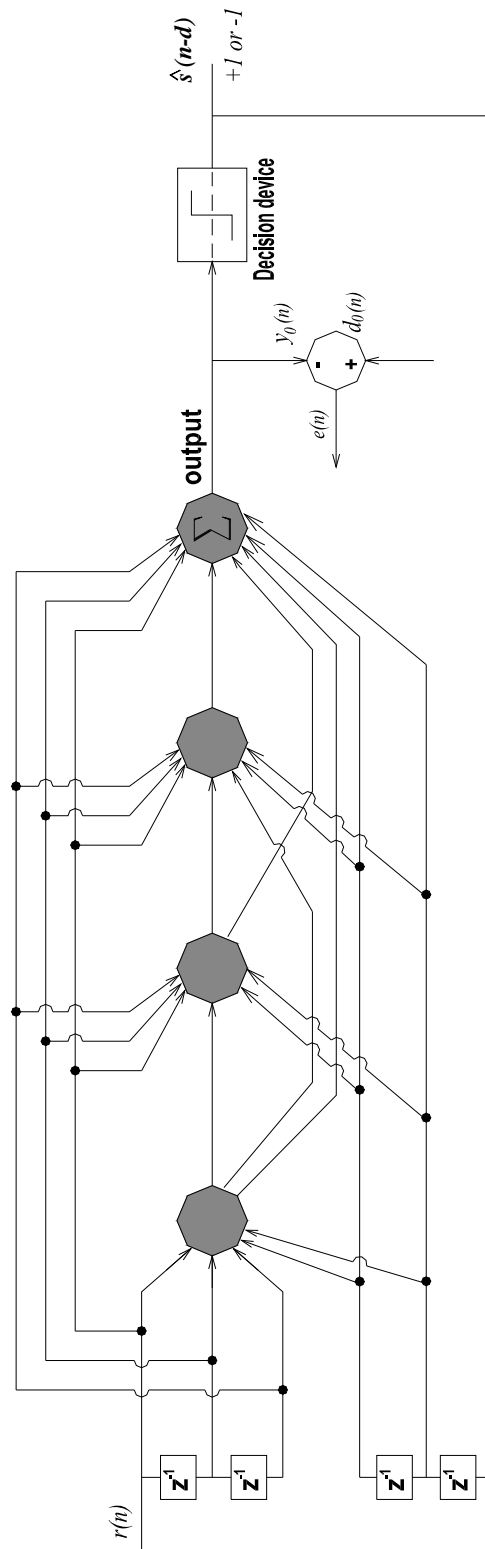


Figure 4.1: Hierarchical knowledge based feedforward neural equaliser structure

The various notations used in the development of algorithm are as follows:

$j$  is the layer index for  $1 \leq j \leq nn$ .  $nx$  is the number of inputs to the equaliser structure.

$\mathbf{V}$  is the weight matrix connecting inputs to the neurons of different layers.

$\mathbf{W}$  is the weight matrix connecting the neurons of different layers.

The learning algorithm comprises of the following steps for updating the connection weights sequentially.

- All the synaptic weights and thresholds are initialized to small random values chosen from a uniform distribution.
- The network is then presented with training samples and for each sample, the following sequences of forward and backward computations are carried out as given hereunder.
- The signal at the input layer of the proposed equaliser can be represented by a  $(m+n_b) \times 1$  vector as

$$\mathbf{x}(n) = [r(n), r(n-1), \dots, r(n-m+1); \hat{s}(n-d-1) \dots \dots, \hat{s}(n-d-n_b)]^T \quad (4.1)$$

- With a *signal vector*  $\mathbf{x}(n)$  applied to the input layer, the activation potentials and function signals of the network by proceeding forward through the network, layer by layer are computed.

The net internal activity level  $c_j(n)$  for a neuron in layer  $j$  is

$$c_j(n) = \sum_{k=1}^{nx} v_{kj}(n).x_k(n) + th_j(n), \quad j = 1 \text{ and } nx = m + n_b \quad (4.2)$$

$$= \sum_{k=1}^{nx} v_{kj}(n).x_k(n) + \sum_{i=1}^{j-1} w_{ij}(n).y_i(n) + th_j(n), \quad 1 < j \leq nn \quad (4.3)$$

Considering sigmoidal activation function, the output of the neuron in  $j^{\text{th}}$  layer is calculated as

$$y_j(n) = F\{c_j(n)\} = \frac{1 - e^{-\phi.c_j(n)}}{1 + e^{-\phi.c_j(n)}}, \quad 1 \leq j < nn \quad (4.4)$$

The final output from the equaliser is computed considering the node in the output layer to be a summing unit as given by

$$y_j(n) = c_j(n), \quad j = nn \quad (4.5)$$

- The output of the neural network is then compared with the desired value and consequently an error signal is produced.

$$e_j(n) = d_o(n) - y_j(n) \quad (4.6)$$

The error signal  $e_j(n)$  actuates a control mechanism, the purpose of which is to apply corrective adjustments to the synaptic weights. This objective is achieved by minimizing a cost function or index of performance,  $J(n)$ . A commonly used cost function based on the mean-squared-error criterion has been applied here.

$$J(n) = \frac{1}{2} \sum_j e_j^2(n) \quad (4.7)$$

- The local gradients (error terms) at each node of the proposed neural network can be computed by propagating the output error backward, layer by layer through all the connection weights towards the input.

$$\delta_j(n) = e_j(n), \quad j = nn \text{ ( for the output layer)} \quad (4.8)$$

$$\delta_j(n) = \{1 - y_j(n)\} \cdot (\phi / 2) \cdot \sum_{i=j+1}^{nn} \delta_i(n) \cdot v_{ji}(n), \quad 1 \leq j < nn \quad (4.9)$$

- The BP learning algorithm updates the network weights that minimize the cost function. The training algorithm applies a weight correction matrix, which is proportional to the instantaneous gradient of the cost surface w.r.t. the neural network synaptic weight vectors. The synaptic weights of the network of a layer are adjusted according to the generalised delta rule as given below.

$$\begin{aligned} v_{kj}(n+1) &= v_{kj}(n) + \Delta v_{kj}(n) \\ &= v_{kj}(n) - \eta \cdot \frac{\partial J(n)}{\partial v_{kj}(n)}, \quad 1 \leq j \leq nn \text{ and } 1 \leq k \leq nx \end{aligned} \quad (4.10)$$

$$w_{ji}(n+1) = w_{ji}(n) - \eta \cdot \frac{\partial J(n)}{\partial w_{ji}(n)}, \quad 1 \leq j < nn \text{ and } j+1 \leq i \leq nn \quad (4.11)$$

These equations can be explicitly expressed as

$$v_{kj}(n+1) = v_{kj}(n) + \alpha[v_{kj}(n) - v_{kj}(n-1)] + \eta \delta_j(n) x_k(n), \quad 1 \leq k \leq nx \quad (4.12)$$

$$w_{ji}(n+1) = w_{ji}(n) + \alpha[w_{ji}(n) - w_{ji}(n-1)] + \eta \delta_j(n) y_j(n), \quad j+1 \leq i \leq nn \quad (4.13)$$

$$th_j(n+1) = th_j(n) + \beta \delta_j(n), \quad 1 \leq j < nn \quad (4.14)$$

where  $\eta$  and  $\beta$  are the *learning-rate parameters* used in adaptation of weights and thresholds respectively and  $\alpha$  is the *momentum constant*.

- Once a new set of weights are evaluated, the response of the network is again computed using this set of weights and by presenting a new set of samples to the network. This process of recursion continues till the objective function is minimized.

## 4.2 Orthogonal basis function based feedforward neural network (OBFNN) equaliser

The proposed neural equaliser structure is based on an *orthogonal basis function* (OBF) expansion technique, motivated by genetic evolutionary concept, which utilises a self-breeding approach [98]. A natural concept underlying the principle of genetic transformation is applied to evolve new information so as to consolidate the final decision of a structure, which correlates the individual opinions of the experts of independent generations. The equaliser structure developed utilising this novel idea has outperformed the conventional multilayer FNN equaliser by a wide margin. It has also a reduced structural complexity and has the potential to become a challenging candidate for real-time implementation issue. The schematic diagram of the proposed neural equaliser structure is shown in Figure 4.2.

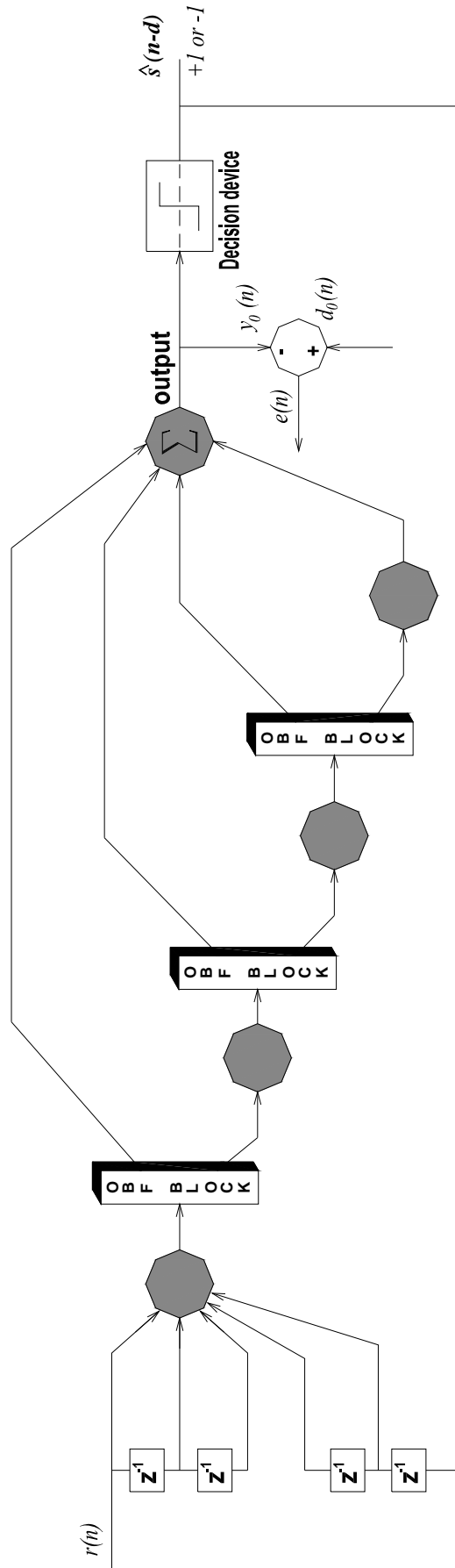


Figure 4.2: Orthogonal basis function based feedforward neural equaliser structure



This structure resembles to that of a multilayer FNN with the exception that each layer comprises of only one neuron and is based on *orthogonal basis function* expansion technique. The basic idea revolves around decomposing the signal 'y' at a node, into two orthogonal pairs as ' $y \cos y$ ' and ' $y \sin y$ ' such that the energy contained by transformed signals remains unaltered. Comparing this technique with self-breeding genetic approach, these two generated signals may be termed as *offsprings* of the present generation 'y' which take part in reproducing further to create a new generation. In this process out of these two *offsprings* only one is allowed to mutate and reproduce further in order to create a new generation, while the other one is constrained to remain as it is to preserve the knowledge of the corresponding generation which it forwards to the output node (expert) responsible for taking the final decision. The process of evolution continues for a number of generations depending upon the complexity of the problem under investigation. The prime objective focussed here is to develop a strategy, where a collective judgment based on the expert opinions evolved from decisions of individual generations can be employed to achieve a more rational and heuristic solution at the equaliser output. The proposed neural equaliser structure is adapted during the learning phase and then its performance is evaluated.

#### 4.2.1 Development of the concept for weight adaptation in the proposed structure

Basically this structure belongs to a class of multilayer FNN and hence the conventional BP algorithm can be applied to adapt the network weights. However, this algorithm needs to be suitably modified to take into account the error propagated through the *orthogonal basis function* (OBF) block.

According to the back propagation algorithm the evaluation of local gradient at each node is of prime importance for updating synaptic weights connected to the corresponding unit. Calculation of local gradients at each node cannot be directly computed using BP algorithm, as the OBF block is positioned in between the neurons of different layers. Hence, some suitable measures must be evolved to overcome this bottleneck. This can be achieved by modifying the BP algorithm to accommodate such limitations.

#### **Example :**

An example of the proposed structure and its expanded view are shown in Figure 4.2.1 and Figure 4.2.2 respectively. It is visualized that two parallel blocks can substitute an OBF block because a pair of orthogonal signals originates from it. Each block comprises of a function block and a multiplier unit. Details of error propagation and mathematical derivation of local gradients at each node are explained as follows.

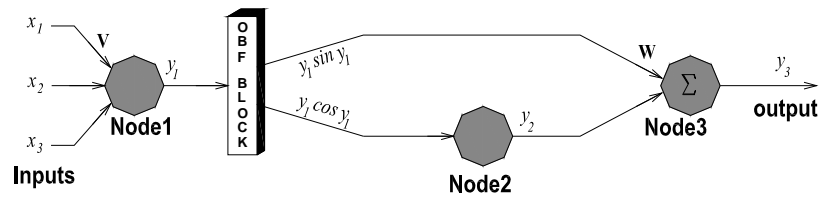


Figure 4.2.1: An example of proposed OBFNN structure

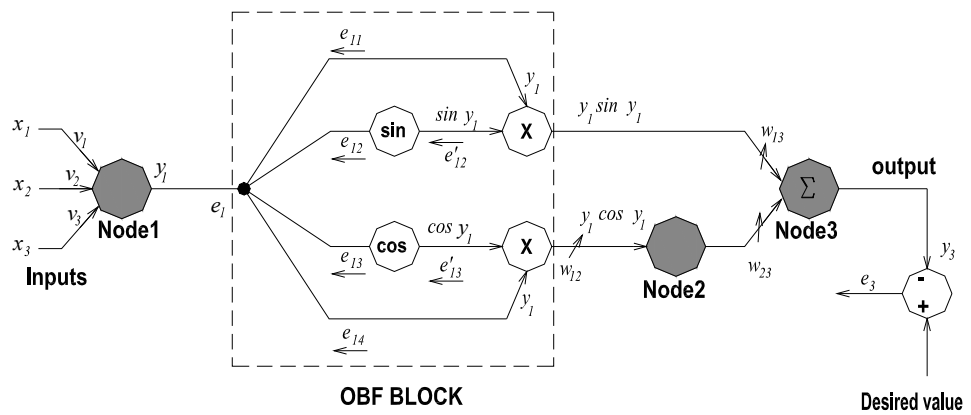


Figure 4.2.2: Expanded view of OBFNN structure

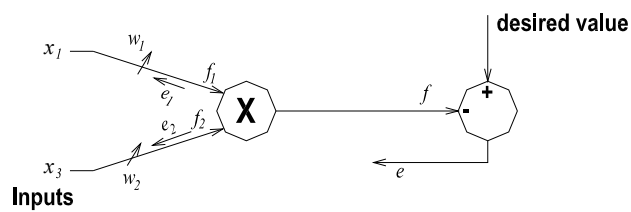


Figure 4.2.3: Error distribution through a multiplier unit

The error at the output node 3 is computed as

$$e_3(n) = d_o(n) - y_3(n) \quad (4.15)$$

Further, as this node is a pure summing unit, the error term is given by,

$$\delta_3(n) = e_3(n) \quad (4.16)$$

The error propagated back to node 2 through the connection weight  $w_{23}(n)$  is given by

$$e_2(n) = \delta_3(n) \cdot w_{23}(n) \quad (4.17)$$

and corresponding local gradient at node 2

$$\delta_2(n) = e_2(n) \{1 - y_2(n)^2\} (\phi/2) \quad (4.18)$$

The errors, propagated through the connection weights  $w_{12}(n)$  and  $w_{13}(n)$ , are  $e_{1C}(n)$  and  $e_{1S}(n)$  respectively and are computed as given below.

$$e_{1C}(n) = \delta_2(n) \cdot w_{12}(n) \quad (4.19)$$

and

$$e_{1S}(n) = \delta_3(n) \cdot w_{13}(n) \quad (4.20)$$

Further, a major problem is encountered in dividing  $e_{1C}(n)$  between the two input connection paths at the multiplier node during error propagation backward. Hence a new strategy is applied in distributing this error between these two paths.

With reference to Figure 4.2.3, the proposed concept for error distribution is explained below more elaborately. For sake of clarity the time index ‘ $n$ ’ is dropped in all the derivations given below. Using the basic descent gradient approach, the change in the network weights are evaluated as.

$$\Delta w_1 = -\eta \frac{\partial J}{\partial w_1} \quad (4.21)$$

$$\Delta w_2 = -\eta \frac{\partial J}{\partial w_2} \quad (4.22)$$

where  $J$  and  $\eta$  denote the objective function and learning parameter respectively.

Following the chain rule, the two terms  $\frac{\partial J}{\partial w_1}$  and  $\frac{\partial J}{\partial w_2}$  are expressed as the product of partial derivatives, which can be computed individually as below.

$$\begin{aligned} \frac{\partial J}{\partial w_1} &= \frac{\partial J}{\partial e} \cdot \frac{\partial e}{\partial f} \cdot \frac{\partial f}{\partial f_1} \cdot \frac{\partial f_1}{\partial w_1} \\ &= e (-1) f_2 x_1 \\ &= -e f_2 x_1 \end{aligned} \quad (4.23)$$

Hence,

$$\Delta w_1 = \eta e f_2 x_1 \quad (4.24)$$

Now, if  $e_1$  is assumed to be the error propagated through the weight  $w_1$  and  $x_1$  is the input connected to weight  $w_1$ , then using the descent gradient approach (LMS algorithm in Adaptive Filter Theory [68,69] ) the change in weight  $\Delta w_1$  is given by

$$\Delta w_1 = \eta e_1 x_1 \quad (4.25)$$

Now equating Equations (4.24) and (4.25), the error  $e_1$  is evaluated as

$$e_1 = e \cdot f_2 \quad (4.26)$$

In a similar manner the error  $e_2$  is calculated by

$$e_2 = e \cdot f_1 \quad (4.27)$$

Based on the explanation as mentioned above, the errors propagated back from the multiplier units are given by

$$e_{11}(n) = e_{1S}(n) \sin y_1(n) \quad (4.28)$$

$$e'_{12}(n) = e_{1S}(n) y_1(n) \quad (4.29)$$

$$e'_{13}(n) = e_{1C}(n) y_1(n) \quad (4.30)$$

$$e_{14}(n) = e_{1C}(n) \cos y_1(n) \quad (4.31)$$

The errors  $e'_{12}(n)$  and  $e'_{13}(n)$ , at the output of function block, are to be propagated back through it. Here, an adhoc solution has been devised by considering the function block, which performs a nonlinear mathematical operation to be equivalent to the sigmoidal nonlinearity of a neuron. Following this strategy, BP algorithm, can be utilised directly for error calculation at the input end of the function block. Thus the errors are computed as

$$e_{12}(n) = e'_{12}(n) \cos y_1(n) \quad (4.32)$$

and

$$e_{13}(n) = - e'_{13}(n) \sin y_1(n) \quad (4.33)$$

Finally, a rational approach is followed to evaluate the total error at the output of node 1,  $e_1(n)$  by combining all the errors  $e_{11}(n)$ ,  $e_{12}(n)$ ,  $e_{13}(n)$  and  $e_{14}(n)$ , as it has been contributed from four different paths connected to that node.

$$\begin{aligned} e_1(n) &= e_{1S}(n) \sin y_1(n) + e_{1S}(n) y_1 \cos y_1(n) + e_{1C}(n) \cos y_1(n) - e_{1C}(n) \cdot y_1 \sin y_1(n) \\ &= \delta_3(n) w_{13}(n) \{ \sin y_1(n) + y_1 \cos y_1(n) \} + \delta_2(n) w_{13}(n) \{ \cos y_1(n) - y_1 \sin y_1(n) \} \end{aligned} \quad (4.34)$$

Hence, the local gradient at node 1 is calculated as

$$\delta_1(n) = e_1(n) \{ 1 - y_1(n)^2 \} (\phi/2) \quad (4.35)$$

Thus all the synaptic weights connected to node 1 can be directly updated based on  $\delta_1(n)$ .

#### 4.2.2 Learning algorithm

The various notations used in the following algorithm are given as below:

$j$  is the layers index,  $1 \leq j \leq nn$

$nx$  is the number of input to the equaliser.

$\mathbf{V}$  is the weight matrix connecting inputs to the neurons of first layer.

$\mathbf{W}$  is the weight matrix connecting the neurons of different layers.

The generalised algorithm is summarised as below:

- The synaptic weights and thresholds are initialised to small random values which are uniformly distributed.
- The signal at the input layer of the proposed equaliser can be represented by a  $(m+n_b) \times 1$  vector as  $\mathbf{x}(n) = [r(n), r(n-1), \dots, r(n-m+1); \hat{s}(n-d-1), \dots, \hat{s}(n-d-n_b)]^T$ .
- Calculation of network output

The forward propagation of signal continues layer by layer till the final output  $y_o(n)$  of the neural structure is calculated. The internal activity of the node at the first layer is given by

$$c_j(n) = \sum_{k=1}^{nx} x_k(n) \cdot v_k(n) + th_j(n), \quad nx = m + n_b \quad (4.36)$$

For all the nodes in the middle layer, the internal activity of each node is calculated as

$$c_j(n) = y_i \cos y_i(n) \cdot w_{ij}(n) + th_j(n), \quad 1 < j < nn \text{ and } i = j - 1 \quad (4.37)$$

The local expert decisions at the nodes of all layers except the output are decided by the sigmoidal nonlinearity as given by

$$y_j(n) = F\{c_j(n)\} = \frac{1 - e^{-\phi \cdot c_j(n)}}{1 + e^{-\phi \cdot c_j(n)}}, \quad 1 \leq j < nn \quad (4.38)$$

The node at the output layer combines the knowledge available from all generations (weighted sum of all the signals) to generate the final output.

$$y_j(n) = y_i(n) w_{ij}(n) + \sum_{k=1}^{nn-2} y_k \sin y_k(n) \cdot w_{k nn}(n), \quad j = nn \text{ and } i = j - 1 \quad (4.39)$$

- Computation of error terms:

As the final node is a summing unit, the error term at time index  $n$  is computed by comparing the output with the desired value.

$$\delta_{nn}(n) = e_j(n) = d_o(n) - y_j(n), \quad j = nn \quad (4.40)$$

For the node in  $(nn - 1)^{\text{th}}$  layer, the error term is calculated as

$$\delta_j(n) = \delta_l(n) w_{jl}(n) \{1 - y_j(n)^2\} (\phi/2), \quad j = nn - 1 \text{ and } l = j + 1 \quad (4.41)$$

For the nodes in all the middle layers, the error terms are calculated as

$$\delta_j(n) = \{\delta_l(n) w_{jl}(n) (\cos y_j(n) - y_j \sin y_j(n)) + \delta_{mn}(n) w_{j mn}(n) (\sin y_j(n) + y_j \cos y_j(n))\} \{1 - y_j(n)^2\} (\phi/2), \quad 1 \leq j \leq nn - 2 \quad (4.42)$$

- Updation of synaptic weights and thresholds is carried out using the generalised delta rule as follows.

$$w_{j nn}(n+1) = w_{j nn}(n) + \eta \delta_{nn}(n) y_j \sin y_j(n) + \alpha \Delta w_{j nn}(n-1), \quad 1 \leq j \leq nn - 2 \quad (4.43)$$

$$w_{jl}(n+1) = w_{jl}(n) + \eta \delta_l(n) y_j \cos y_j(n) + \alpha \Delta w_{jl}(n-1), \quad 1 \leq j \leq nn - 2 \quad (4.44)$$

$$w_{jl}(n+1) = w_{jl}(n) + \eta \delta_{mn}(n) y_j(n) + \alpha \Delta w_{jl}(n-1), \quad j = nn - 1 \quad (4.45)$$

$$v_k(n+1) = v_k(n) + \eta \delta_l(n) x_k(n) + \alpha \Delta v_k(n-1), \quad 1 \leq k \leq nx \quad (4.46)$$

$$th_j(n+1) = th_j(n) + \beta \delta_j(n), \quad 1 \leq j < nn \quad (4.47)$$

where  $\eta$  and  $\beta$  are the *learning-rate parameters* of weights and thresholds respectively and  $\alpha$  is the *momentum constant*.

- This process of recursion continues till a given performance index is achieved.

### 4.3 Transform domain based feedforward neural network(TDFNN) equaliser

In this present work a hybrid configuration has been proposed where a real-valued discrete transform (RDT) block is embedded within the framework of a conventional FNN structure [99,100]. A signal vector is mapped from a given domain to another when fed to a transform block, because basically the transform block performs a fixed filtering operation. Almost an identical operation is carried out in a neuron block i.e. an FNN structure where signal transformation also takes place. But the basic difference between the transform block and the neural block is that while adaptive weights are associated with the later, fixed weights are inherent in the former. Hence, such a hybrid network representing a heterogeneous configuration has been proposed to solve the channel equalisation problem. Further, as this work is aimed at designing real-time implementable structure, RDT has been purposefully utilised. Though various transforms like *Discrete Cosine Transform(DCT)*, *Discrete Sine Transform(DST)* and *Discrete Hartley Transform(DHT)* [101] have been exhaustively tried initially in the simulation, the performance of DCT is found to be best amongst all. Thus, the entire analysis in this section is devoted to DCT.

The proposed equaliser given in Figure 4.3 is equivalent to a conventional multilayer feedforward neural structure with an exception that only one layer of an FNN structure is employed, followed by a discrete cosine transform block. This new idea has originated from the transform-domain adaptive filter theory [101,102]. A real-valued transform is a powerful signal decorrelator which performs whitening of the signal by causing the eigen value spread of an auto-correlation matrix to reduce. Further, these transformed signals undergo a power normalisation process [102], which speeds up the convergence of adaptive weights and also improves the performance significantly. The final output of the proposed structure is evaluated as the weighted sum of all normalised signals from the transform block. The BP algorithm is then applied to adapt the weights in the proposed hybrid structure with certain modifications incorporated based on the structural changes when compared with a conventional FNN structure, because the transform block placed at the output end poses difficulty in the propagation of error back through it. This problem is overcome by taking the *inverse discrete cosine transform (IDCT)* of the errors and computation of the local gradients of all the nodes of the FNN structure is done. The training algorithm for the proposed neural based equaliser structure has been described step by step as follows.

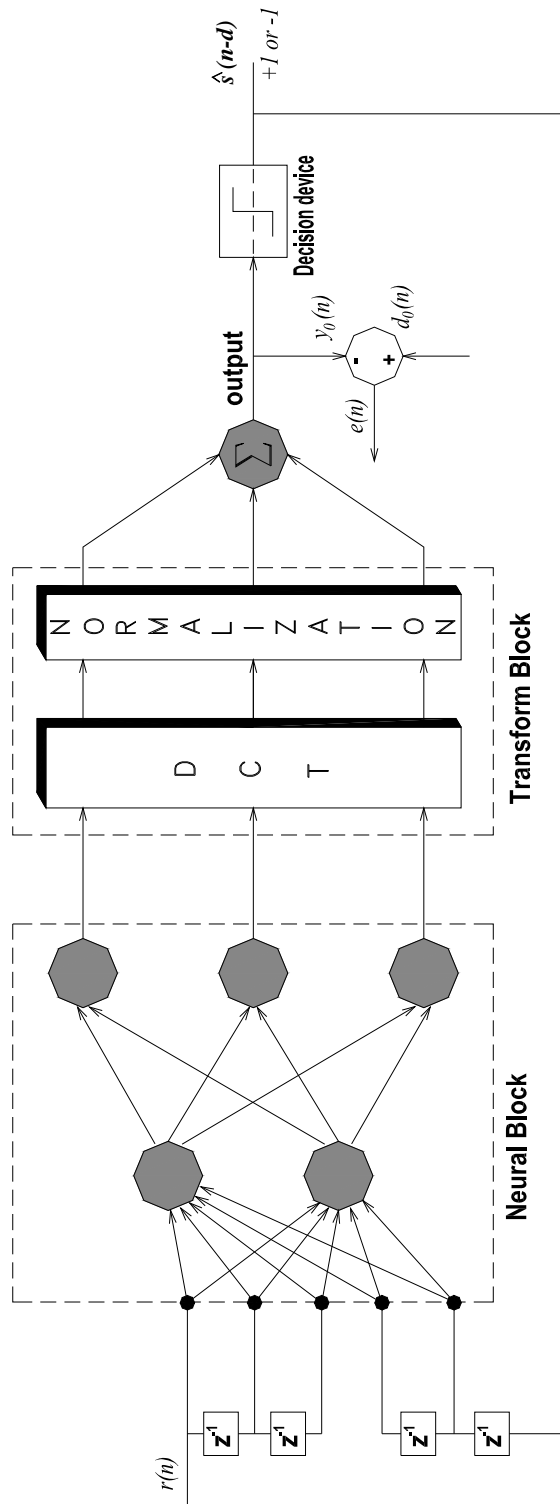


Figure 4.3: Transform domain based feedforward neural equaliser structure

### 4.3.1 Learning algorithm

The various notations used in the development of algorithm are as follows:

$nx$  is the number of inputs to the equaliser structure.

$ns$  is the number of neurons in the single layer FNN structure.

$nn$  is the output layer index.

$\mathbf{V}$  is the weight matrix connecting inputs to the neurons of first layer.

$\mathbf{W}$  is the weight matrix connecting transformed signals to the output unit.

The training algorithm is summarised as below:

- Initialisation of all the network weights and thresholds to small random values that are uniformly distributed.
- The signal at the input layer of the proposed equaliser can be represented by a  $(m+n_b) \times 1$  vector as  $\mathbf{x}(n) = [r(n), r(n-1), \dots, r(n-m+1); \hat{s}(n-d-1), \dots, \hat{s}(n-d-n_b)]^T$ .
- Calculation of network output:

The output of  $k^{th}$  neuron of input layer at time  $n$  is given by

$$y_k(n) = \frac{1 - e^{-\phi \cdot c_j(n)}}{1 + e^{-\phi \cdot c_j(n)}}, \quad 1 \leq k \leq ns \quad (4.48)$$

$$\text{where } c_j(n) = \sum_{i=1}^{nx} v_{ik}(n)x_i(n) + th_k(n) \quad (4.49)$$

The outputs from the single layer FNN form a signal vector  $\mathbf{y}(n)$ , which is forwarded to the DCT block for necessary processing.

The  $k^{th}$  element of the transformed signal is given by

$$y_{\mathcal{T}k}(n) = \mathcal{T}y_k(n) \quad (4.50)$$

The  $\mathcal{T}_{pq}^{th}$  element of the  $N \times N$  transform matrix  $\mathcal{T}$  is defined as

$$\mathcal{T}_{pq} = \begin{cases} \frac{1}{\sqrt{N}}, & p = 0; q = 0, 1, \dots, N-1 \\ \left( \frac{\sqrt{2}}{\sqrt{N}} \right) \cos \frac{\pi(2q+1)p}{2N}, & p = 1, 2, \dots, N-1; q = 0, 1, \dots, N-1 \end{cases} \quad (4.51)$$

Transformed signal  $y_{\mathcal{T}k}(n)$  is then normalised by the square root of their power  $\mathcal{B}_k(n)$  which can be estimated by filtering the  $y_{\mathcal{T}k}^2(n)$  with an exponentially decaying window of scaling parameter  $\gamma \in [0, 1]$  as derived in the literature [102] and shown below.

The  $k^{th}$  element of the normalized signal,

$$y_{\mathcal{N}k}(n) = \frac{y_{\mathcal{T}k}(n)}{\sqrt{\mathcal{B}_k(n) + \varepsilon}} \quad (4.52)$$

$$\mathcal{B}_k(n) = \gamma \mathcal{B}_k(n-1) + (1 - \gamma) y_{\mathcal{T}k}^2(n) \quad (4.53)$$



The small constant  $\varepsilon$  is introduced to avoid numerical instabilities when signal power  $\mathcal{B}_k(n)$  is close to zero. The resulting equal power signals from the normalized block are given to a summing unit, which produce the final output given by

$$y_j(n) = \sum_{k=1}^{ns} y_{\mathcal{N}k}(n) \cdot v_k(n) \quad (4.54)$$

- Calculation of error terms:

The error at the output  $e_j(n)$  is found by comparing the final output with the desired value of equaliser at time index  $n$ ,

$$e_j(n) = d_o(n) - y_j(n), \quad j = nn \quad (4.55)$$

Using the Back-Propagation algorithm, the output error is propagated backwards through all the connections layer by layer. First the error at the output of normalisation block is calculated as,

$$e_{\mathcal{M}k}(n) = e_j(n) \cdot w_k(n), \quad 1 \leq k \leq ns \quad (4.56)$$

The power normalisation can be considered as a process, whose operation is quite similar to the nonlinear transformation produced by sigmoid activation function of a neuron. Following this concept, the error terms (i.e., local gradients) at the output of the transform block can be calculated using the following equation.

$$\begin{aligned} \delta_{\mathcal{T}k}(n) &= e_{\mathcal{M}k}(n) \cdot \frac{\partial y_{\mathcal{N}k}(n)}{\partial y_{\mathcal{T}k}(n)} \\ &= e_{\mathcal{M}k}(n) \cdot \frac{\sqrt{\mathcal{B}_k(n) + \varepsilon} - \frac{\partial \sqrt{\mathcal{B}_k(n) + \varepsilon}}{\partial y_{\mathcal{T}k}(n)} y_{\mathcal{T}k}(n)}{\mathcal{B}_k(n) + \varepsilon} \\ &= e_{\mathcal{M}k}(n) \cdot \frac{\frac{y_{\mathcal{T}k}(n)}{y_{\mathcal{N}k}(n)} - y_{\mathcal{N}k}(n)(1-\gamma)y_{\mathcal{T}k}(n)}{\frac{y_{\mathcal{T}k}^2(n)}{y_{\mathcal{N}k}^2(n)}} \\ &= e_{\mathcal{M}k}(n) (y_{\mathcal{N}k}(n) / y_{\mathcal{T}k}(n)) \{1 - (1-\gamma) y_{\mathcal{T}k}^2(n)\} \end{aligned} \quad (4.57)$$

Once the errors are back propagated through the transform block, the errors at its input  $e_k(n)$  can be evaluated using the *Inverse Discrete Cosine Transform*(IDCT) operation [101].

$$e_k(n) = IDCT\{\delta_{\mathcal{T}k}(n)\} \quad (4.58)$$

Further, the local gradients (error terms) at the nodes of the input layer of the proposed structure are calculated accordingly.

$$\delta_k(n) = e_k(n) \{1 - y_k(n)^2\}(\phi/2), \quad 1 \leq k \leq ns \quad (4.59)$$

- Adaptation of network weights:

The synaptic weights and thresholds of the input layer and the output layer are updated using the generalised delta rule.

$$v_{ik}(n+1) = v_{ik}(n) + \eta \delta_k(n) x_i(n) + \alpha \Delta v_{ik}(n-1), \quad 1 \leq i \leq nx \text{ and } 1 \leq k \leq ns \quad (4.60)$$

$$th_k(n+1) = th_k(n) + \beta \delta_k(n), \quad 1 \leq k \leq ns \quad (4.61)$$

$$w_k(n+1) = w_k(n) + \eta e(n) y_{nk}(n), \quad 1 \leq k \leq ns \quad (4.62)$$

where  $\eta$  and  $\beta$  are the *learning-rate parameters* of weights and thresholds respectively and  $\alpha$  is the *momentum constant*.

- This recursion procedure is continued till the objective function is minimised.

#### 4.4 Fuzzy tuned feedforward neural network (FZTUNFNN) equaliser

In a neural network paradigm the synaptic weights and threshold values are generally considered as free parameters in conventional sense, which are adapted using appropriate learning algorithms in order to train the network. However there are many other parameters like slope of the sigmoidal activation function, learning-rate parameter for synaptic weights, thresholds and momentums etc., which can also be tuned to enhance the adaptability of the network. The slope of the sigmoidal activation function of each neuron in a multilayer neural network paradigm plays an important role and is a key parameter in the decision-making ability of that specific node. Performance of conventional MLP neural equaliser can be improved by tuning the slope of the activation function along with weight updation. In the present research work attempt has been made to develop a new neural structure by adapting the slope of the sigmoid activation function using fuzzy logic controller approach.

Fuzzy logic controller approach has proved to be a potential candidate in varieties of control applications recently. The development of fuzzy logic controllers has resulted in improved performance of dynamic systems in comparison to conventional PID controllers [103,104]. The most important feature of fuzzy logic controller is the application of linguistic information derived from the abstract values of the plant parameters to evaluate a control action. On the other hand, the PID controllers sometimes fail to deliver optimal performances and setting of the controller gain is highly sensitive to the types of disturbances experienced by the systems [105,106]. This background idea about controllers has motivated the research to develop a novel equaliser structure on a multilayer FNN framework. The proposed structure is a hybrid one because in this configuration, while the BP algorithm takes control by recursively updating the network weights and threshold values the fuzzy logic controller approach adjusts the slope of the sigmoid activation function of all the nodes of the network at the same time [107].

In this section, the entire focus of analysis is aimed at tuning the slope parameter of the sigmoidal activation function. As the proposed equaliser under investigation is a conventional FNN with a reduced structure, the BP algorithm described in Appendix A is applied to update the network connection weights. The BP algorithm provides an estimation of the local gradient of the neurons in each layer and taking these into consideration adaptation of the slope parameter  $\phi$  is done by the fuzzy logic controller approach. The proposed work is aimed at reduction of the training time as well as improvement in performance of conventional MLP neural equaliser. This entire process of sigmoid slope tuning is explained in detail in the following section.

#### 4.4.1 Description of the proposed concept of sigmoid slope tuning

For a fully trained multilayer neural network, the error term at the output node (termed as global error) is not only at the lowest level but the error term at each node except output node (termed as local error) of the network is at the lowest level too and ideally may be zero. Under such circumstances there will be no further change in the synaptic weights or the threshold values of the network. This confirms the basic concept embedded in the BP algorithm that the change in synaptic weights and thresholds is only possible if the error term ' $\delta_j^{(l)}(n)$ ' exists in the nodes at all, which can be verified from the mathematical equations governing the updation of the above parameters. The changes in the synaptic weights and thresholds of the network are expressed as

$$\Delta w_{ij}^{(l)}(n+1) = \eta \delta_j^{(l)}(n) y_j^{(l-1)}(n) + \alpha \Delta w_{ij}^{(l)}(n) \quad (4.63)$$

and

$$\Delta th_j^{(l)}(n+1) = \beta \delta_j^{(l)}(n) \quad (4.64)$$

If the above equations yield non-zero values, provided  $\delta_j^{(l)}(n)$  exists, then the network continues to remain in the learning mode, else the network is said to be fully trained. If the error term ' $\delta_j^{(L)}(n)$ ' in the output node (global error) is appreciably small, then by propagating it back into the network, will further result in almost negligible error terms (local errors) at hidden nodes of the network. Hence the error term at individual (neuron) of a neural structure is to be minimised to get a pseudo-optimal solution. The fuzzy logic controller technique is applied to determine the amount of correction needed for the slope of the sigmoidal activation function at each node of the network such that output of the nodes are changed and local errors are minimal. As a result the final output of the network can provide a near optimal solution faster with significant reduction of the training time.

**4.4.1.1 Description of the fuzzy logic controller technique**

A schematic block diagram of fuzzy logic controller is presented in Figure 4.4. Basically a fuzzy controller evaluates the change in the control action based on the information regarding error and rate of change of error at the process output. The same concept is adopted in the proposed work considering the error term and rate of change of error term at a node of the network as the two inputs to the fuzzy controller block. The fuzzy logic controller structure used here consists of five layers and is explained in detail in Appendix C.

The node error term is known as  $\delta_j^{(l)}(n)$  and its rate of change is described by

$$\Delta\delta_j^{(l)}(n) = \delta_j^{(l)}(n) - \delta_j^{(l)}(n-1) \tag{4.65}$$

are fed into the fuzzy controller block as shown in Figure 4.5. The output generated from the control block  $\Delta\phi(n)$ , as shown in Figure 4.5, is used to obtain the changed slope at the  $(n+1)^{th}$  time index  $\phi(n+1)$  of the sigmoidal activation function using the relation

$$\phi(n+1) = \phi(n) + \Delta\phi(n) \tag{4.66}$$

In the present investigation, seven categories of linguistic variables {Large Positive (LP), Medium Positive (MP), Small Positive (SP), Zero (ZE), Small Negative (SN), Medium Negative (MN) and Large Negative (LN)} are employed to describe both the input variables ( $\delta_j^{(l)}(n)$  and  $\Delta\delta_j^{(l)}(n)$ ) and the output  $\Delta\phi(n)$ . The membership functions of fuzzy controller structure are assumed to have Gaussian type distribution [108] and have fixed centres and widths. The fuzzified inputs are used to construct the rule base. Taking into account the linguistic information of the inputs and with *a-priori* knowledge about the bounds of the sigmoid slope variation, the controller output is decided. In order to reflect this concept a fuzzy rule base is constructed as given in Table 4.1.

The fuzzy control rules [109,110] are expressed in the form of ‘IF...THEN’ statements where the indexing terms  $l$  and  $j$  have been dropped for the sake of clarity. The interpretations of the fuzzy rules for sigmoidal slope adjustments are listed below.

- If  $\delta(n)$  is SN and  $\Delta\delta(n)$  is SP, then  $\Delta\phi(n)$  is ZE.
- If  $\delta(n)$  is SP and  $\Delta\delta(n)$  is LP, then  $\Delta\phi(n)$  is LP.
- .....
- .....
- .....
- If  $\delta(n)$  is MP and  $\Delta\delta(n)$  is LN, then  $\Delta\phi(n)$  is SN.

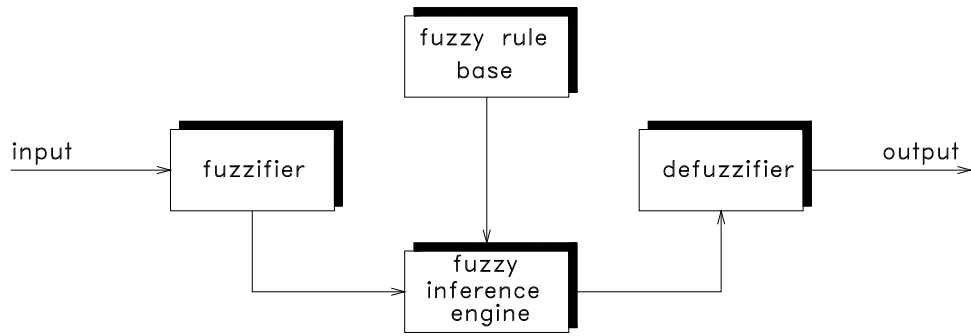


Figure 4.4: Schematic block diagram of fuzzy logic controller

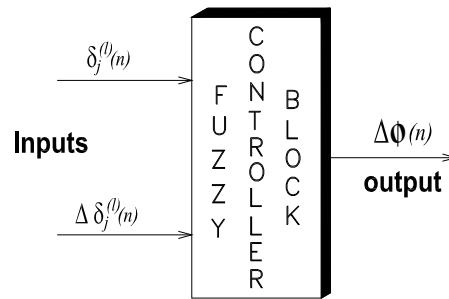


Figure 4.5: Fuzzy logic approach for tuning sigmoid slope

		$\Delta \delta(n)$		← Error term change →						
				LN	MN	SN	ZE	SP	MP	LP
Error term ↑ ↓	$\delta(n)$	LN	MN	SN	ZE	SP	MP	LP		
	LN	LN	LN	LN	LN	MN	SN	ZE		
	MN	LN	LN	LN	MN	SN	ZE	SP		
	SN	LN	LN	MN	SN	ZE	SP	MP		
	ZE	LN	MN	SN	ZE	SP	MP	LP		
	SP	MN	SN	ZE	SP	MP	LP	LP		
	MP	SN	ZE	SP	MP	LP	LP	LP		
	LP	ZE	SP	MP	LP	LP	LP	LP		

Table 4.1 Fuzzy Rule Table

**Example:**

A typical case study for the necessary slope correction of the sigmoid activation function of a given node in the network using the fuzzy logic controller approach is undertaken in this work and is explained here as given below.

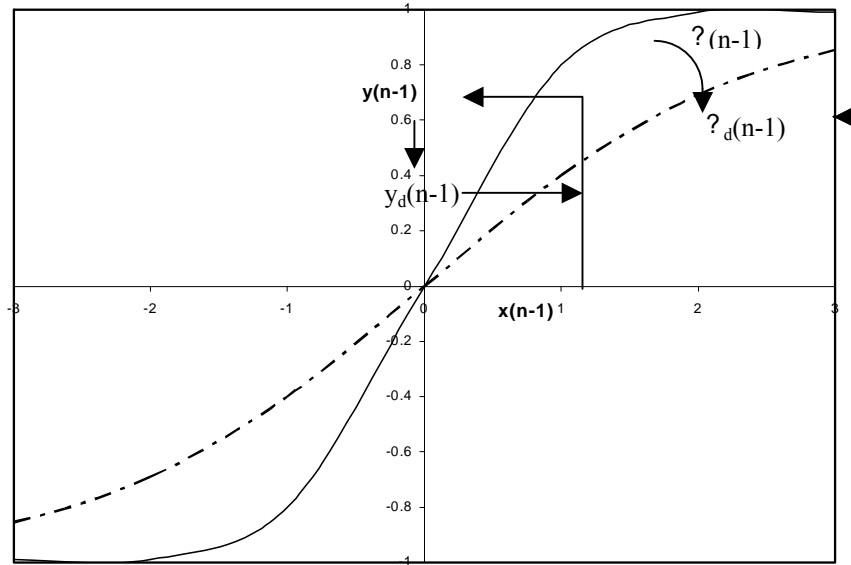
In Figure 4.6a, the continuous line shows the sigmoid activation function at  $(n-1)^{th}$  time index with slope parameter  $\phi(n-1)$ . Then corresponding to a given internal activity  $x(n-1)$ , the output of the neuron is  $y(n-1)$ . Assuming the local error  $\delta(n-1)$  at the neuron (node) to be large negative (LN), the desired output value at the node  $y_d(n-1)$  will lie below  $y(n-1)$  as shown in the above figure. Now, if the local error is to be minimized a solution is to be sought such that for a given  $x(n-1)$  the output should be  $y_d(n-1)$ . In order to achieve this the sigmoidal activation function must be rotated through the origin in clock-wise direction so that it will occupy a new position representing the desired value  $\phi_d(n-1)$  as shown (dotted line) in Figure 4.6a. This activation function will be such that it should pass through the intersection of the vertical line through  $x(n-1)$  and the horizontal line through  $y_d(n-1)$ .

Now in Figure 4.6b the continuous line represents the sigmoidal activation function of the node at the  $n^{th}$  time index with slope parameter  $\phi_d(n-1)$ . Here corresponding to a given input  $x(n)$ , the output at the node is  $y(n)$ . Assuming that the error term  $\delta(n)$  to be small positive (SP), the desired output value  $y_d(n)$  will lie above  $y(n)$  as shown. Thus a new position of the activation function need to be estimated such that the error term at the given node is minimized as mentioned before and it should pass through the intersection of the vertical line through  $x(n)$  and the horizontal line through  $y_d(n)$  which will represent the desired slope parameter  $\phi_d(n)$ . At this instant of time ( $n^{th}$  time index) the fuzzy controller approach is applied to compute the change in the sigmoid slope at the given node for  $(n+1)^{th}$  time index, which takes into account the estimates of  $\delta(n)$  and  $\Delta\delta(n)$  where

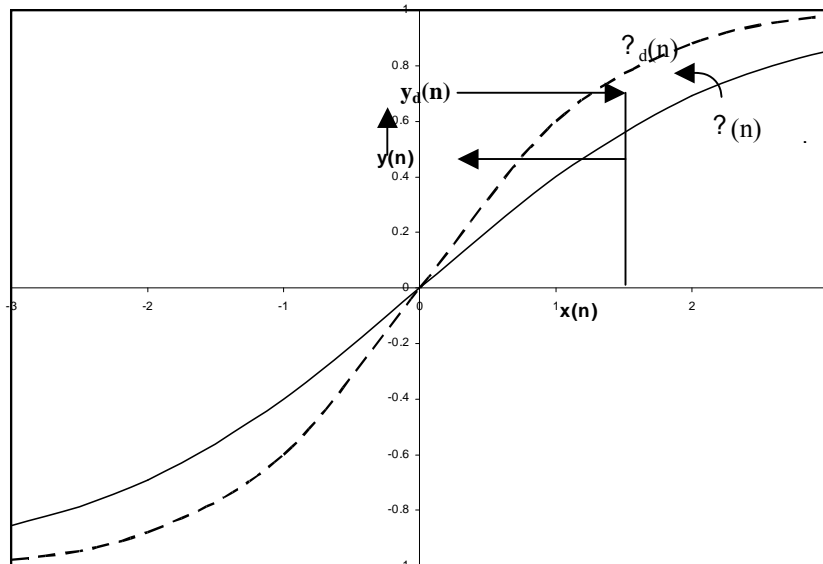
$$\Delta\delta(n)=\delta(n) - \delta(n-1) \tag{4.67}$$

Considering  $\delta(n)$  to be small positive (SP) and  $\delta(n-1)$  to be large negative (LN),  $\Delta\delta(n)$  will then be categorised as large positive (LP). Now corresponding to the inputs  $\delta(n)$  and  $\Delta\delta(n)$  applied to the fuzzy controller block shown in Figure 4.5, the change in the sigmoid slope  $\Delta\phi(n)$  is estimated to be large positive(LP). This leads to the correction of sigmoid slope parameter at  $(n+1)^{th}$  time index after suitable defuzzification incorporated in Figure 4.6c and is given by

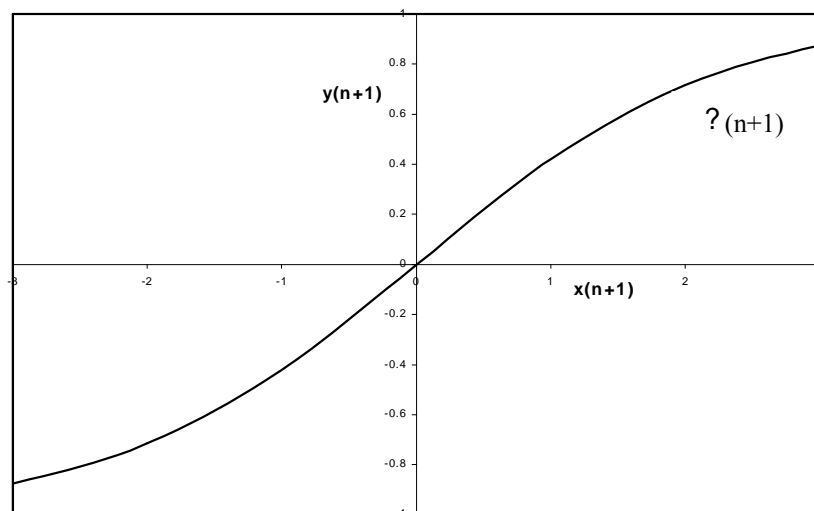
$$\phi(n+1) = \phi(n) + \Delta\phi(n) \tag{4.68}$$



(a)



(b)



(c)

Figure 4. 6: Sigmoid activation function (a)  $(n-1)^{th}$ , (b)  $n^{th}$ , and (c)  $(n+1)^{th}$  time

## 4.5 Simulation study

An exhaustive computer simulation study has been undertaken for evaluating the performance of all the proposed neural equaliser structures based on FNN topologies for a variety of linear and nonlinear real communication channel models. The simulation model of an adaptive equaliser considered, is illustrated in Figure 4.7. In the simulation study the channel under investigation is excited with a 2-PAM signal, where the symbols are extracted from uniformly distributed bipolar random numbers  $\{-1,1\}$ . The channel output is then contaminated by an AWGN (Additive White Gaussian Noise). The pseudo-random input and noise sequences are generated with different seed values for the random number generators. For mathematical convenience, the received signal power is normalised to unity. Thus the received signal to noise ratio (SNR) is simply the reciprocal of the noise variance at the input of the equaliser. The power of additive noise has been taken as 0.01, representing a SNR of 20dB. The actual performance measure of an equaliser is the bit error rate (BER) corresponding to the misclassification of transmitted symbols. For all structures based on FNN topology in the simulation, the bit error rates are obtained with detected symbols being fed back, as this technique presents a more realistic scenario in comparison with correct symbol feed back [29]. The proposed equaliser structures are trained using the learning algorithms reported in this work and the connection weights are frozen after a training phase consisting of 1000 training samples. Then the BER performances for each SNR value under study are evaluated based on  $10^7$  more received symbols (test samples) and averaged over 20 independent realisations.

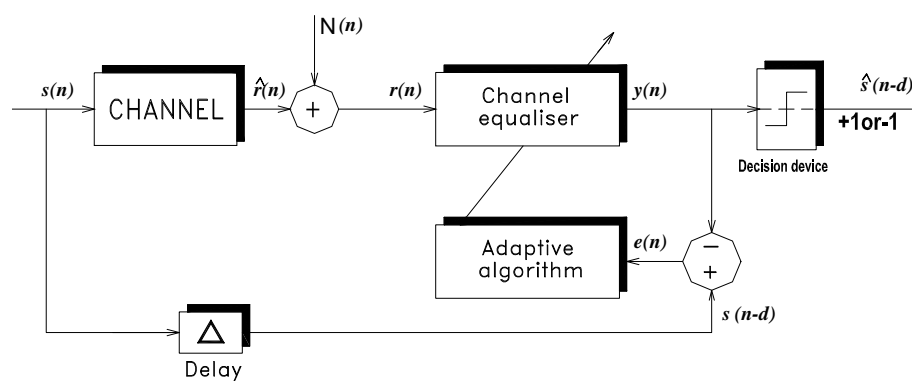


Figure 4.7: Simulation model of an adaptive equaliser



Further, it is observed that increasing the structural complexity (i.e., either increasing layer and/or number of nodes) the performance of the proposed neural equalisers improve as the opinions of more experts can be merged together to arrive at a decision. However, the most important objective of the proposed research work is to develop FNN based equaliser structures with low network complexity. So keeping this in view, a compromise has been made between the performance and the structural complexity. Therefore the proposed equalisers in all the examples considered here are chosen to be of reduced structural complexity as shown in Figures 4.8a-e and a comparison of the configurations of all these structures with the conventional FNN one is given in Table 4.2. The HKFNN structure has a two layer configuration with one node in each layer and the final node is a summing unit. The OBFNN structure consists of three layers with one node in each layer and the final node is a summing unit. The TDFNN structure has a single layer FNN with two nodes cascaded with a Transform block (a 2X2 DCT with normalisation) and the final node is a summing unit only. The FZTUNFNN structure chosen here is a reduced structure {two layer (1,1) FNN} with adaptable sigmoid slope parameter for each node. The structure design parameters for all the FNN equalisers with decision feedback configurations and the decision delay have been selected based on the new approach for parameter selection discussed in Section 3.2 of Chapter-3 in order to optimise the performance.

The Back-Propagation algorithm given in Appendix-A is applied to update the CFNN {a two-layer structure} and the proposed FZTUNFNN structure, where as the other proposed equaliser structures are adapted based on the algorithms as explained elaborately in Sections 4.1.1, 4.2.2 and 4.3.1. The values of various parameters ( $\eta$ ,  $\beta$  and  $\alpha$ ) chosen for weight adaptations used in the learning algorithms of all the structures are provided in Table 4.3.

Sl. No.	Equaliser structures	No. of Neurons	No. of Summing units	No. of Adaptable weights	No. of Fixed weight	Other Adaptable parameters	Figure No.
1	CFNN	6	-	36	-	-	4.8a
2	HKFNN	1	1	11	-	-	4.8b
3	OBFNN	2	1	08	-	-	4.8c
4	TDFNN	2	1	12	4	-	4.8d
5	FZTUNFNN	2	-	06	-	Slope( $\bullet$ )	4.8e

**Table 4.2 : Structural configuration comparison of the proposed FNN based equalisers with the conventional FNN, Feedforward order of all Equalisers ' $m$ '= 3 and feedback order ' $n_b$ ' =2.**

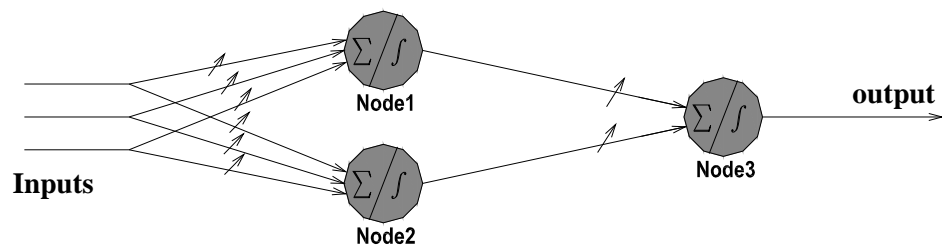


Figure 4.8a: A two layer CFNN structure

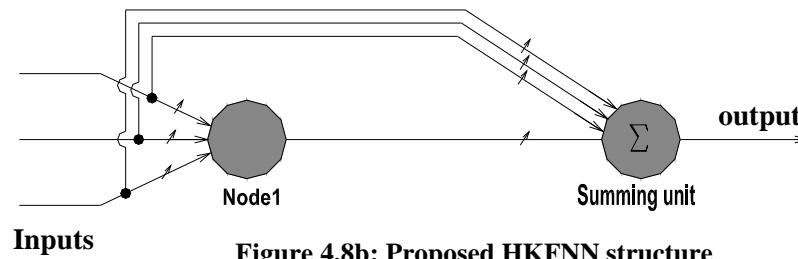


Figure 4.8b: Proposed HKFNN structure

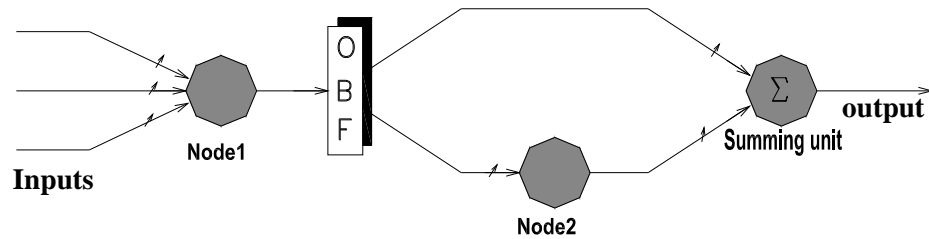


Figure 4.8c: Proposed OBFNN structure

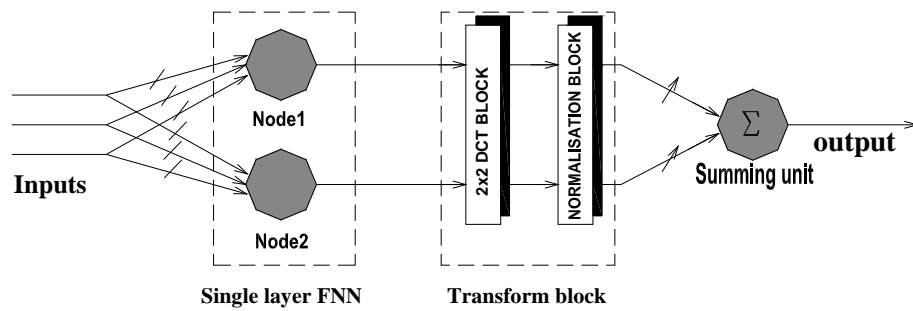


Figure 4.8d: Proposed TDFNN structure

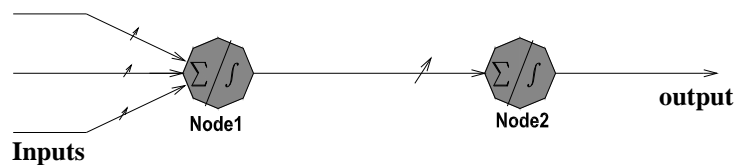


Figure 4.8e: Proposed FZTUNFNN structure

CHANNEL PARAMETERS		CFNN	HKFNN	OBFNN	TDFNN	FZTUNFNN
$H_1(z)$	$\eta$	0.8	0.8	0.2	0.8	0.3
	$\beta$	0.8	0.8	0.5	0.8	0.3
	$\alpha$	0.3	0.8	0.2	0.8	0.8
$H_2(z)$	$\eta$	0.5	0.6	0.2	0.5	0.3
	$\beta$	0.5	0.6	0.2	0.5	0.3
	$\alpha$	0.5	0.8	0.2	0.5	0.3
$H_3(z)$	$\eta$	0.2	0.5	0.2	0.2	0.3
	$\beta$	0.2	0.5	0.2	0.2	0.3
	$\alpha$	0.2	0.5	0.2	0.3	0.3
$H_4(z)$	$\eta$	0.8	0.5	0.09	0.2	0.2
	$\beta$	0.3	0.5	0.09	0.2	0.2
	$\alpha$	0.5	0.5	0.1	0.2	0.2
$H_5(z)$	$\eta$	0.3	0.5	0.09	0.5	0.09
	$\beta$	0.3	0.5	0.09	0.5	0.09
	$\alpha$	0.3	0.8	0.8	0.8	0.5
$H_6(z)$	$\eta$	0.2	0.5	0.1	0.3	0.2
	$\beta$	0.2	0.5	0.1	0.3	0.2
	$\alpha$	0.5	0.5	0.2	0.5	0.2
$H_7(z)$	$\eta$	0.3	0.5	0.09	0.9	0.3
	$\beta$	0.3	0.5	0.09	0.9	0.3
	$\alpha$	0.3	0.5	0.1	0.2	0.3
$H_8(z)$	$\eta$	0.3	0.2	0.5	0.1	0.2
	$\beta$	0.3	0.2	0.5	0.1	0.2
	$\alpha$	0.5	0.5	0.5	0.3	0.6
$H_9(z)$	$\eta$	0.3	0.2	0.2	0.1	0.3
	$\beta$	0.3	0.2	0.2	0.1	0.3
	$\alpha$	0.6	0.2	0.2	0.1	0.6
$H_{10}(z)$	$\eta$	0.2	0.3	0.09	0.02	0.2
	$\beta$	0.2	0.3	0.09	0.02	0.2
	$\alpha$	0.3	0.5	0.9	0.8	0.3
$H_{11}(z)$	$\eta$	0.5	0.3	0.1	0.8	0.3
	$\beta$	0.3	0.3	0.1	0.8	0.3
	$\alpha$	0.3	0.8	0.5	0.8	0.5
$H_{12}(z)$	$\eta$	0.3	2	0.1	0.09	0.06
	$\beta$	0.3	2	0.1	0.09	0.06
	$\alpha$	0.3	0.3	0.1	0.09	0.5
$H_{14}(z)$	$\eta$	0.2	0.3	0.3	0.6	0.4
	$\beta$	0.2	0.3	0.2	0.6	0.4
	$\alpha$	0.4	0.3	0.3	0.6	0.4
$H_{15}(z)$	$\eta$	0.2	0.5	0.3	0.5	0.35
	$\beta$	0.2	0.5	0.3	0.5	0.35
	$\alpha$	0.4	0.5	0.3	0.5	0.35

Table 4.3: Learning Parameters of Proposed FNN Based Equaliser Structures

#### 4.5.1 Performance analysis of the proposed FNN based equalisers

##### Example 1 : Channel $H_1(z)=1+0.5z^{-1}$

The first example used is a two-tap minimum-phase channel [12] defined by its transfer function  $H_1(z)=1+0.5z^{-1}$ . Figure 4.9 illustrates the comparison of the proposed structures with a conventional FNN (CFNN) configuration {a two-layer (2,1)} in terms of BER performance. The structures of all the proposed neural equalisers employed here are already mentioned in the previous section alongwith 'm'=2 (two samples in the feedforward section), 'n<sub>b</sub>'=1(one sample in the feedback section) and 'd'=1(transmitted sequence delayed by one sample). These three design parameters are selected from the channel characteristic by Equation 3.14 given in Chapter-3 following the new approach suggested in this present work for optimising the performance of equaliser. It is observed that the proposed equalisers on FNN framework yield a significant improvement in BER performance in comparison to their conventional counterpart. However, under severe noise conditions (SNR < 7dB), the conventional FNN equaliser structure and all the proposed equaliser configurations yield similar performance. But the superiority in the performance of the proposed ones over the conventional ones is distinct as the signal to noise ratio improves (i.e. for more realistic SNR levels). For example, at a prefixed error probability level (BER) of  $10^{-4}$ , the proposed HKFNN, FZTUNFNN and OBFNN equalisers are able to provide SNR gain of about 1.8 dB over the conventional FNN one.

##### Example 2 : Channel $H_2(z)=1+0.7z^{-1}$

The second example considered here is a channel with transfer function  $H_2(z)=1+0.7z^{-1}$ , a simple minimum phase channel [32] (no zeros near the unit circle) with flat frequency response is shown in Appendix-E. The design parameters of all the proposed equaliser structures chosen are  $m=2$ ,  $n_b=1$  and  $d=1$ . Comparison of BER performances of all proposed structures with a two layer conventional FNN {a (5,1) structure} is depicted in Figure 4.10, after presenting a training sequence of 1000 samples. It is noticed that proposed FNN based equalisers designed on a reduced structural configuration are able to provide better BER performance compared to the conventional FNN structure. For example, the proposed OBFNN equaliser achieves about 1.3dB improvement in SNR over the CFNN at a prefixed BER level of  $10^{-5}$ , where as the other new equalisers are also superior performance wise.

Example 3 : Channel  $H_3(z)=1+0.95z^{-1}$ 

The third example considered here is a typical 2-tap channel with transfer function defined by  $H_3(z)=1+0.95z^{-1}$  (zero close to unit circle) [111]. All the proposed equalisers (decision feedback type) are selected with parameters  $m = 2$ ,  $n_b = 1$  and  $d = 1$ , as this combination can provide the best BER performance based on Equation 3.14. After exposing the equaliser structures with 1000 training samples, the BER plots of all the proposed equalisers along with a CFNN equaliser {(2,1) structure} are illustrated in Figure 4.11. It is observed that application of new equaliser structures in FNN domain result in improvement of the BER performance. The proposed TDFNN and OBFNN equalisers are preferred for this channel model considering their performance (i.e., SNR gain of 1dB at a BER of  $10^{-4}$  over the CFNN equaliser).

Example 4 : Channel  $H_4(z)= 0.5 + z^{-1}$ 

The fourth example under study is a linear non-minimum phase channel [76] with transfer function described by  $H_4(z) = 0.5+ z^{-1}$ . All the equaliser structures used here for comparative study are initially trained with a sequence of 1000 samples, then the weights are frozen and the BER performances are evaluated. For such type of channel, three input samples (two in the feedforward section ‘ $m=2$ ’ and one in the feedback section ‘ $n_b=1$ ’) are presented to all proposed FNN based equalisers. Further, the training sequence is delayed by one sample ‘ $d=1$ ’ as decided in confirmation with Equation 3.15 described in Chapter 3. The BER plots given in Figure 4.12, show that all the proposed equalisers with much reduced structural complexity are able to provide improved BER performance in comparison to a CFNN one {a conventional FNN (5,1) structure} for various SNR conditions.

Example 5 : Channel  $H_5(z)=0.3482+ 0.8704 z^{-1}+ 0.3482 z^{-2}$ 

Another example of a linear non-minimum phase channel [12] with transfer function,  $H_5(z)=0.3482+ 0.8704 z^{-1}+ 0.3482 z^{-2}$ , is considered. Such type of channel is close to those encountered in practical communication system and widely used in technical literatures. Figure 4.13 depicts the BER performance comparison of a CFNN {(3,2) DFE with a 2 layer (2,1) FNN structure} equaliser with all the proposed structures discussed in this chapter. The decision delay ‘ $d$ ’ for the transmitted training sequence is chosen to be 2. The selection of parameters has been done in accordance with Equation 3.11 as this channel model is a symmetrical one. It is observed from Figure 4.13 that all those proposed equalisers gain in terms of BER performance over the conventional FNN equaliser.

Example 6 : Channel  $H_6(z)=0.4084 + 0.8164z^{-1}+0.4084z^{-2}$

The next example is a typical linear channel, given by  $H_6(z) = 0.4084 + 0.8164 z^{-1} + 0.4084 z^{-2}$ . This example represents a kind of worst scenario as the performance of this channel [70] is mainly limited by the fact that there is coincidence of channel states corresponding to different classes in the input observation space. All the equalisers chosen for a comparative analysis here are DFE structures with parameters  $m=3$ ,  $n_b=2$ , and  $d=2$ . The number of training samples is fixed at 1000 samples for all equaliser configurations and the BER performance is evaluated. It is observed from Figure 4.14 that under high noise situations (SNR < 7 dB), no considerable gain in performance of all the proposed structures is noticed. However, all the proposed structures offer improvement in BER performance over the CFNN one {a (5,1) structure} beyond 20dB SNR conditions. Especially, at a prefixed BER level of  $10^{-5}$ , the HKFNN equaliser offers SNR gain of about 1.3dB over the CFNN equaliser.

Example 7 : Channel  $H_7(z)=1- 2z^{-1}+ z^{-2}$

Further, the dominance of the proposed equalisers over the existing CFNN structure is established by considering an example of a partial response channel [34] described by transfer function  $H_7(z)= 1- 2z^{-1}+ z^{-2}$ . This channel has a double zero on the unit circle. Such channels are frequently encountered in magnetic recording. The CFNN equaliser is a 2 layer (5,1) structure with parameters chosen as  $m=3$ ,  $n_b=2$  and  $d=2$ . The robustness of the new equaliser structures in FNN domain is confirmed from Figure 4.15, which illustrates the BER curves after presentation of 1000 samples during the training phase. The proposed TDFNN equaliser shows significant performance gain in terms of the minimum SNR to get a prefixed BER (16 dB SNR against 18.5dB SNR to obtain an error probability level of  $10^{-3}$ ). Application of the HKFNN equaliser results in an improvement of 1.4 dB in SNR level at a prefixed BER of  $10^{-3}$ .

Example 8 : Channel  $H_8(z)=0.407 - 0.815 z^{-1}-0.407 z^{-2}$

Another three tap channel characterised by  $H_8(z)=0.407 - 0.815 z^{-1} - 0.407 z^{-2}$  [52] has been studied. Figure 4.16 depicts the significant BER performance enhancement by all the proposed equalisers when compared with a CFNN one {(2,1) structure} with parameters chosen as  $m=3$ ,  $n_b=2$  and  $d=2$  after being trained with 1000 samples. While the OBFNN equaliser is a clear winner providing a significant gain of 2.4dB in SNR level, the other proposed structures also result in gains of about 1.6 dB-1.8 dB in SNR level at a BER of  $10^{-4}$  over the CFNN equaliser.

Further, other examples of typical four tap  $H_9(z)$ ,  $H_{10}(z)$  [29,74] and five tap channels models  $H_{11}(z)$ ,  $H_{12}(z)$  [29,112] are considered. Their zero locations and frequency responses are described in Appendix D and E respectively. Amongst these examples taken, channel  $H_{12}(z)$  is a deep-null communication channel with significant inter symbol distortion. The transfer functions of all these channels are characterised by

$$H_9(z) = 0.7255 + 0.584 z^{-1} + 0.3627 z^{-2} + 0.0724 z^{-3}$$

$$H_{10}(z) = 0.35 + 0.8 z^{-1} + 1.0 z^{-2} + 0.8 z^{-3}$$

$$H_{11}(z) = -0.2052 - 0.5131 z^{-1} + 0.7183 z^{-2} + 0.3695 z^{-3} + 0.2052 z^{-4}$$

$$H_{12}(z) = 0.9413 + 0.3841 z^{-1} + 0.5684 z^{-2} + 0.4201 z^{-3} + z^{-4}$$

The BER performances of all the four proposed FNN based equaliser structures are compared with the CFNN equaliser after 1000 samples are shown during training phase and weights are frozen. In Figure 4.17, the proposed OBFNN and TDFNN structures provide better result over the two layer {2,1} CFNN equaliser for the channel  $H_9(z)$  in terms of BER performance. For the channel  $H_{10}(z)$ , the proposed TDFNN, HKFNN and FZTUNFNN structures show significant improvement in BER performance when compared with a two layer {9,1} CFNN equaliser as shown in Figure 4.18. Further, all the proposed equalisers in FNN domain are able to provide performance improvement over the two layer {2,1} conventional FNN as described in Figure 4.19 for  $H_{11}(z)$ . In the BER performance comparison of proposed FNN based structures with a two layer {5,1} conventional FNN equaliser for channel  $H_{12}(z)$  shown in Figure 4.20, the TDFNN structure is a clear winner though the HKFNN, FZTUNFNN and OBFNN structures result gain in BER performance.

In order to prove the robustness and consistency in performance of all the proposed neural structures, equalisation of nonlinear channels is simulated. Such nonlinear channels are frequently encountered in several places like the telephone channel, in data transmission over digital satellite links, especially when the signal amplifiers operate in their high gain limits and in mobile communication where the signal may become non-linear because of atmospheric nonlinearities. These typical channels encountered in real scenario and commonly referred to in technical literatures [32,49] are described by following transfer functions.

$$H_{14}(z) = (1 + 0.5 z^{-1}) - 0.9 (1 + 0.5 z^{-1})^3$$

and

$$H_{15}(z) = (0.3482 + 0.8704 z^{-1} + 0.3482 z^{-2}) + 0.2 (0.3482 + 0.8704 z^{-1} + 0.3482 z^{-2})^2$$

The simulation studies conducted on such channels for a 2-PAM signalling scheme have further confirmed that all the proposed FNN based equalisers are performance wise superior in comparison to conventional FNN structure as shown in Figure 4.21 and Figure 4.22 for the channels  $H_{14}(z)$  and  $H_{15}(z)$  respectively.

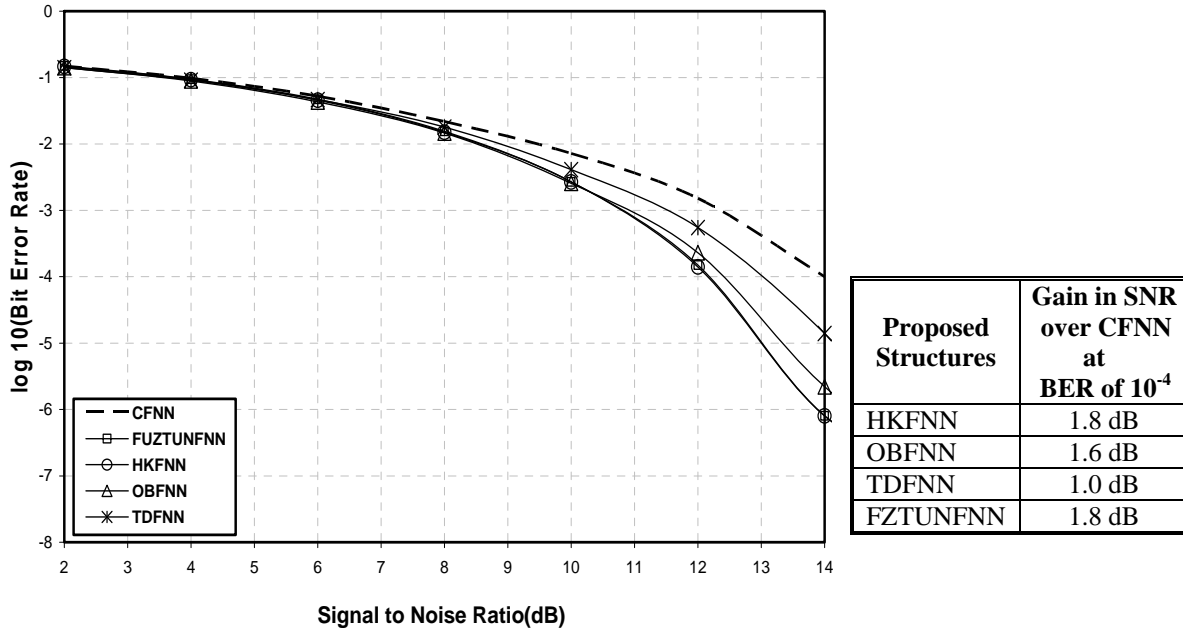


Figure 4.9: BER performance comparison of proposed FNN based equalisers with conventional FNN for Channel  $H_1(z)$

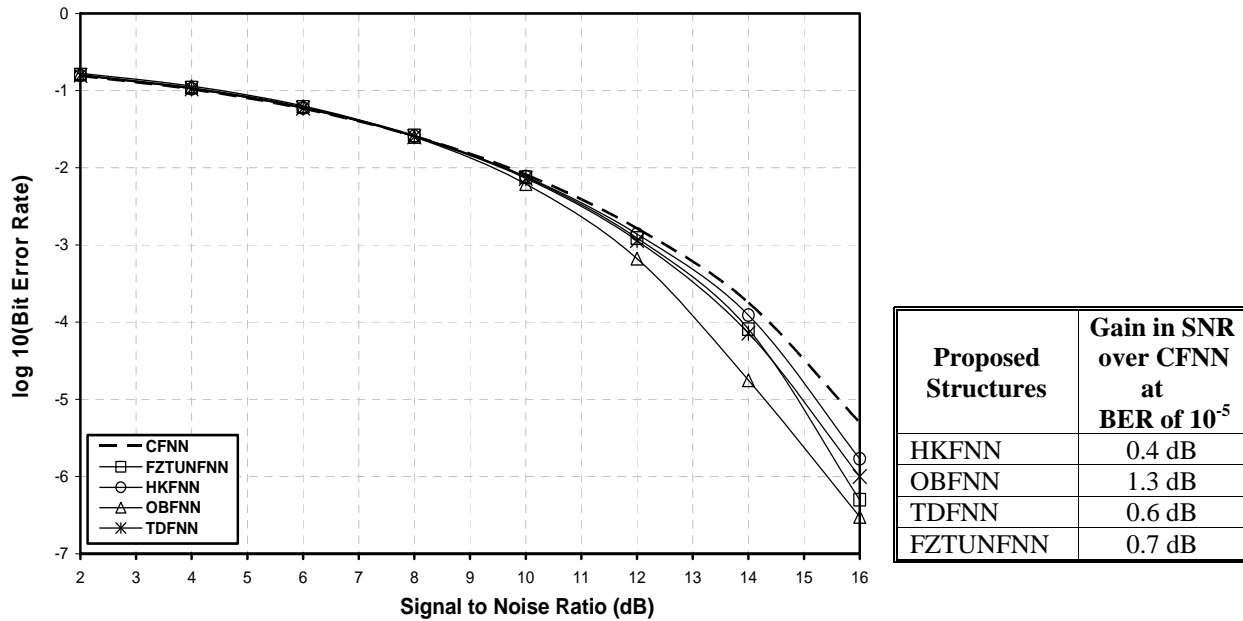


Figure 4.10: BER performance comparison of proposed FNN based equalisers with conventional FNN for Channel  $H_2(z)$



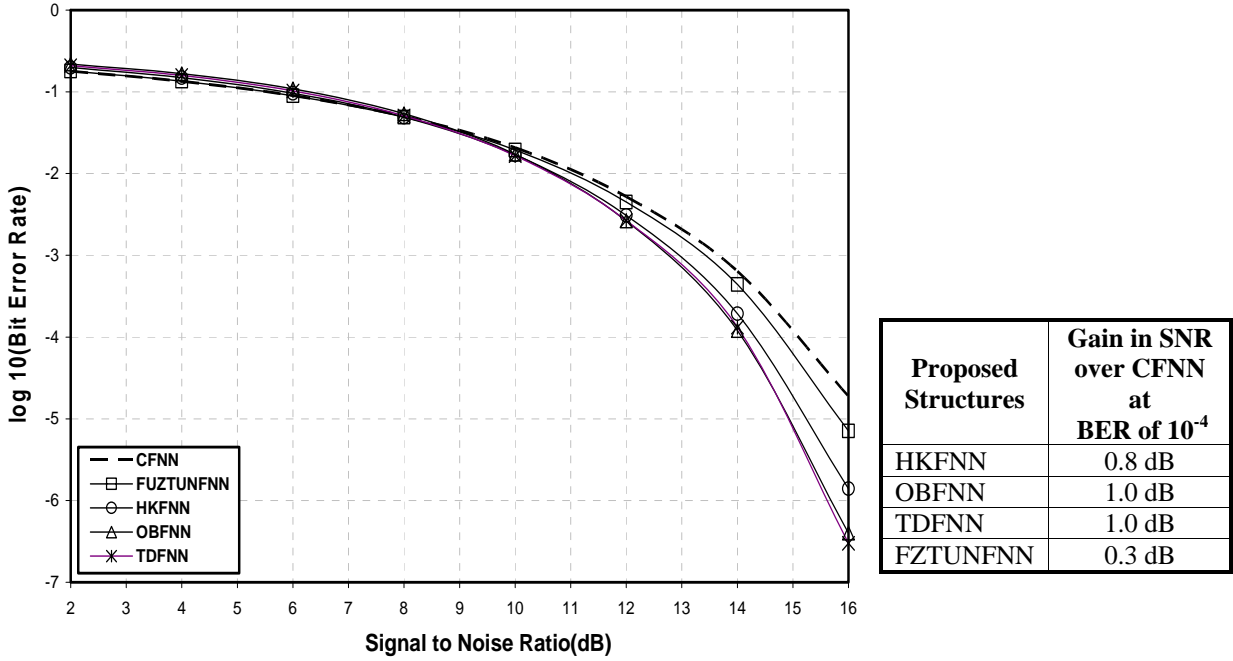


Figure 4.11: BER performance comparison of proposed FNN based equalisers with conventional FNN for Channel  $H_3(z)$

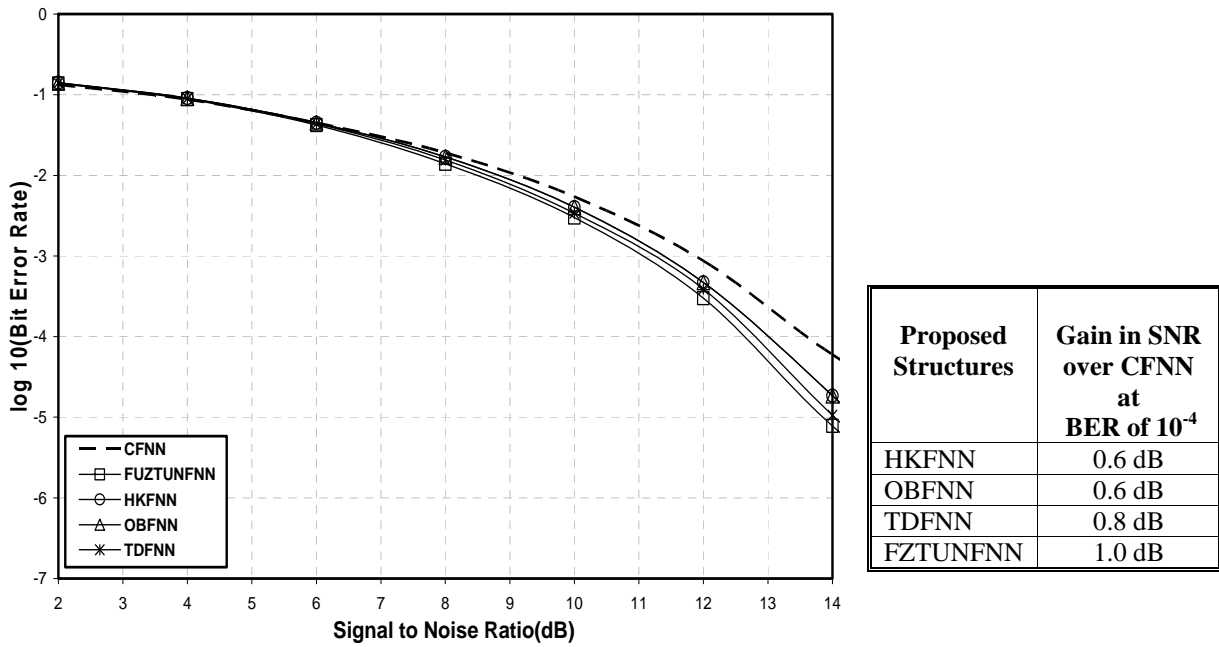


Figure 4.12: BER performance comparison of proposed FNN based equalisers with conventional FNN for Channel  $H_4(z)$

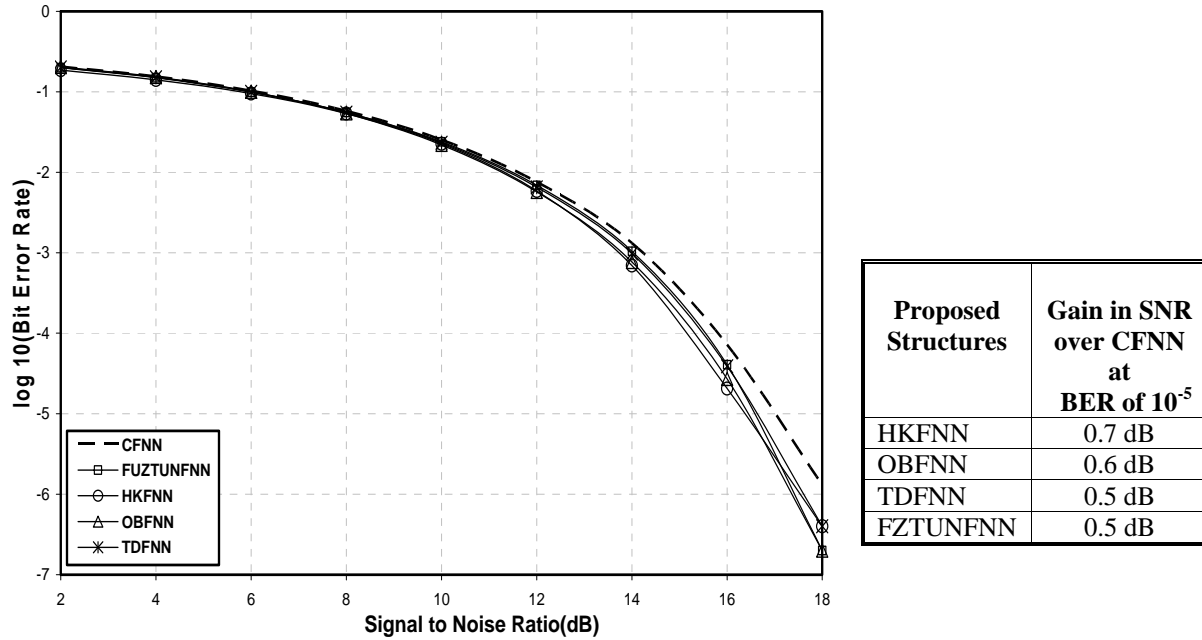


Figure 4.13: BER performance comparison of proposed FNN based equalisers with conventional FNN for Channel  $H_5(z)$

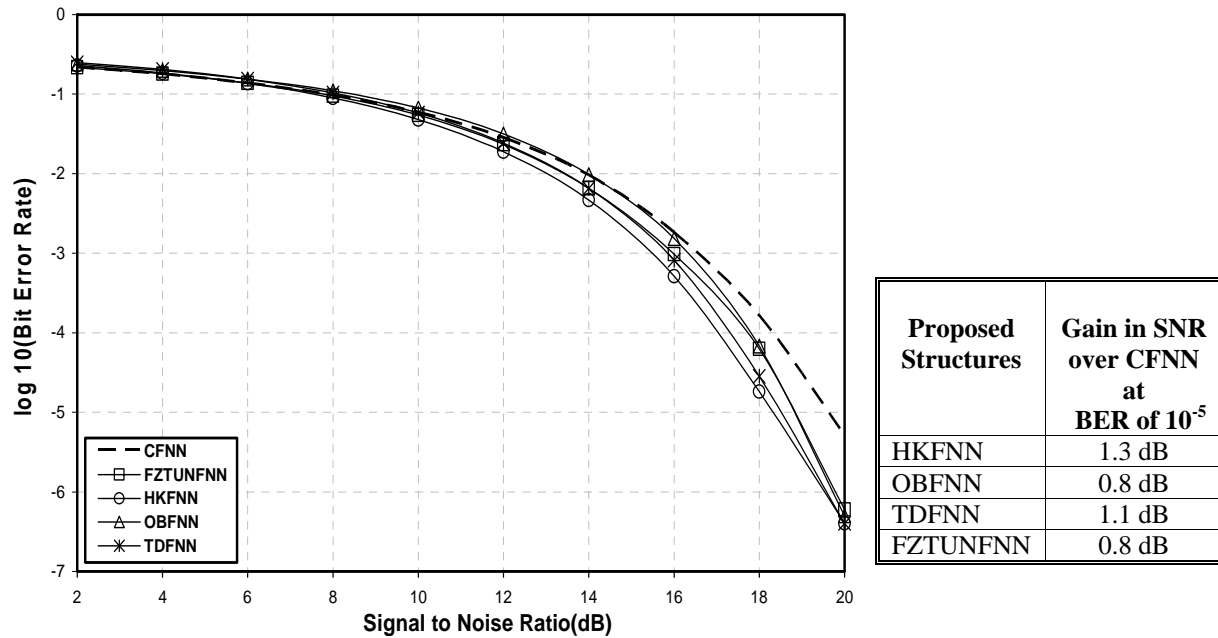


Figure 4.14: BER performance comparison of proposed FNN based equalisers with conventional FNN for Channel  $H_6(z)$

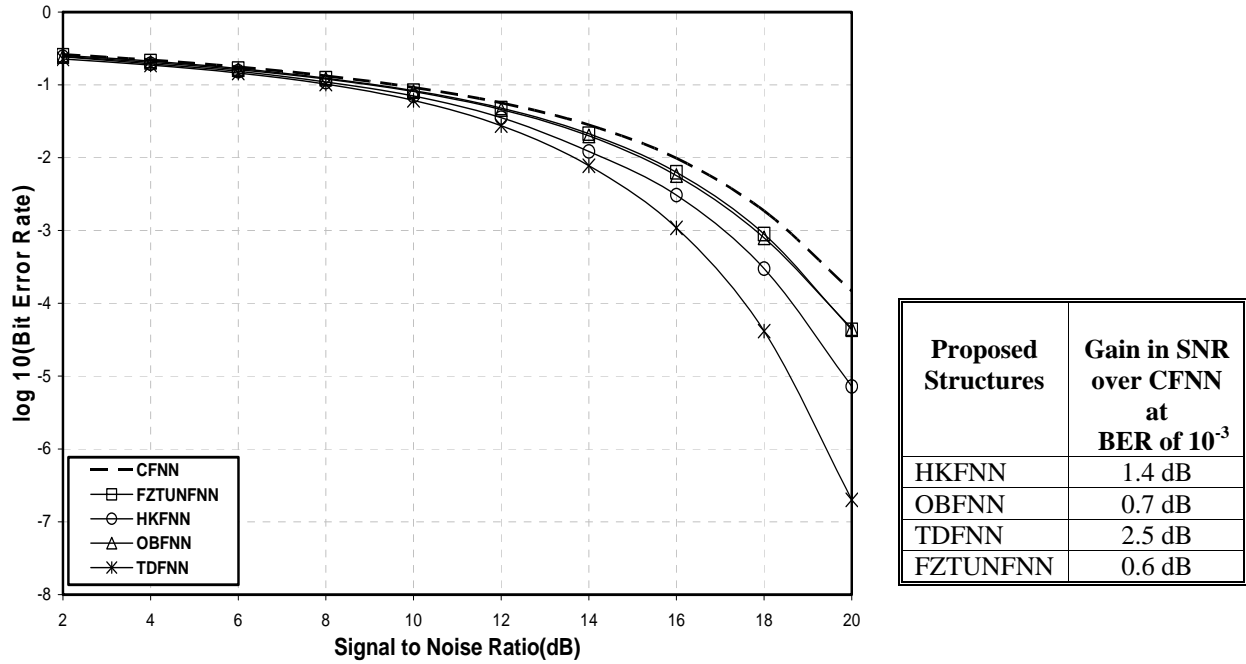


Figure 4.15: BER performance comparison of proposed FNN based equalisers with conventional FNN for Channel  $H_7(z)$

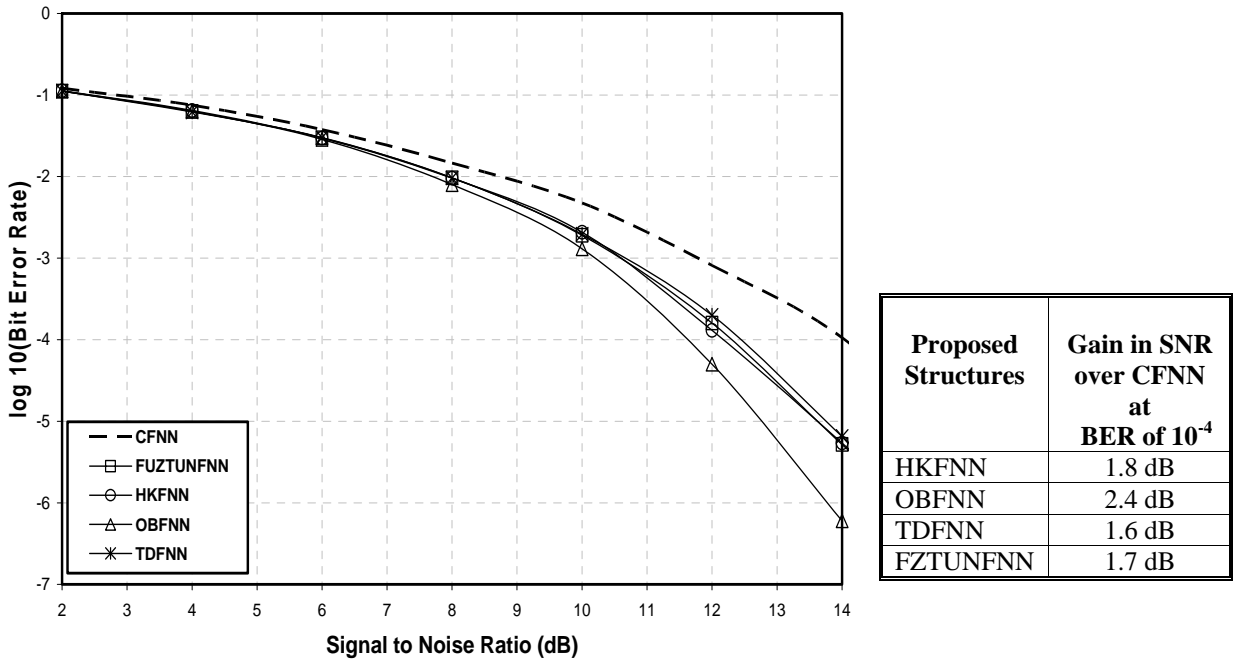


Figure 4.16: BER performance comparison of proposed FNN based equalisers with conventional FNN for Channel  $H_8(z)$

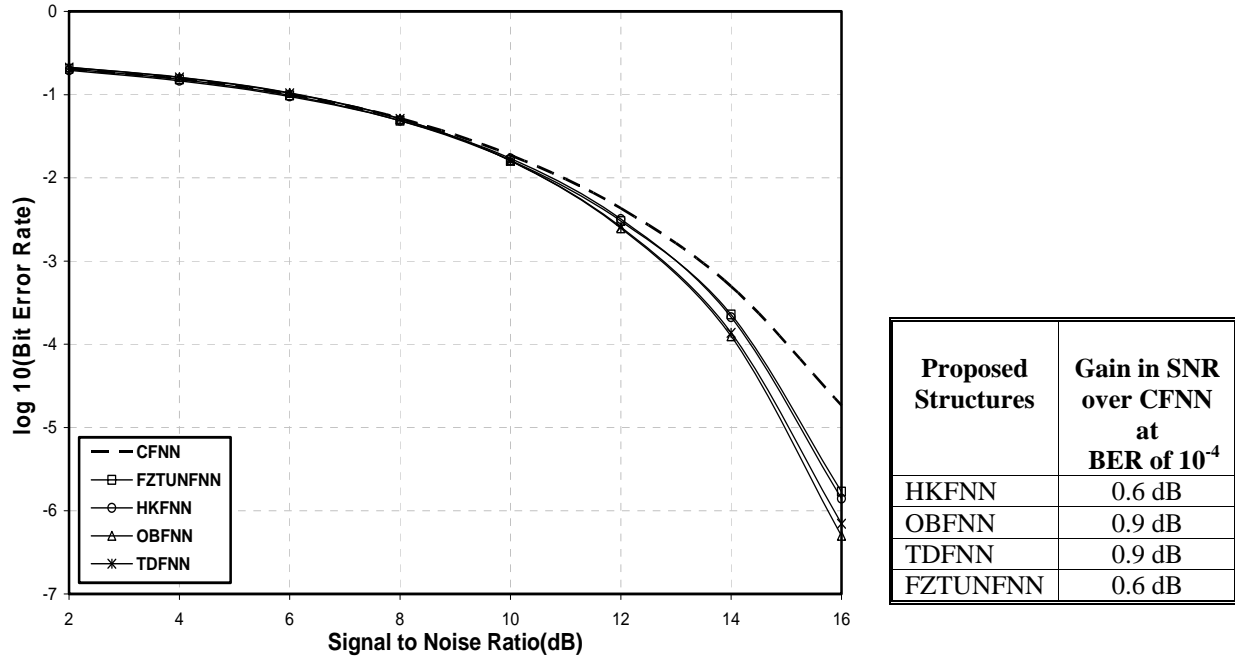


Figure 4.17: BER performance comparison of proposed FNN based equalisers with conventional FNN for Channel  $H_9(z)$

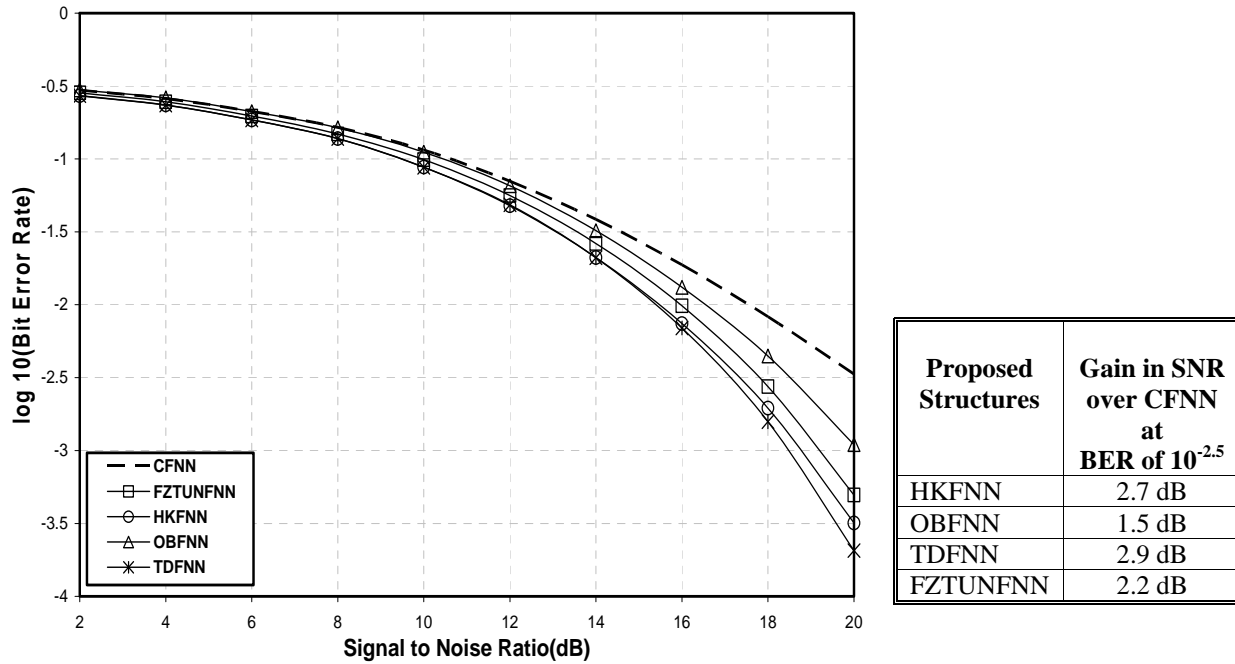


Figure 4.18: BER performance comparison of proposed FNN based equalisers with conventional FNN for Channel  $H_{10}(z)$

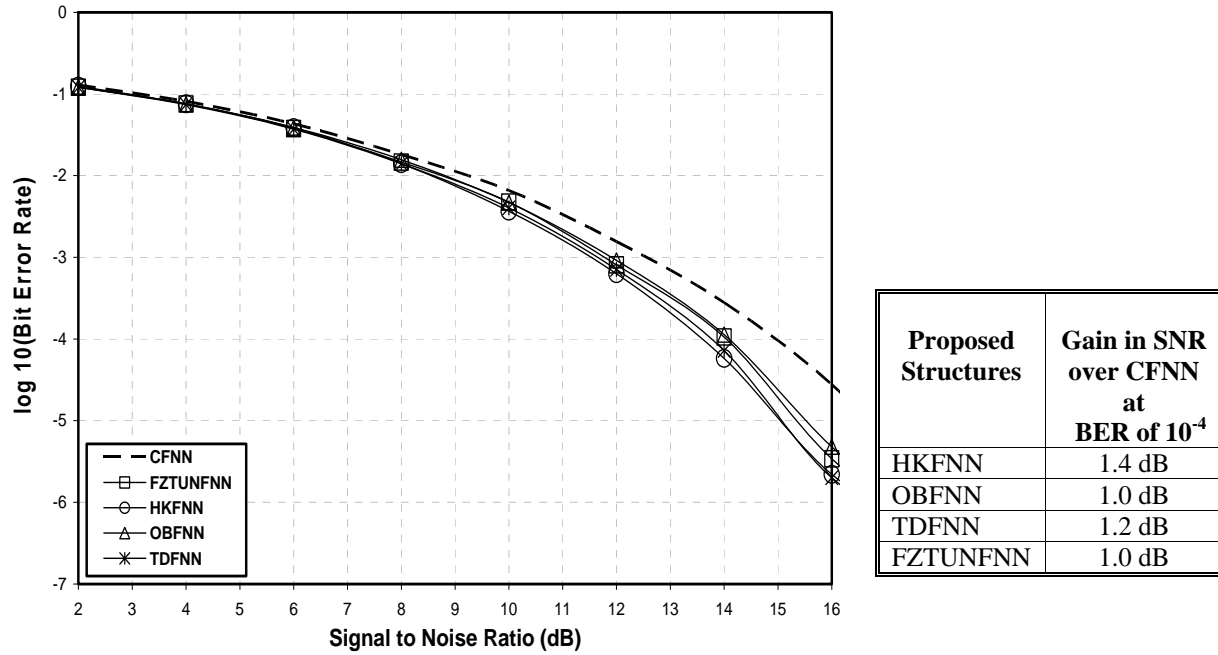


Figure 4.19: BER performance comparison of proposed FNN based equalisers with conventional FNN for Channel  $H_{11}(z)$

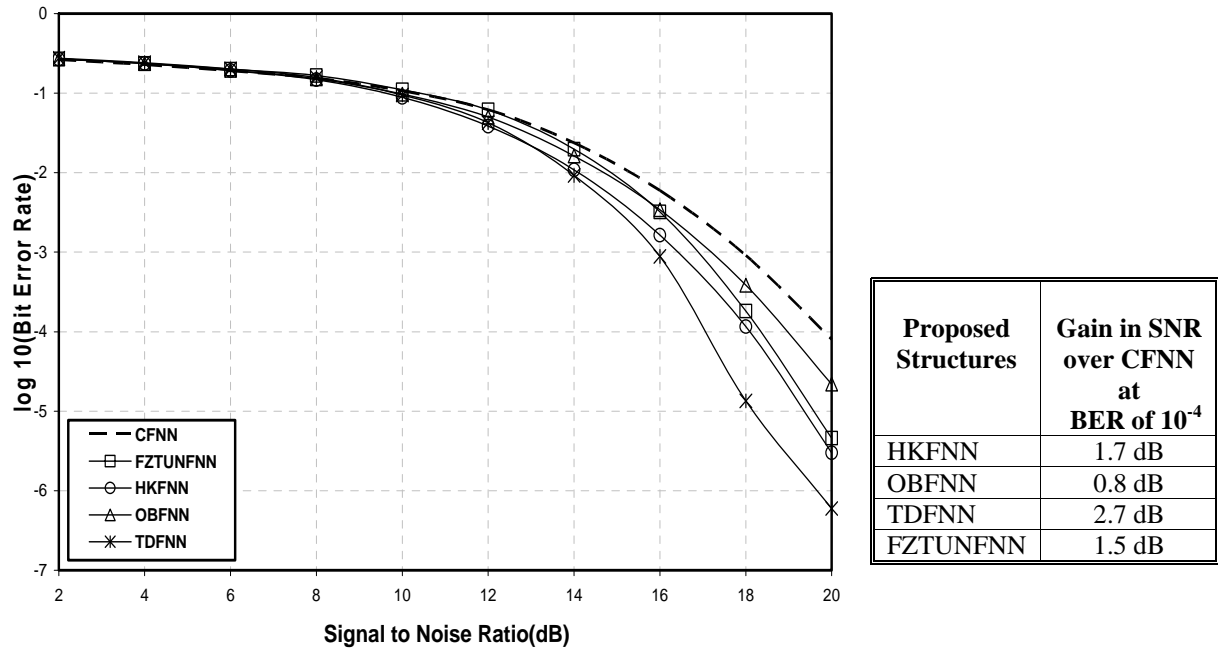


Figure 4.20: BER performance comparison of proposed FNN based equalisers with conventional FNN for Channel  $H_{12}(z)$

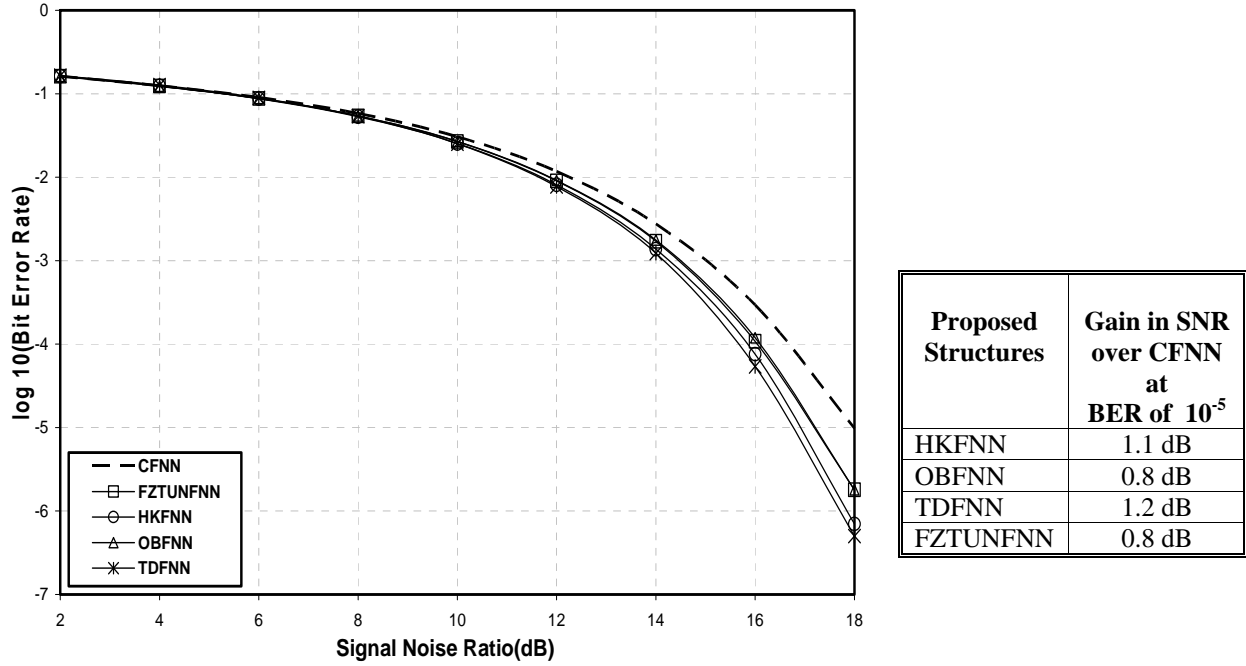


Figure 4.21: BER performance comparison of proposed FNN based equalisers with conventional FNN for Channel  $H_{14}(z)$

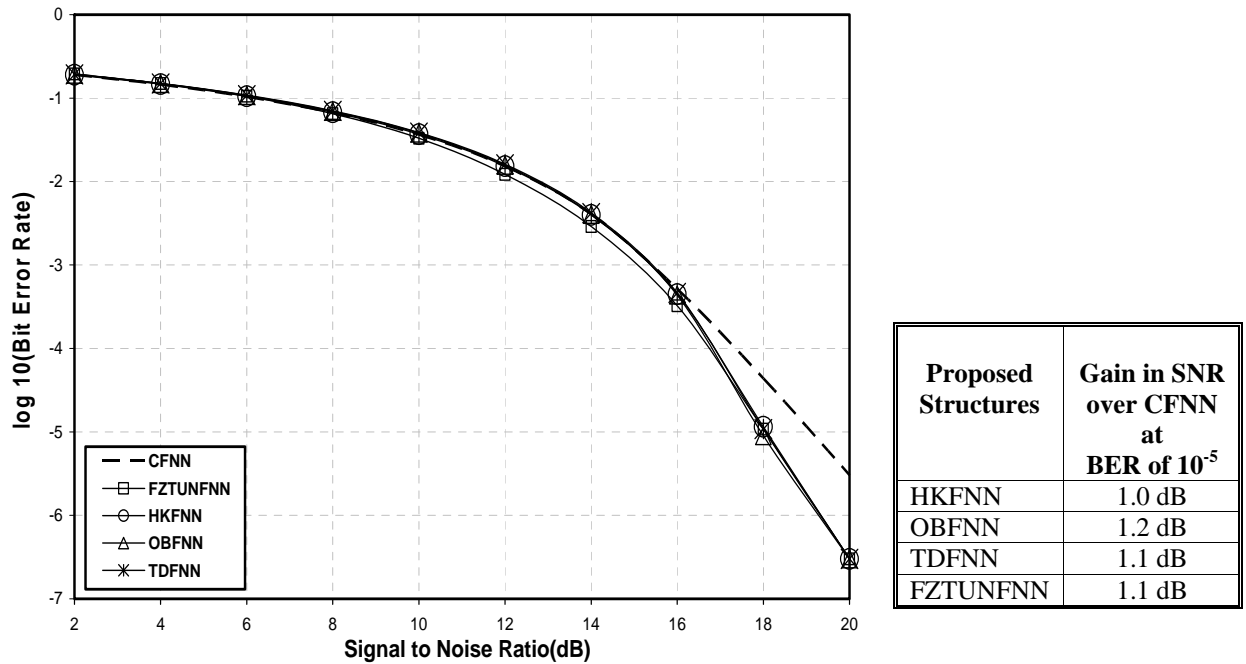
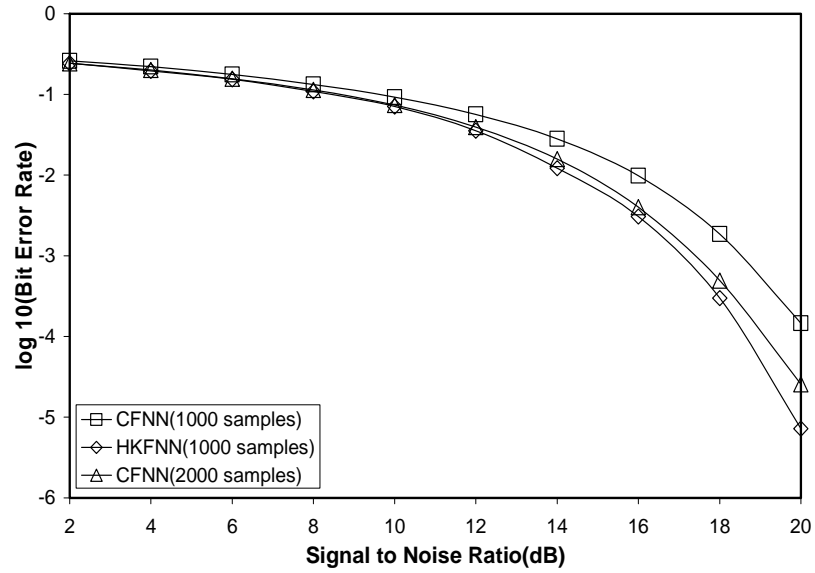


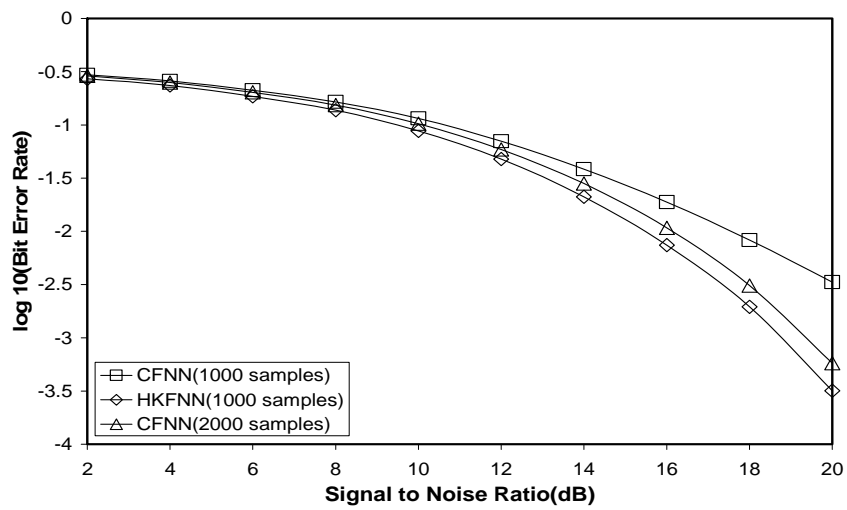
Figure 4.22 : BER performance comparison of proposed FNN based equalisers with conventional FNN for Channel  $H_{15}(z)$

In all the above comparative studies used to evaluate BER, the number of training samples presented to all the proposed equaliser structures been restricted to 1000 samples only as satisfactory BER performance is observed. Further, the advantage gained in terms of performance enhancement and faster training by the proposed FNN based equalisers can be clearly demonstrated by comparing their BER performance with a CFNN structure exposed to more number of samples in training phase. In Figure 4.23, it is observed that for examples of channels  $H_7(z)$  and  $H_{10}(z)$ , the CFNN equaliser needs to be trained with more samples (2000 samples) to achieve the BER performance level of the proposed HKFNN structure. For examples of channels  $H_3(z)$  and  $H_8(z)$  shown in Figure 4.24, the OBFNN equaliser is still found to be superior performance wise even if the conventional FNN equaliser is trained using 2000 samples. Further, it is observed in examples of channels  $H_7(z)$  and  $H_{10}(z)$  from Figure 4.25 that the conventional FNN structure can not obtain the performance of the proposed TDFNN equaliser though the length of training phase is increased to 2000 samples. Lastly, the gain in terms of training samples and performance employing the proposed FZTUNFNN equaliser over the CFNN one is observed from Figure 4.26 by considering examples of channels  $H_1(z)$  and  $H_8(z)$ .

Finally it is concluded here that all the proposed equaliser structures in the FNN framework reported in this chapter yield superior results not only in terms of BER performance but also provide faster learning (i.e., exactly half the number of training samples in comparison to a conventional FNN based equaliser). Also all the proposed equalisers consist of reduced structures in comparison to their conventional FNN counterpart.



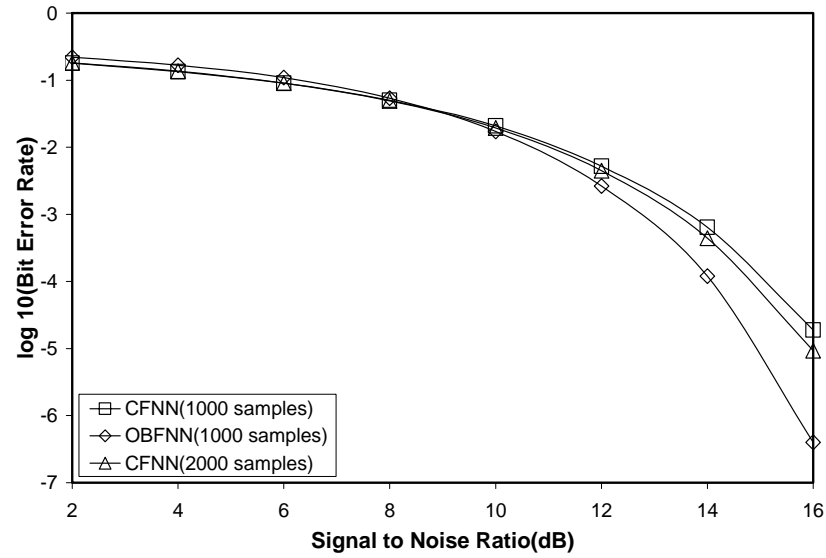
(a)



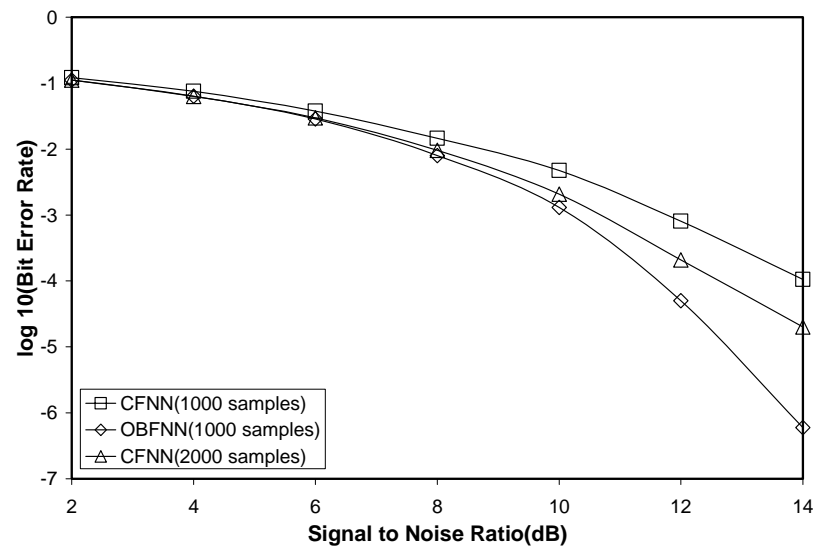
(b)

Figure 4.23: BER performance comparison of HKFNN equaliser with CFNN w.r.t. training samples for (a) Channels  $H_7(z)$  and (b)  $H_{10}(z)$



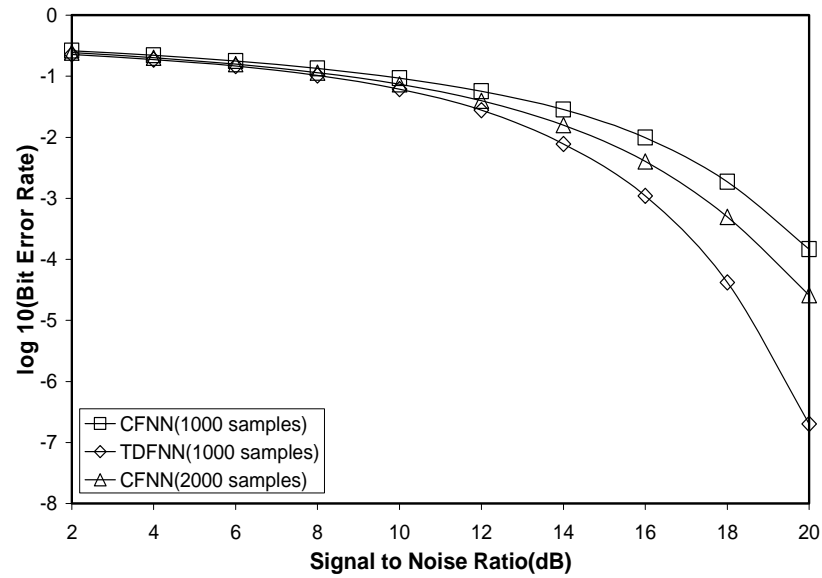


(a)

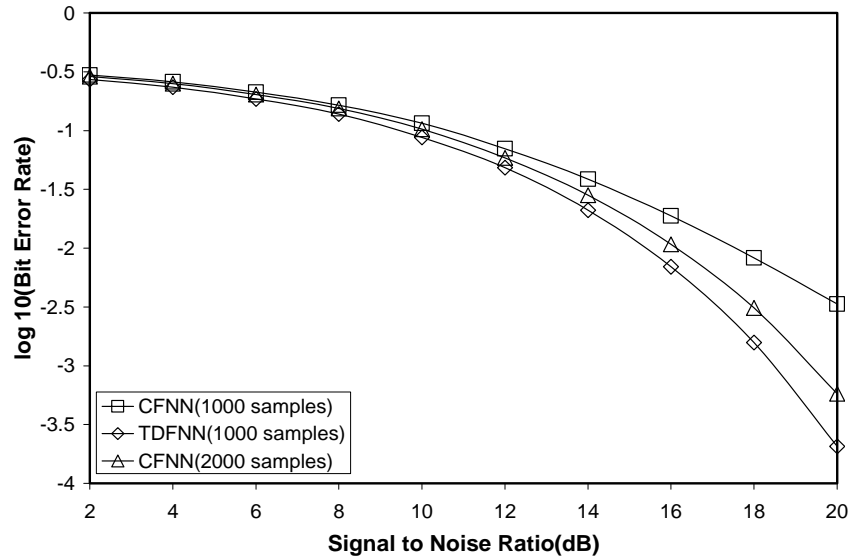


(b)

Figure 4.24 : BER performance comparison of OBFNN equaliser with CFNN w.r.t. training samples for Channels (a)  $H_3(z)$  and (b)  $H_8(z)$

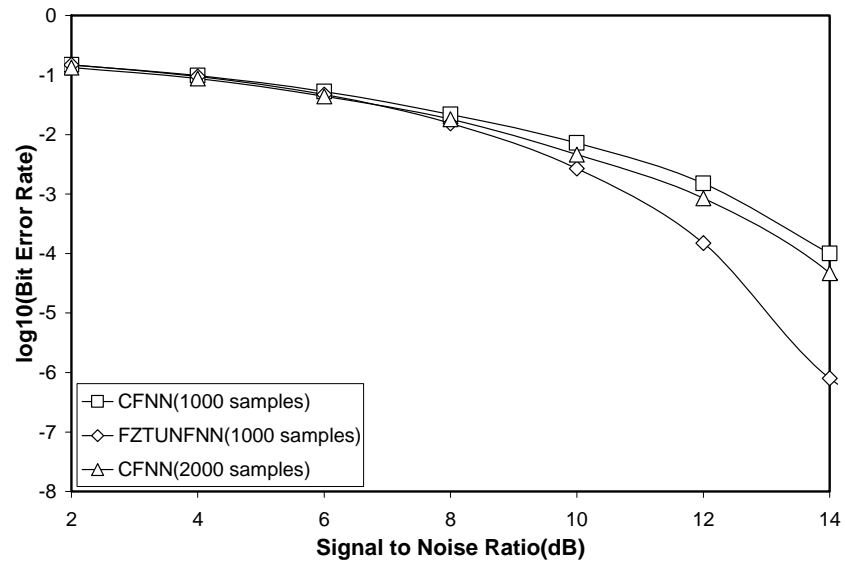


(a)

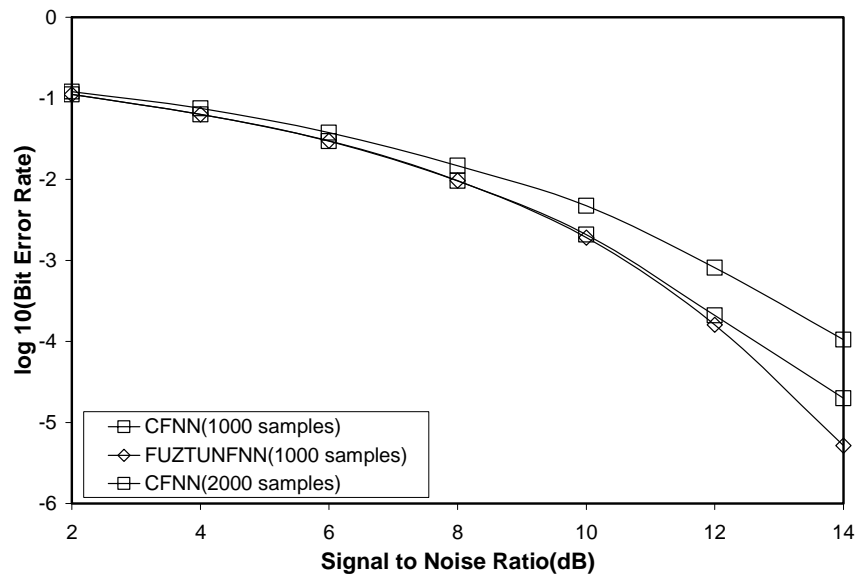


(b)

Figure 4.25 : BER performance comparison of TDFNN equaliser with CFNN w.r.t. training samples for Channels (a)  $H_7(z)$  and (b)  $H_{10}(z)$



(a)



(b)

Figure 4.26 : BER performance comparison of FZTUNFNN equaliser with CFNN w.r.t. training samples for Channels (a)  $H_1(z)$  and (b)  $H_8(z)$

## 4.6. Conclusion

Various innovative approaches like the hierarchical knowledge reinforcement, the orthogonal basis function expansion technique using evolutionary concept, the transform domain based technique and the sigmoidal slope tuning using fuzzy logic control approach have been incorporated into an FNN framework for developing various efficient equaliser structures. The proposed structures based on FNN topology result significant improvement in BER performance in comparison to the conventional FNN equaliser. Here the major advantage gained is that all the proposed equalisers are of reduced structural configurations and thus the basic goal of research is maintained. Although all the training algorithms are basically based on the conventional BP technique, suitable modifications have been introduced considering the structural changes. For the weight adaptation in the orthogonal basis function based FNN, a new concept has been developed to propagate the output error backwards considering the positioning of the OBF block in this structure in Section 4.2.1. Again in the transform domain based FNN, the DCT block followed by power normalisation does not allow the application of error back propagation technique using the BP algorithm directly. Such inherent limitations of the proposed TDFNN equaliser has been overcome by considering the inverse discrete cosine transform in the learning algorithm as discussed in Section 4.3.1. Further concept of tuning the slope of the sigmoid activation function of a conventional FNN equaliser by fuzzy controller approach, utilised in FZTUNFNN equaliser results improved BER performance. Thus it is concluded that all the proposed structures show satisfactory BER performance after a training phase with almost half the number of samples of that required for conventional FNN one. It has also been observed from the exhaustive simulation studies that though all the equalisers proposed have resulted in encouraging performances, the gains obtained are entirely channel dependent. For example, in channel  $H_1(z)$ , the proposed HKFNN, FZTUNFNN and OBFNN equalisers perform better, while in channel  $H_{12}(z)$ , the TDFNN equaliser provides significant performance improvement over conventional FNN equaliser. Similarly, the proposed OBFNN equaliser achieves a significant gain over the conventional FNN one for channel  $H_8(z)$ .

---

## CHAPTER 5

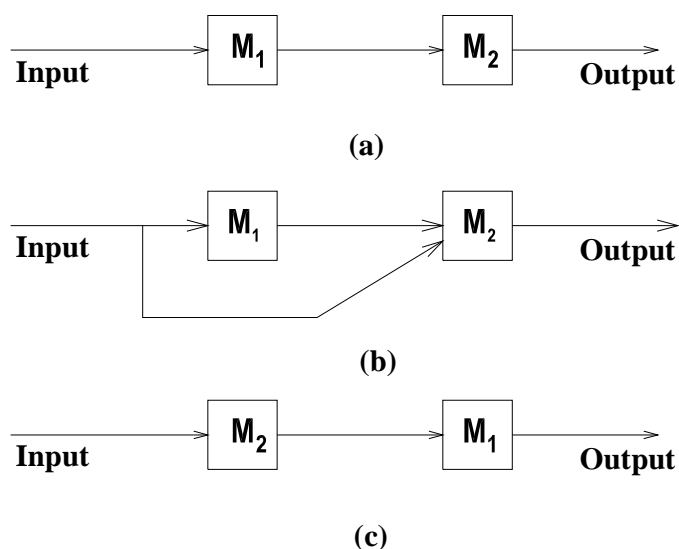
# Proposed RNN Based Cascaded Equalisers

---

Recurrent Neural Networks have emerged as a potential tool in the recent past for solving adaptive channel equalisation problems [31,32,33,34]. In this proposed work, much emphasis has been given to the design of new hybrid structures [112] using RNN as an integral module (block) and FNN as its supplementary module (block). Here, the cascading technique is adopted to evolve new topologies based on various combinations of RNN and FNN. However, the basic objective of this research of developing reduced network configurations remains and hence, while cascading is employed, it is ensured that under no circumstances this main purpose be defeated. It is a fact that cascading is responsible for structural growth and thus proper selection of number of nodes in both RNN and FNN modules becomes an extremely important criterion. In the light of above, these new hybrid networks possess low structural complexity.

The fundamental concept used here is that two sub-networks, RNN and FNN characterised as modules or blocks are cascaded in typical sequences as shown in Figures 5.1a-c. While the output of each node in each module is designated as an individual expert, the final output from these modules is termed as domain expert. Thus the opinion of a domain expert is formed as a collective measure of the opinions of the individual experts. Here, the original information (external input data) fed to a sub-network depending upon the cascaded configuration is pre-processed by the respective individual experts, giving rise to the opinion of the corresponding domain expert. Thereafter, the final decision of the given structure is obtained by suitably combining all the decisions from the various domain experts. It is logical to believe that decisions based on the judgment of a set of experts rather than individual ones can significantly improve the performance and this novel concept forms the basis of designing all the proposed equaliser structures in the present work.

This chapter highlights the development of some hybrid structures as discussed below. In Section 5.1, a cascaded network configuration consisting of a FNN module followed by RNN one has been proposed and suitable learning algorithm for the network is developed. The following Section 5.2 discusses another novel architecture which is basically the same as its predecessor, but with a structural change as the knowledge base of the domain responsible for the final decision has been enriched by reinforcing it with the original input information into it. The next Section 5.3 describes another variant of the cascaded configuration which is identical to hybrid structure, mentioned in Section 5.1 with the only exception that the FNN and RNN modules are swapped. The subsequent Section 5.4 deals with a novel idea, where even though cascading concept is used, it does not involve FNN module but is replaced by an orthogonal transform block. In Section 5.5, a fuzzy controller approach adopted to tune the slope of the sigmoid activation functions of all the nodes of the RNN module is presented and the technique employed is similar to the one already reported in Section 4.4. Simulation study and performance analysis of all the proposed equalisers are provided in Section 5.6. Lastly, this chapter is summarised in Section 5.7.



**Figure 5.1: Cascaded Configurations, where  $M_1$  denotes module 1 (FNN) and  $M_2$  denotes module 2 (RNN)**

## 5.1 FNN – RNN cascaded (FRCS) equaliser

The proposed architecture is a combination of two sub-networks, where FNN (module 1) is followed by RNN (module 2). Here, the knowledge base of module 2 is strengthened after pre-processing of the original information in module 1. The final output is the expert opinion from one of the RNN nodes as the equaliser structure has to be a single output system. It is evident from the network that as the RNN module is at the output end, an inherent decision feedback strategy (pseudo-decision feedback) has been incorporated into it, because RNN is by default a self-feedback network. The network has to be adapted during the training phase and then its performance evaluation is carried out.

This structure being a hybrid one, the weight adaptation in different modules becomes a challenging task as no such direct algorithm exists to do so. In the cascaded configuration, the FNN module is placed at the input end while the RNN module is at the output end. Here the updation of the weights of the RNN module can be carried out straightforward using the RTRL algorithm. However the weight adaptation for the FNN module (at the input) cannot be accomplished by using the BP algorithm directly without the estimate of the local gradients at all the FNN nodes. This requirement necessitates the determination of the local gradients at the RNN nodes first because RNN module follows the FNN module. As the RTRL algorithm does not provide any explicit estimate of the local gradients of the RNN nodes (termed as pseudo-local gradient) directly, problem is encountered in updating weights of the FNN module using the BP algorithm. This lacuna has motivated the present research in developing a new strategy, discussed elaborately in the following subsection.

### 5.1.1 Description of the proposed structure

The proposed structure, as shown in Figure 5.2, has  $nx$  external input connections,  $nf$  neurons in the FNN module and  $nr$  processing units (nodes) in the RNN module. The output vector  $z(n)$  of the FNN module and one step delayed output vector  $y(n)$  of the RNN module are concatenated to form an input vector  $u(n)$  to the RNN module, whose  $l^{th}$  element at time index  $n$  is denoted by  $u_l(n)$ . Let  $A$  denote the set of indices  $k$  for which  $y_k(n)$  is the output of  $k^{th}$  neuron of the RNN module and let  $B$  denote the set of indices  $f$  for which  $z_f(n)$  is the output of  $f^{th}$  neuron of the FNN module[06].

$$u_l(n) = \begin{cases} y_k(n) & \text{if } k \in A, A = \{1,2,\dots, nr\} \\ z_f(n) & \text{if } f \in B, B = \{1,2,\dots, nf\} \end{cases} \quad (5.1)$$

$W$  denotes  $nr$  by  $(nf + nr)$  weight matrix of RNN module and  $V$  denotes  $nf$  by  $nx$  weight matrix of FNN module.

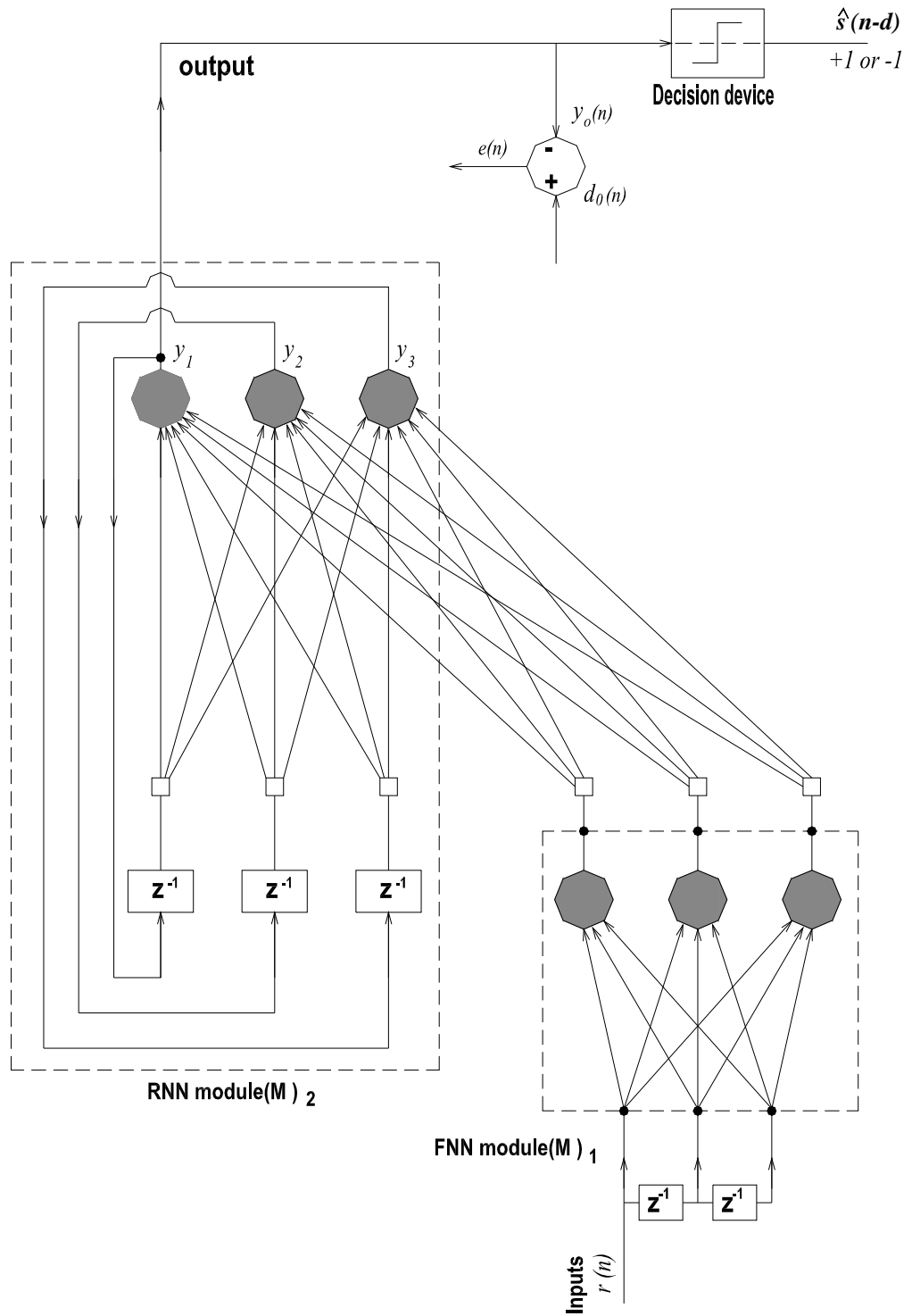


Figure 5.2: FNN-RNN cascaded equaliser structure



The input signal to the proposed equaliser structure is represented by a  $m \times 1$  vector given by

$$\mathbf{x}(n) = [r(n), r(n-1), \dots, r(n-m+1)]^T \quad (5.2)$$

The response of  $f^{th}$  neuron in the FNN module at time index  $n$  is given by

$$z_f(n) = \mathbf{F} \left\{ \sum_{i=1}^{nx} v_{fi}(n) \cdot x_i(n) \right\}, \quad 1 \leq f \leq nf \quad (5.3)$$

The output of  $k^{th}$  neuron of RNN module at time index  $n$  is evaluated as

$$y_k(n) = \mathbf{F} \{c_k(n)\} = \frac{1 - e^{-\phi \cdot c_k(n)}}{1 + e^{-\phi \cdot c_k(n)}} \quad (5.4)$$

where the net internal activity of neuron  $k$  is given by

$$c_k(n) = \sum_{l \in A \cup B} w_{kl}(n) \cdot u_l(n), \quad 1 \leq k \leq nr \quad (5.5)$$

$A \cup B$  is the union of sets  $A$  and  $B$ . Neurons with sigmoidal activation functions with slope parameter  $\phi$  are used in both FNN and RNN modules.  $C$  represents a set of visible neurons that provides externally reachable units. The output of  $j^{th}$  neuron of RNN module is  $y_j(n)$ , which provides the estimated output of the hybrid structure. The remaining neurons of the processing layer of RNN module are considered to be hidden. Let  $d_o(n)$  denotes the desired value of the equaliser output at time index  $n$  and the error at this time index is calculated as

$$e_j(n) = d_o(n) - y_j(n), \quad j \in C \quad (5.6)$$

The cost function at time index  $n$  is defined as

$$J(n) = \frac{1}{2} \sum_{j \in C} e_j(n)^2 \quad (5.7)$$

The objective here is to minimise the cost function i.e. to change the weights in the direction that minimizes  $J(n)$ .

### 5.1.2 Development of a novel concept proposed for network adaptation

A new concept developed in this present work to update the weights of the cascaded configuration is based on an *equivalence approach*, which has been explained sequentially as mentioned below:

- (i) The change in connection weights of RNN module using RTRL algorithm is evaluated first.
- (ii) Then these changes in the connection weights are taken as reference.
- (iii) Assuming that the BP algorithm is applied to calculate these changes in weights, the evaluation of corresponding local gradient at nodes of RNN module is carried out.

In order to provide a better understanding, a mathematical treatment of the above concept has been discussed below.

In this cascaded configuration while BP algorithm is applied to update the weights of the FNN module, the RTRL algorithm is used to update the weights of the RNN module. Application of RTRL algorithm involves primarily the evaluation of sensitivity parameter, a triply indexed set of variables  $\{p_{kl}^j\}$  defined as [06]

$$p_{kl}^j(n) = \frac{\partial y_j(n)}{\partial w_{kl}(n)}, \quad k \in A \quad \text{and} \quad l \in A \cup B \quad (5.8)$$

In the beginning,  $p_{kl}^j(n)$  is evaluated as follows.

$$p_{kl}^j(n+1) = F' \{c_j(n)\} \left[ \sum_{i \in A} w_{ji}(n) \cdot p_{kl}^i(n) + \partial_{kj} \cdot u_l(n) \right] \quad (5.9)$$

$$\text{with initial condition } p_{kl}^j(0) = 0 \quad \text{and} \quad F' \{c_j(n)\} = \{1 - y_j(n+1)^2\}(\phi/2) \quad (5.10)$$

and  $\partial_{kj}$  is termed as *Kronecker delta* as given by,

$$\partial_{kj} = 1 \text{ for } j = k \text{ and zero otherwise.} \quad (5.11)$$

The incremental change in connection weights of the RNN module is determined by the following expression.

$$\Delta w_{kl}(n) = \lambda \sum_{j \in C} e_j(n) p_{kl}^j(n), \quad (5.12)$$

where  $\lambda$  is the learning-rate parameter.

Now if BP algorithm would have been applied here, to obtain the same weight change  $\Delta w_{kl}(n)$  (chosen as reference) for the RNN module, then the mathematical expression would have become

$$\Delta w_{kl}(n) = \lambda \cdot \delta_{mn-node_k}(n) \cdot z_l(n) \quad (5.13)$$

Following the equivalence approach, the estimate of the pseudo-local gradient at node  $k$  of RNN module at time index  $n$  has been evaluated as

$$\delta_{mn-node_k}(n) = \frac{\sum_{i \in A} e_j(n) \cdot p_{kl}^i(n)}{z_l(n)} \quad (5.14)$$

The BP algorithm can now be used to update the connection weights of FNN module.

**Example:**

An example has been cited with detailed analysis as given below in order to provide a clear understanding about the development of the mentioned approach.

A hybrid structure consisting of a single layer FNN module with two neurons cascaded with a RNN module with two recurrent units is shown in Figure 5.2.1. The node1 of the RNN module has been taken as the only visible unit with output  $y_1(n)$ .

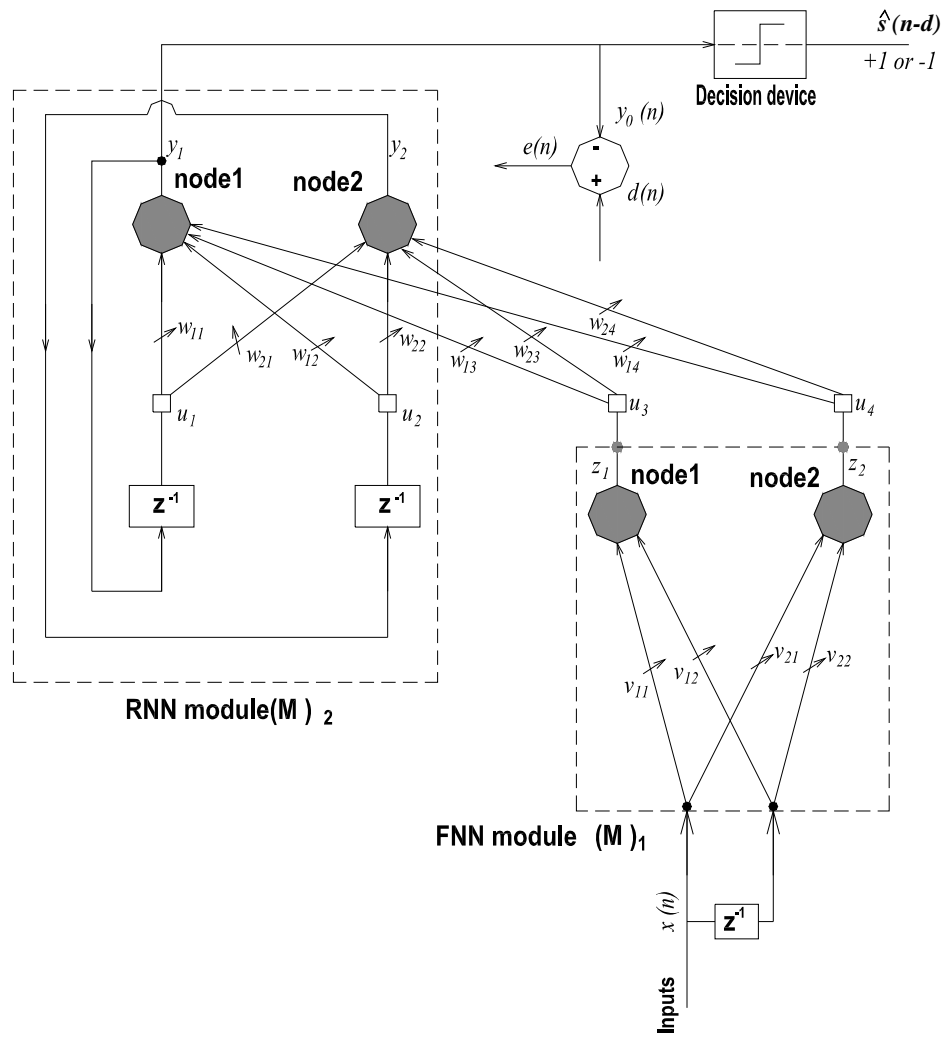


Figure 5.2.1: An example of proposed FRCS equaliser

By comparing  $y_1(n)$  with the desired value  $d_o(n)$  of equaliser output at time index  $n$ , error  $e_1(n)$  is computed. Based on the estimate of  $e_1(n)$  connection weights of RNN module are updated using RTRL algorithm. Implementation of the equivalence approach, as proposed, is initiated by choosing the incremental change in weight  $\Delta w_{23}(n)$  as the reference where subscripts 2 and 3 denote node 2 of RNN module and node 1 of FNN module respectively.

According to RTRL algorithm,

$$\Delta w_{23}(n) = \lambda \cdot e_1(n) \cdot p_{23}^1(n) \quad (5.15)$$

where in  $p_{23}^1(n)$  the superscript 1 represents only one output ( $j=1$ ) w.r.t. Equation(5.8).

If BP algorithm would have been used, to obtain the same weight change, the mathematical expression would have become

$$\Delta w_{23}(n) = \lambda \cdot \delta_{rnn-node_2}(n) \cdot z_1(n) \quad (5.16)$$

The application of proposed approach helps us to evaluate the *pseudo-local gradient* at node 2 of RNN module as given by

$$\delta_{rnn-node_2}(n) = \frac{e_1(n) \cdot p_{23}^1(n)}{z_1(n)} \quad (5.17)$$

Proceeding in the similar manner as mentioned above and taking incremental change in connection weight  $\Delta w_{24}(n)$  as reference, where subscripts 2 and 4 denote node 2 of RNN module and node 2 of FNN module respectively, the *pseudo-local gradient* of node 2 of the RNN module can be expressed as

$$\delta_{rnn-node_2}(n) = \frac{e_1(n) \cdot p_{24}^1(n)}{z_2(n)} \quad (5.18)$$

As only two nodes have been considered in the FNN module of the equaliser structure, the *pseudo-local gradient* at node 2 of the RNN module is estimated finally as the averaged one and is given by

$$\delta_{rnn-node_2}(n) = \frac{e_1(n)}{2} \left[ \frac{p_{23}^1(n)}{z_1(n)} + \frac{p_{24}^1(n)}{z_2(n)} \right] \quad (5.19)$$

Once the *pseudo-local gradient* at all nodes of the RNN module have been evaluated, the task becomes easier to estimate the local gradients at all nodes of FNN module by applying the BP algorithm directly. Hence the local gradient of node 1 of the FNN module is given by the equation

$$\delta_{fnn-node_1}(n) = \{1 - z_1(n)\}^2 \cdot (\phi / 2) \left[ \delta_{rnn-node_1}(n) w_{13}(n) + \delta_{rnn-node_2}(n) w_{23}(n) \right] \quad (5.20)$$

The calculation of this local gradient helps in updating the weights connected to that particular node of the FNN module to the external inputs. For example, based on  $\delta_{fnn-node_1}(n)$ , the incremental changes in connection weights are given by

$$\Delta v_{11}(n) = \eta \cdot \delta_{fnn-node_1}(n) \cdot x_1(n) \quad (5.21)$$

and

$$\Delta v_{12}(n) = \eta \cdot \delta_{fnn-node_1}(n) \cdot x_2(n) \quad (5.22)$$

where  $\eta$  is a learning-rate parameter for updation of the FNN structure.

Further, generalized mathematical expression w.r.t. Equation (5.20) and Equation (5.21) is given by

$$\delta_{fnn-node_f}(n) = \{1 - z_f(n)^2\} \cdot (\phi / 2) \left[ \sum_{k=1}^{nr} \delta_{rnn-node_k}(n) \cdot w_{kl}(n) \right], \quad 1 \leq f \leq nf \text{ and } l = nr + f \quad (5.23)$$

$$\Delta v_{fi}(n) = \eta \cdot \delta_{fnn-node_f}(n) \cdot x_i(n), \quad 1 \leq f \leq nf \text{ and } 1 \leq i \leq nx \quad (5.24)$$

### 5.1.3 Training algorithm

In summary, the proposed algorithm for updating the connection weights in the hybrid structure, proceeds as follows:

- The initial values of connection weights in the FNN and RNN modules are chosen from a set of uniformly distributed random numbers. The sensitivity parameters for the RNN module,  $\{p_{kl}^j\}$  are initialized to zero.
- The final output of hybrid structure is computed using Equation (5.4).
- The sensitivity parameters  $\{p_{kl}^j\}$  for all appropriate  $j, k$  and  $l$  are evaluated as per Equations (5.8) and (5.9).
- The error signal  $e_f(n)$ , which is the difference between the desired response  $d_o(n)$  and the final estimated output  $y_f(n)$  is computed.
- Then the local gradients at all nodes of the RNN module ( $\delta_{rnn-node}$ ) are computed using Equation (5.14).
- Further, the local gradients at all nodes of the FNN module ( $\delta_{fnn-node}$ ) are evaluated following Equation (5.23).
- While the RTRL algorithm using Equation (5.13) computes the incremental weight changes  $\Delta w_{kl}(n)$ , the BP algorithm using Equation (5.24) evaluates the incremental weight changes  $\Delta v_{fi}(n)$ . Finally, the connection weights in FNN and RNN modules are updated as follows.

$$w_{kl}(n+1) = w_{kl}(n) + \Delta w_{kl}(n), \quad 1 \leq k \leq nr \text{ and } 1 \leq l \leq (nr + nf) \quad (5.25)$$

$$v_{fi}(n+1) = v_{fi}(n) + \Delta v_{fi}(n), \quad 1 \leq f \leq nf \quad (5.26)$$

- This process of weight updation is continued till the network is fully trained achieving a desired value of performance index.

## 5.2 Hierarchical knowledge reinforced FNN-RNN cascaded (HKFRCS) equaliser

A highly efficient structure is designed employing the *hierarchical knowledge reinforcement* to the cascaded network discussed in the Section 5.1. In the present cascaded framework the external inputs are also fed directly to the RNN module. The external inputs are already being fed to the FNN module for preprocessing as described in Section 5.1. It is expected that by feeding more information to the final processing layer RNN module (original information in addition to the output from the FNN module), its knowledge base is further strengthened which helps in improving the decision making capability. Hence, significant enhancement in the performance of the proposed equaliser is observed. The network has to be adapted during the training phase, then weights are frozen and its BER performance evaluation is carried out. The training algorithm for this neural structure is presented in the following section.

### 5.2.1 Training algorithm

The proposed structure, as shown in Figure 5.3 comprises of  $nx$  external inputs,  $nf$  neurons in the FNN module and  $nr$  processing units in the RNN module.

The sequence of operation followed for the application of the training algorithm is given as below.

- The initial values of all the synaptic weights of FNN and RNN modules are chosen from a set of uniformly distributed random numbers. The sensitivity parameters  $\{p_{kl}^j\}$  of the RNN module are initialized to zero.
- The input signal to the proposed equaliser structure is represented by a  $m \times 1$  vector  $\mathbf{x}(n) = [r(n), r(n-1), \dots, r(n-m+1)]^T$ . First, the output of the FNN module is calculated and for the  $f^{th}$  neuron at time index  $n$ , the output  $z_f(n)$  is given by,

$$z_f(n) = \mathbf{F} \left\{ \sum_{i=1}^{nx} v_{fi}(n) \cdot x_i(n) \right\}, \quad 1 \leq f \leq nf \quad (5.27)$$

- Next the output vector  $\mathbf{z}(n)$  of the FNN module, one step delayed output vector  $\mathbf{y}(n)$  of the RNN module and the external input vector to FNN  $\mathbf{x}(n)$  are taken together to form the input vector  $\mathbf{u}(n)$  to the RNN module whose  $l^{th}$  element is given by  $u_l(n)$ .

$$u_l(n) = \begin{cases} y_k(n), & 1 \leq k \leq nr \\ z_f(n), & nr+1 \leq f \leq nf \\ x_i(n), & nr+nf \leq i \leq nx \end{cases} \quad (5.28)$$

The output of  $k^{th}$  neuron of the RNN module at time index  $n$  is defined as,

$$y_k(n) = \frac{1 - e^{-\phi \cdot c_k(n)}}{1 + e^{-\phi \cdot c_k(n)}}, \quad (5.29)$$

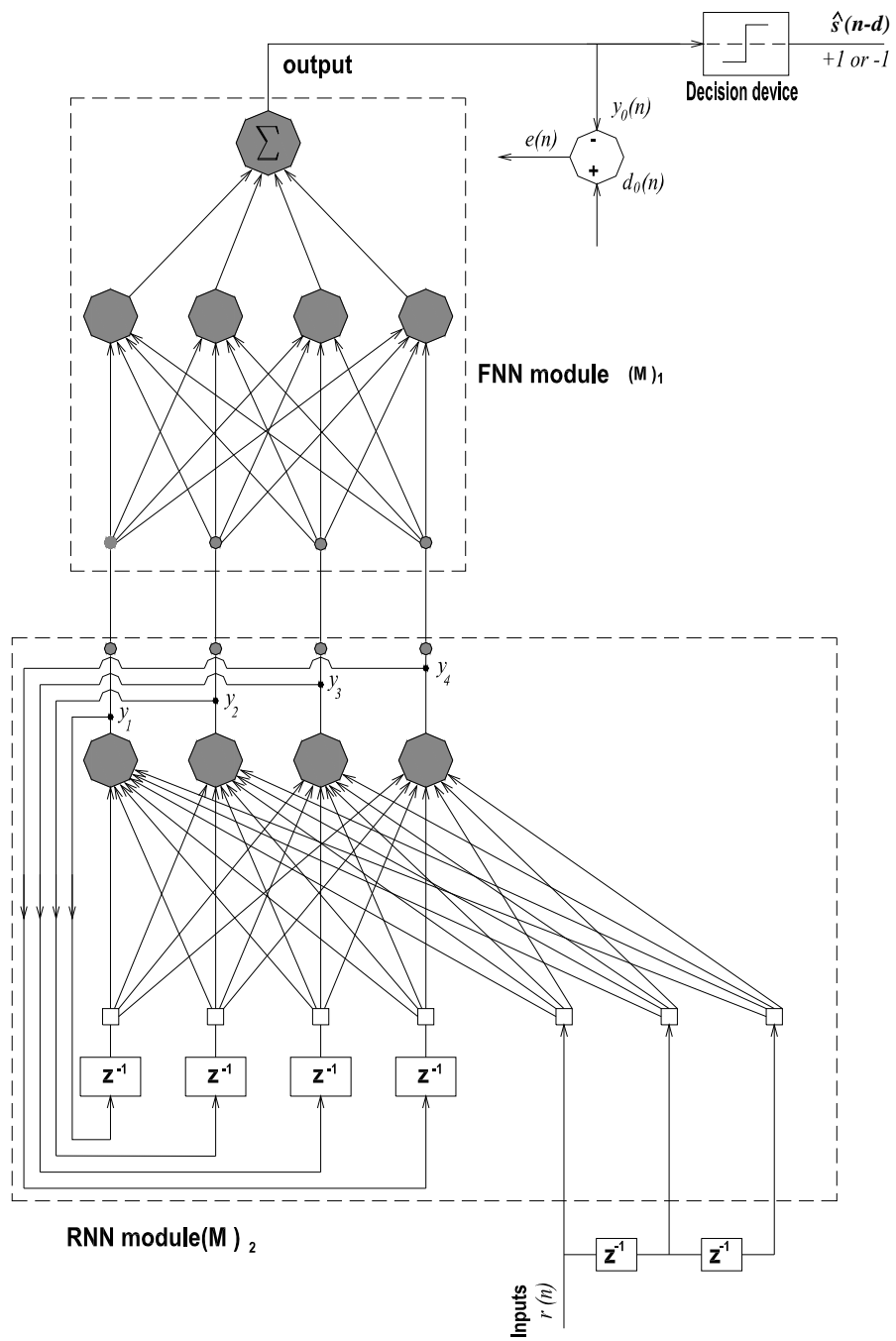


Figure 5.4: RNN-FNN cascaded equaliser structure

$$\text{where the net internal activity } c_k(n) = \sum_{l=1}^{nr+nf+nx} w_{kl}(n) \cdot u_l(n) \quad (5.30)$$

Sigmoidal activation function ( $F$ ) with slope parameter  $\phi$  has been considered for the neurons of both the FNN and RNN modules.

- The estimated output is  $y_j(n)$ , the output of  $j^{\text{th}}$  neuron of RNN module, is compared with the desired value  $d_o(n)$  to calculate the output error at time index  $n$  as given by

$$e_j(n) = d_o(n) - y_j(n) \quad (5.31)$$

- The sensitivity parameter  $\{p_{kl}^j\}$  is defined as

$$p_{kl}^j(n) = \frac{\partial y_j(n)}{\partial w_{kl}(n)}, \quad 1 \leq k \leq nr \text{ and } 1 \leq l \leq (nr+nf+nx) \quad (5.32)$$

The evaluation of this parameter is carried out as follows:

$$p_{kl}^j(n+1) = F' \{c_j(n)\} \left[ \sum_{i=1}^{nr} w_{ji}(n) \cdot p_{ki}^i(n) + \partial_{kj} u_l(n) \right], \quad (5.33)$$

where the first derivative is given by

$$F' \{c_j(n)\} = \{1 - y_j(n+1)^2\} (\phi/2) \quad (5.34)$$

and  $\partial_{kj}$  is termed as *Kronecker delta* which is already defined in (5.11)

- Following the equivalence concept already explained in Section 5.1.2, the pseudo- local gradient of  $k^{\text{th}}$  node in the RNN module is given as

$$\delta_{mn-node_k}(n) = \frac{e_j(n)}{nf} \left[ \sum_{l=nr+1}^{nf+nr} \frac{p_{kl}^j(n)}{z_{l-nr}(n)} \right], \quad 1 \leq k \leq nr \quad (5.35)$$

- The local gradient at each node of the FNN module is evaluated using the BP algorithm by propagating back the error through all the connections from the nodes of the RNN module to the corresponding node of the FNN module and is given by

$$\delta_{fnn-node_f}(n) = \{1 - z_f(n)^2\} (\phi/2) \left[ \sum_{k=1}^{nr} \delta_{mn-node_k}(n) \cdot w_{kf}(n) \right], \quad (5.36)$$

$$1 \leq f \leq nf \text{ and } l=(nr+f)$$

- The connection weights of the FNN module are updated using BP algorithm and basic RTRL algorithm is utilised for updating the weights of the RNN module. The incremental change in the connection weights  $\Delta v_{fi}(n)$  and  $\Delta w_{kl}(n)$  of the RNN and FNN modules respectively are given by the following mathematical expressions.

$$\Delta v_{fi}(n) = \eta \cdot \delta_{fnn-node_f}(n) \cdot x_i(n), \quad 1 \leq f \leq nf \text{ and } 1 \leq i \leq nx \quad (5.37)$$



$$\Delta w_{kl}(n) = \lambda \cdot e_j(n) \cdot p_{kl}^j(n), \quad 1 \leq k \leq nr \text{ and } 1 \leq l \leq (nr+nf+nx) \quad (5.38)$$

- Once the incremental weight changes have been estimated, the updated weights of both the FNN and RNN modules are given by

$$v_{fi}(n+1) = v_{fi}(n) + \Delta v_{fi}(n) \quad (5.39)$$

$$w_{kl}(n+1) = w_{kl}(n) + \Delta w_{kl}(n) \quad (5.40)$$

where  $\lambda$  and  $\eta$  are the *learning-rate parameters* of weights adaptation in RNN and FNN modules respectively.

- This process of recursion continues till a given performance index is achieved.

### 5.3 RNN-FNN cascaded (RFCS) equaliser

Another variant in the cascaded configuration has been proposed in this research work as shown in the Figure 5.4, which resembles that of the structure already discussed in Section 5.1 with the only exception that two modules (RNN and FNN) are being swapped. In this case, the output of the RNN module is fed to the input of an FNN module. If this proposed structure is investigated thoroughly, then it can be inferred that an intermediate decision feedback mechanism has been embedded into the network paradigm, because the RNN module exists at the input end of this heterogeneous combination. The proposed equaliser is to be trained first before its BER performance evaluation is done. In this network configuration the weight adaptation does not pose any problem as has been witnessed in Section 5.1. The connection weights of FNN module can be updated using the BP algorithm as the local gradients of the neurons at all its nodes are computed from the output error directly. Thus employing the estimates of those local gradients, the errors at the nodes of the RNN module can be directly computed utilising error back propagation approach. Thereafter, the existing RTRL algorithm can be applied directly to update the connection weights in the RNN module. The step by step procedure of the training algorithm employed here is explained in the following section.

#### 5.3.1 Training algorithm

The proposed configuration is shown in Figure 5.4. It has  $nx$  external input connections,  $nr$  processing units in the RNN module and  $nf$  neurons in the FNN module.

- The initial values of all connection weights of both the FNN and the RNN modules are chosen from a set of uniformly distributed random numbers. The initial value of the sensitivity parameters  $\{p_{kl}^j\}$  of the the RNN module are set to zero.

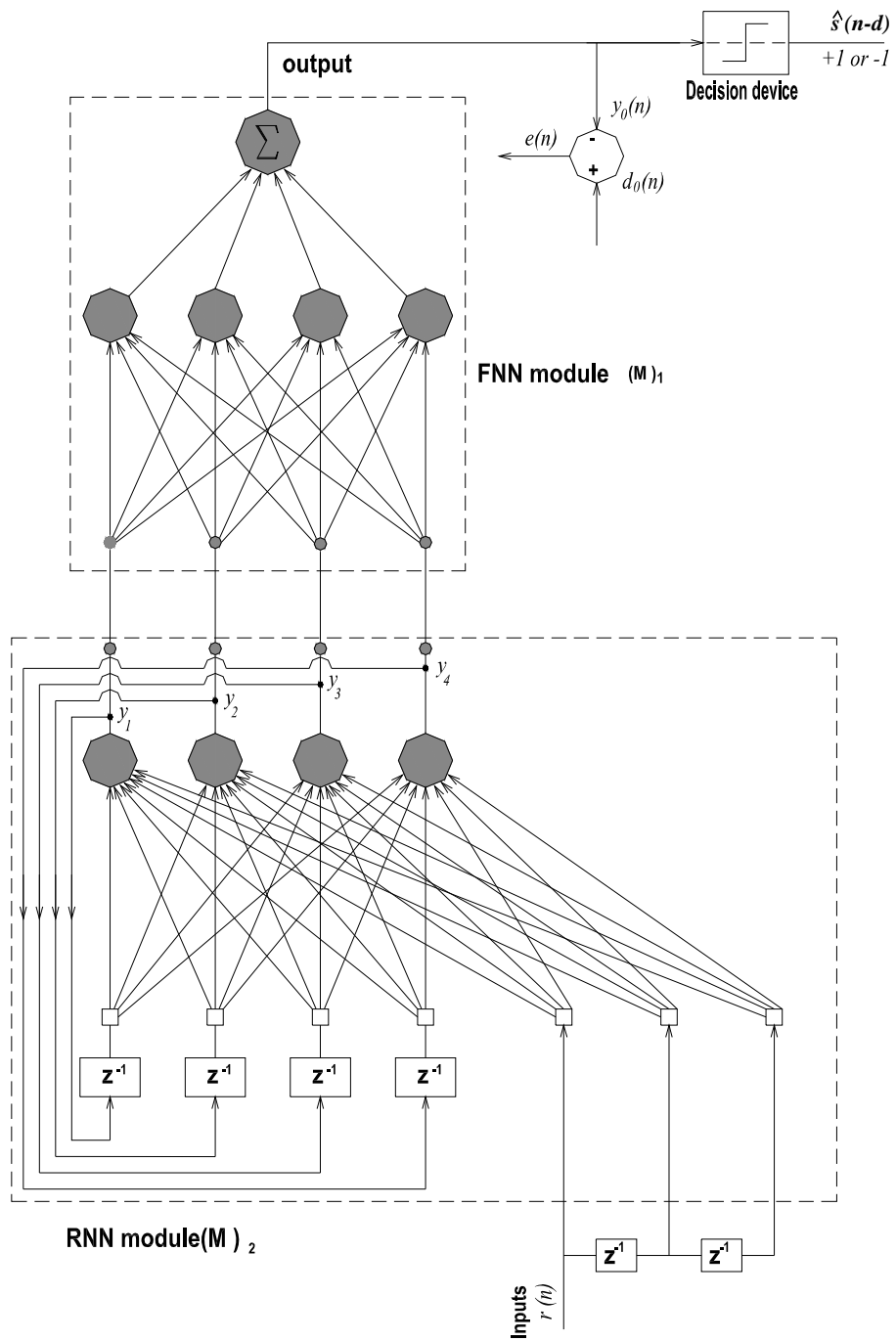


Figure 5.4: RNN-FNN cascaded equaliser structure

- The input signal to the proposed equaliser structure is represented by a  $m \times 1$  vector  $\mathbf{x}(n) = [r(n), r(n-1), \dots, r(n-m+1)]^T$ . An input vector  $\mathbf{u}(n)$  applied to the RNN module is formed whose  $l^{th}$  element,  $u_l(n)$  is defined as

$$u_l(n) = \begin{cases} y_k(n), & 1 \leq k \leq nr \\ x_k(n), & 1 \leq k \leq nx \end{cases} \quad \text{for } 1 \leq l \leq (nx+nr) \quad (5.41)$$

- The output of the  $k^{th}$  neuron of the RNN module at time index  $n$  is computed as

$$y_k(n) = \frac{1 - e^{-\phi \cdot c_k(n)}}{1 + e^{-\phi \cdot c_k(n)}}, \quad (5.42)$$

where the net internal activity of the  $k^{th}$  neuron

$$c_k(n) = \sum_{l=1}^{nf+nr} w_{kl}(n) \cdot u_l(n), \quad 1 \leq k \leq nr \quad (5.43)$$

where  $\mathbf{W}$  denotes  $nr$  by  $(nx + nr)$  weight matrix of the RNN module.

- In the cascaded framework of this hybrid configuration, the outputs from the RNN module are fed as the inputs to the FNN module. Here a single layer FNN module with  $nf$  neurons is considered only restricting the structural complexity.
- The output of the  $f^{th}$  neuron of the FNN module at time index  $n$  is given by,

$$z_f(n) = \mathbf{F} \left\{ \sum_{j=1}^{nout} v_{ff}(n) \cdot y_j(n) \right\}, \quad 1 \leq f \leq nf \quad (5.44)$$

where all the processing units of the RNN module are considered as externally reachable units. Sigmoidal activation functions ( $\mathbf{F}$ ) with slope parameter  $\phi$  are chosen for all the neurons of both FNN and RNN modules. Evaluation of final output of the hybrid structure  $y_j(n)$  at time index  $n$  is carried out by providing a summing unit.

$$y_j(n) = \sum_{f=1}^{nf} g_f(n) z_f(n) \quad (5.45)$$

where  $\mathbf{g}$  denotes the weight matrix at the output end.

- The error  $e_j(n)$  at the output is computed by comparing its estimated value with the desired one of the equaliser  $d_o(n)$ .

$$e_j(n) = d_o(n) - y_j(n) \quad (5.46)$$

- With the knowledge of this error  $e_j(n)$ , the local gradient of error at each neuron of the FNN module is calculated.

$$\delta_{fnn-node_f}(n) = e_j(n) g_f(n) \{1 - z_f(n)^2\} (\phi / 2), \quad 1 \leq f \leq nf \quad (5.47)$$

- Thereafter the error at each node of the RNN module is evaluated as

$$err_{mn-node_j}(n) = \sum_{f=1}^{nf} \delta_{fnn-node_f}(n) \cdot v_{ff}(n), \quad 1 \leq j \leq nr \quad (5.48)$$

- Then updation of the sensitivity parameter is done as follows

$$p_{kl}^j(n+1) = F' \{c_j(n)\} \cdot \left[ \sum_{i=1}^{nr} w_{ji}(n) \cdot p_{kl}^i(n) + \partial_{kj} \cdot u_l(n) \right] \quad (5.49)$$

$$\text{where } F' \{c_j(n)\} = \{1 - y_j(n+1)^2\}(\phi/2) \quad (5.50)$$

and  $\partial_{kj}$  is a *Kronecker delta* already defined in Equation (5.11)

- The incremental change in weights of both the FNN and RNN modules are evaluated using BP algorithm and RTRL algorithms respectively and corresponding weight updation is carried out.

$$\Delta v_{ff}(n) = \eta \cdot \delta_{fnn-node_f}(n) \cdot y_j(n), \quad 1 \leq f \leq nf \text{ and } 1 \leq j \leq nr \quad (5.51)$$

$$v_{ff}(n+1) = v_{ff}(n) + \Delta v_{ff}(n) \quad (5.52)$$

$$\Delta w_{kl}(n) = \lambda \cdot \sum_{j=1}^n err_{mn-node_j}(n) \cdot p_{kl}^j(n), \quad 1 \leq k \leq nr \text{ and } 1 \leq l \leq (nx+nr) \quad (5.53)$$

$$w_{kl}(n+1) = w_{kl}(n) + \Delta w_{kl}(n) \quad (5.54)$$

$$\Delta g_f(n) = \theta \cdot e_j(n) \cdot z_f(n), \quad 1 \leq f \leq nf \quad (5.55)$$

$$g_f(n+1) = g_f(n) + \Delta g_f(n) \quad (5.56)$$

where  $\lambda$  and  $\eta$  are the *learning-rate parameters* of the RNN and FNN modules respectively.  $\theta$  is the learning-rate parameter of the adaptable weights in the output layer.

- This process of weight updation continues till the network is fully trained

## 5.4 RNN –Transform cascaded (RTCS) equaliser

The analysis carried out in detail in Section 4.3 with reference to the transform domain based FNN has motivated to extend this concept into the RNN framework. This cascaded configuration although similar as that reported in Section 5.3, is different in the sense that the FNN module is replaced with a transform block. Here, a discrete cosine transform followed by normalisation block is cascaded with an RNN module at the output end as given in Figure 5.5.

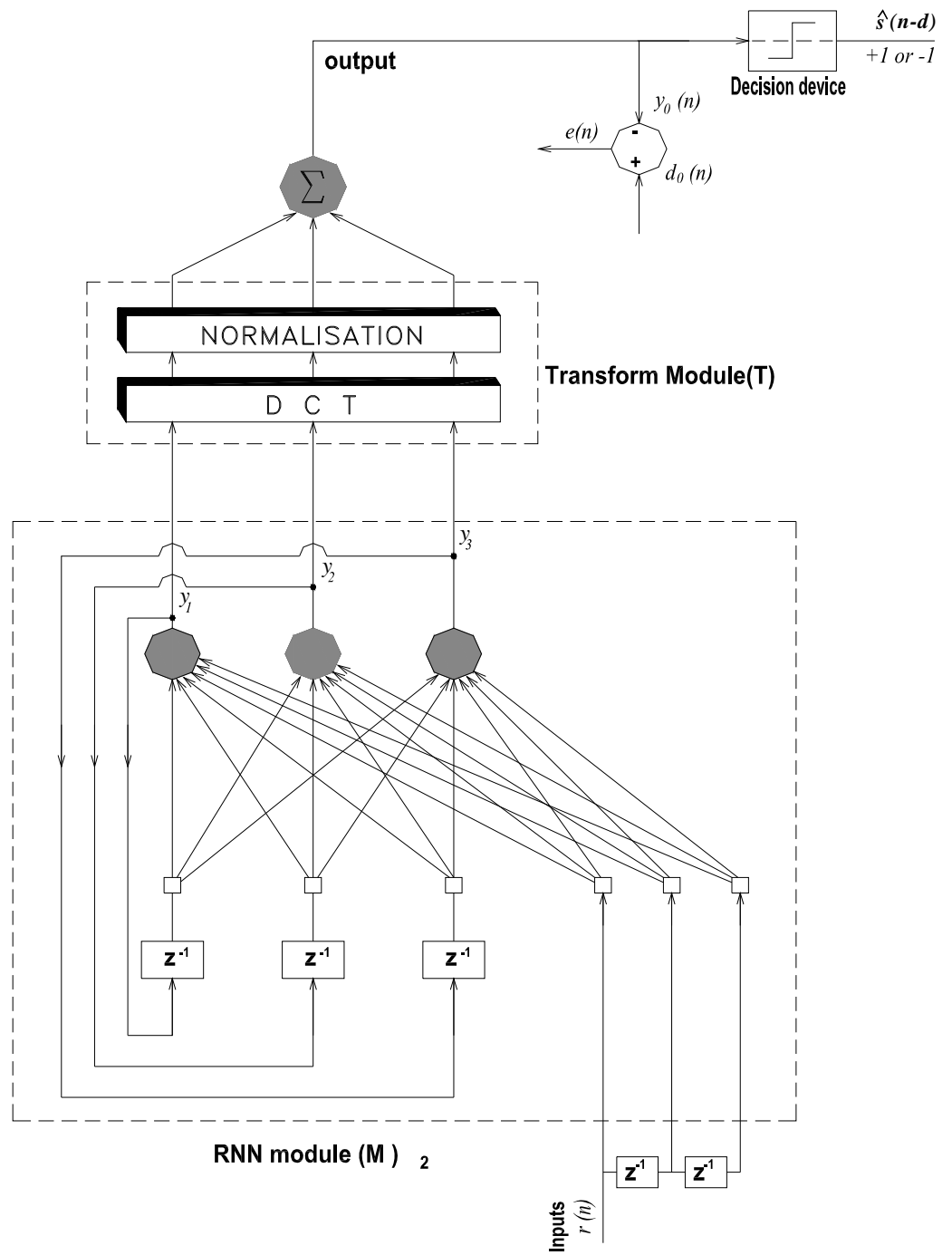


Figure 5.5: RNN-Transform cascaded equaliser structure

As far as the choice of transform is concerned, *discrete cosine transform* (DCT) has come out a clear winner because of significant performance enhancement over its counterparts like DST, DHT etc., which has been observed during computer simulation. Power normalisation technique [102] is applied to the transformed signals as mentioned earlier in Section 4.3 of Chapter-4 and the final output of the proposed structure is evaluated as a weighted sum of all normalised signals by providing a summing unit. The equaliser has to be adapted during the training phase, then its weights are frozen and after that its BER performance evaluation is done.

In order to update all the connection weights of this cascaded framework during the training phase of the proposed equaliser, a novel idea has been developed based on propagation of the output error through the network in the light of the conventional BP algorithm, to obtain the estimate of the error at the output of the transform block. It is obvious that the transform block does not require any weight adaptation unlike the FNN module (as done in Section 5.3) as it consists of fixed weights, but the RNN module needs updation of the connection weights. Application of the standard RTRL algorithm necessitates the determination of errors at the nodes of the RNN module. But this estimate can not be accomplished directly by using the BP algorithm due to positioning of the transform block at the output end of the cascaded structure, so problem is encountered here in propagating the final output error backwards into the network. To circumvent this difficulty, an adhoc solution has been evolved. First the error estimation at the input end of the transform block is done from the knowledge of the error at its output by considering inverse discrete cosine transform. Then the connection weights in the RNN module are adapted based on the error calculated at its nodes. The mathematical expressions governing this concept are described in the subsequent section.

#### 5.4.1 Training algorithm

The proposed structure shown in Figure 5.5 consists of  $nr$  processing units in the RNN module with  $nx$  external inputs and a transform block. A step by step procedure has been adopted to update the weights of the network as mentioned below.

- The connection weights of the cascaded structure are set to small random numbers that are uniformly distributed. Sensitivity parameters  $\{p_{kl}^j\}$  of all RNN nodes are initialised to zero.
- The input signal to the proposed equaliser structure is represented by a  $m \times 1$  vector  $\mathbf{x}(n) = [r(n), r(n-1), \dots, r(n-m+1)]^T$ . Input signal vector to the RNN module is defined as  $\mathbf{u}(n)$ ,  $l^{\text{th}}$  element of which is

$$u_l(n) = \begin{cases} y_j(n), & 1 \leq j \leq nr \\ x_i(n), & 1 \leq i \leq nx \end{cases} \quad \text{for } 1 \leq l \leq (nr+nx) \quad (5.57)$$

The output of  $j^{\text{th}}$  neuron of the RNN module at time index  $n$  is given by

$$y_j(n) = \frac{1 - e^{-\phi \cdot c_j(n)}}{1 + e^{-\phi \cdot c_j(n)}} \quad (5.58)$$

where the net internal activity is described by

$$c_j(n) = \sum_{l=1}^{nx+nr} w_{kl}(n) u_l(n), \quad 1 \leq k \leq nr \quad (5.59)$$

where  $\mathbf{W}$  denotes  $nr$  by  $(nx + nr)$  weight matrix of the RNN module. Sigmoidal activation functions ( $\mathbb{F}$ ) with slope parameter  $\phi$  for neurons of the RNN module have been considered. Input signal vector to the transform block can be expressed as  $z(n)$ , whose  $j^{\text{th}}$  element is denoted as,

$$z_j(n) = y_j(n), \quad j = nr \quad (5.60)$$

Here all the processing units of the RNN module act as visible units giving externally reachable outputs. The  $j^{\text{th}}$  element of the output from the transform block is defined as

$$z_{\mathcal{T}j}(n) = DCT \{z_j(n)\} = \mathcal{T} z_j(n) \quad (5.61)$$

where the elements of the transform matrix  $\mathcal{T}$  for the *Discrete Cosine Transform* (DCT) has been already explained in Equation (4.51). Transformed signal  $z_{\mathcal{T}j}(n)$  are then normalised by the square root of their power  $\mathcal{B}_j(n)$  as discussed in Section 4.3.

The  $j^{\text{th}}$  element of the normalised signal,

$$z_{\mathcal{N}j}(n) = \frac{z_{\mathcal{T}j}(n)}{\sqrt{\mathcal{B}_j(n) + \varepsilon}} \quad (5.62)$$

$$\text{and } \mathcal{B}_j(n) = \gamma \mathcal{B}_j(n-1) + (1 - \gamma) z_{\mathcal{T}j}^2(n) \quad (5.63)$$

The scaling parameter  $\gamma \in [0, 1]$ . The small constant  $\varepsilon$  is introduced to avoid numerical instabilities when signal power  $\mathcal{B}_j(n)$  is close to zero.

The final output of the hybrid structure at time index  $n$ ,  $y_o(n)$  is expressed as the weighted sum of all normalised signals from the transform block.

$$y_o(n) = \sum_{j=1}^{nr} g_j(n) \cdot z_{\mathcal{N}j}(n) \quad (5.64)$$

where  $\mathbf{g}$  denotes the weight matrix at the output end of the proposed network.

- The error at the equaliser output at time index  $n$  is given by,

$$e(n) = d_o(n) - y_o(n) \quad (5.65)$$

With the knowledge of the output error, the errors at all the nodes of RNN module can be evaluated in order to facilitate the updation of weights using RTRL algorithm. But this is not possible directly as already explained before and hence an innovative technique has been employed to tackle this situation.

- At first the error  $e(n)$  is back propagated through various connection paths. Then the error at the  $j^{\text{th}}$  output of normalisation block is computed as given by

$$e_{\mathcal{N}_j}(n) = e(n).g_j(n), \quad 1 \leq j \leq nr \quad (5.66)$$

- The error terms at the output of the transform block  $\delta_{\mathcal{T}_j}(n)$  can be calculated using Equation (4.57) following the approach as explained in Section 4.3.

$$\begin{aligned} \delta_{\mathcal{T}_j}(n) &= e_{\mathcal{N}_j}(n) \cdot \frac{\partial z_{\mathcal{N}_j}(n)}{\partial z_{\mathcal{T}_j}(n)} \\ &= e_{\mathcal{N}_k}(n) (z_{\mathcal{N}_k}(n) / z_{\mathcal{T}_k}(n)) \{1 - (1 - \gamma) z_{\mathcal{T}_k}^2(n)\} \end{aligned} \quad (5.67)$$

- Further, to propagate the error back through the transform block and to estimate the error magnitudes at the input side of the transform block, *Inverse Discrete Cosine Transform* (IDCT) is applied. This provides an estimate of the error at the input end of the transform block and the error at the  $j^{\text{th}}$  processing unit of the RNN module at time index  $n$  is given by

$$err_{mn-node_j}(n) = IDCT \{ \delta_{\mathcal{T}_j}(n) \} \quad (5.68)$$

- The sensitivity parameters  $\{p_{kl}^j\}$  are updated as follows

$$\begin{aligned} p_{kl}^j(n+1) &= F' \{c_j(n)\} \left[ \sum_{i=1}^{nr} w_{ji}(n) \cdot p_{kl}^i(n) + \partial_{kj} u_l(n) \right], \\ &1 \leq j \leq nr, 1 \leq k \leq nr \quad \text{and} \quad 1 \leq l \leq (nr + nx) \end{aligned} \quad (5.69)$$

where  $F' \{c_j(n)\}$  is given in Equation (5.50) and  $\partial_{kj}$  is defined in Equation (5.11).

- While the incremental weight change  $\Delta g_j(n)$  is calculated using BP algorithm, RTRL algorithm computes the incremental weight change  $\Delta w_{kl}(n)$ .

$$\Delta g_j(n) = \theta \cdot e(n) \cdot z_{\mathcal{N}_j}(n), \quad 1 \leq j \leq nr \quad (5.70)$$

$$\Delta w_{kl}(n) = \lambda \cdot \sum_{j=1}^{nr} err_{mn-node_j}(n) \cdot p_{kl}^j(n), \quad 1 \leq k \leq nr \quad \text{and} \quad 1 \leq l \leq (nr + nx) \quad (5.71)$$

where  $\lambda$  and  $\theta$  are the learning-rate parameters of the RNN module and the output layer respectively.

The connection weights are updated as given below:

$$g_j(n+1) = g_j(n) + \Delta g_j(n) \quad (5.72)$$

$$w_{kl}(n+1) = w_{kl}(n) + \Delta w_{kl}(n) \quad (5.73)$$



- The recursion process of updating weights of the cascaded network continues till a predefined condition is achieved as mentioned earlier.

## 5.5 Fuzzy tuned recurrent neural network (FZTUNRNN) equaliser

A concept is evolved using fuzzy controller approach to tune the slope of sigmoidal activation function of the RNN nodes so that the network is made adaptive. The entire methodology adopted here is exactly identical to the approach already discussed in Section 4.4 of Chapter 4, where a *fuzzy logic controller technique* is employed to adjust the slope parameter  $\phi$  of the sigmoidal activation function based on the knowledge of the local gradient of the node in the conventional RNN framework.

### 5.5.1 Details of proposed method

The role of slope ( $\phi$ ) of the sigmoidal activation function used in each of the processing unit of RNN is extremely important as it contributes to decision making ability of that node. As the value of slope is changed, the decision of expert node is altered as the nonlinear mapping changes. The proposed structure is built around this concept of tuning the slope parameter. The proposed equaliser has to be adapted during the training phase, then weights are frozen and its BER performance evaluation is done. While the basic RTRL algorithm updates the network weights, the sigmoid slope tuning by *fuzzy logic controller* approach [106,107], already explained in Section 4.4, adjusts the slope of the sigmoidal activation function at the same time index. Significant performance improvement and faster learning are observed using this new technique in a conventional RNN equaliser.

The recurrent node error term  $\delta_{mn-node_k}(n)$ , as evaluated in Equation (5.14), is referred to because direct estimate of the node error in RNN is not available and the error term change corresponding to the  $k^{\text{th}}$  node of the RNN block

$$\Delta \delta_{mn-node_k}(n) = \delta_{mn-node_k}(n) - \delta_{mn-node_k}(n-1) \quad (5.74)$$

are fed into the fuzzy controller block (FLC) as shown in Figure 5.6.

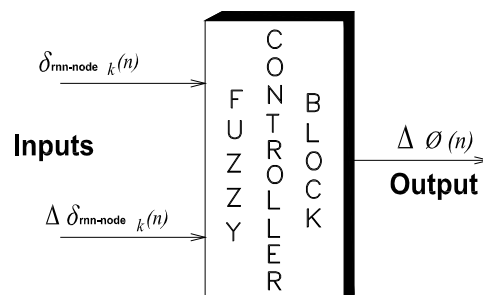


Figure 5.6: Fuzzy logic approach for tuning sigmoid slope

The fuzzy controller evaluates the control action required based on the past information of error and rate of change at the process output. Here depending on the inputs fed, the output generated from the FLC block is the change of slope  $\Delta\phi(n)$ . Then the slope of the activation function is updated using the mathematical expression in Equation (4.66) and its new value is incorporated in the next time index for the output calculation.

Here also for fuzzification seven categories of linguistic variables are declared namely LP, MP, SP, ZE, SN, MN and LN for both the inputs ( $\delta_{rmn-node_k}(n)$  and  $\Delta\delta_{rmn-node_k}(n)$ ) and the output  $\Delta\phi(n)$ . The membership functions are assumed to have *Gaussian type distribution* [108] and have fixed centres and widths. The fuzzified inputs are used to construct the rule base from the IF conditions [109] and the final output is obtained from the defuzzification of the THEN conditions.

## 5.6 Simulation study and the performance analysis of the proposed RNN based equalisers

Equalisation of different types of channel models (both linear and non-linear type) are attempted in order to establish the efficacy of the proposed equaliser structures based on RNN topology and to prove their robustness. It has been already reported in the literatures [32,34], that a two-unit, one input, one output RNN is a non-linear IIR model which is sufficient to model many communication channels. Considering this aspect, all the proposed cascaded equalisers in RNN framework are compared with a conventional RNN equaliser (CRNN) with two recurrent units and one external input sample from the channel output.

The configurations of all the proposed RNN based equalizers, illustrated in Figures 5.7a-e, indicate that their structural complexities have been kept low purposefully to maintain the basic objective of the research work. Table 5.1 gives a comparison between the various proposed configurations with the conventional RNN one. All the equalisers chosen have one external input only. In the proposed FRCS configuration, one neuron in the FNN module and one neuron in RNN module are considered. The proposed HKFRCS structure is the same as the previous one with the exception that the external input is also fed directly to the RNN node alongwith the output from the FNN node. The proposed RFCS structure also has one node in FNN module and one node in RNN module. Further the RTCS structure has two nodes in RNN module followed by a 2x2 DCT block with power normalisation and a summing unit at the output end. The proposed FZTUNRNN equaliser has the same structure as a conventional RNN one and provided with two nodes, having slope parameter adapted using the fuzzy logic control technique.

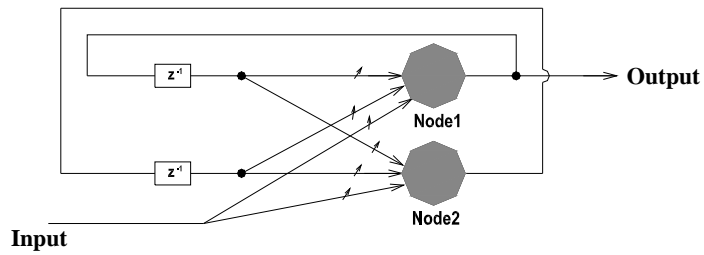


Figure 5.7a: Conventional RNN / Proposed FZTUNRNN structure

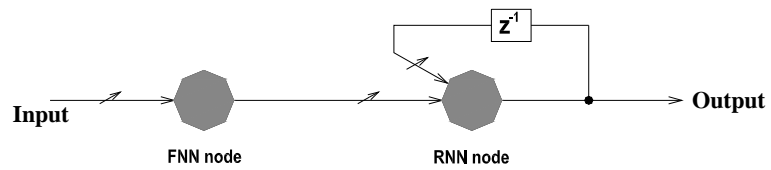


Figure 5.7b: Proposed FRCS structure

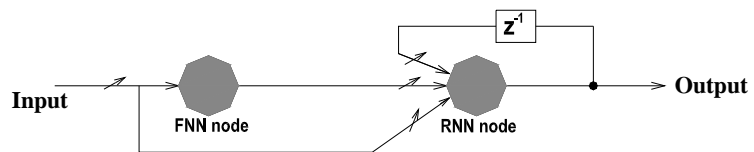


Figure 5.7c: Proposed HKFRCS structure

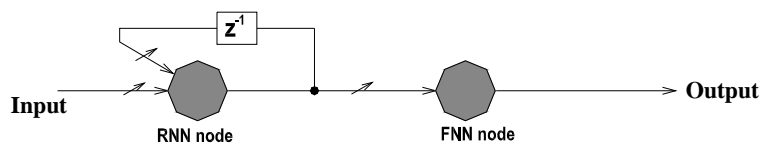


Figure 5.7d: Proposed RFCS structure

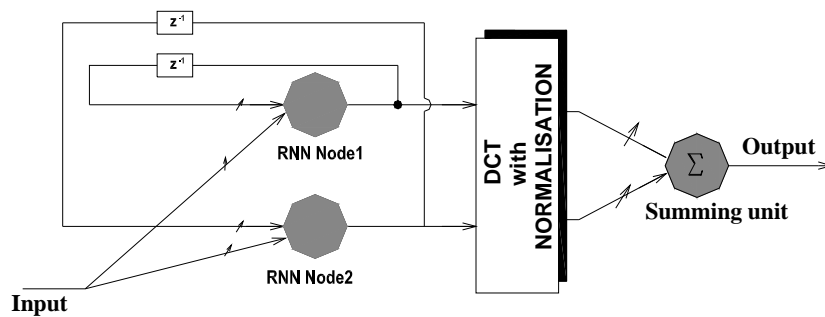


Figure 5.7e: Proposed RTCS structure

SL. No.	Equaliser structures	No. of Neurons	No. of Summing units	No. of Adaptable weights	No. of Fixed weight	Other Adaptable parameters	Figure No.
1.	CRNN	2	-	6	-	-	5.7a
2.	FRCS	2	-	3	-	-	5.7b
3.	HKFRCS	2	-	4	-	-	5.7c
4.	RFCS	2	-	3	-	-	5.7d
5.	RTCS	2	1	8	4	-	5.7e
6.	FZTUNRNN	2	-	6	-	Slope( $\bullet$ )	5.7a

**Table 5.1: Structural configuration comparison of the proposed RNN based equalisers with the conventional RNN, Feedforward order of all equalisers ‘ $m$ ’= 1.**

For a comparative study and analysis purpose the number of training samples presented to all the proposed equalisers considered here are restricted to 200 samples only as it is observed that their performances are quite satisfactory. The BER performance comparison of the proposed equaliser structures based on RNN topology has been carried out after all the structures have undergone a training phase (200 samples) in accordance with the training algorithms already discussed in Sections 5.1.3, 5.2.1, 5.3.1, 5.4.1 and 5.5.1 of this chapter. The weight vectors of the equalisers are frozen after the training stage and then the test is continued. The BER performances for each SNR are evaluated, based on  $10^7$  more received symbols (test samples) and averaged over 20 independent realisations. The parameters used for adaptation of all proposed equalisers for various linear and nonlinear channels under study are provided in Table 5.2.

CHANNEL PARAMETERS		CRNN	FRCS	HKFRCS	RFCS	RTCS	FZTUNRNN
$H_1(z)$	$\lambda$	0.5	4	6	0.5	4	0.8
	$\eta$	1	0.01	0.01	0.5	0.01	1
$H_2(z)$	$\lambda$	0.5	8	3	0.5	8	0.5
	$\eta$	1	0.01	0.01	0.5	0.01	1
$H_3(z)$	$\lambda$	0.5	6	4	1	4	0.5
	$\eta$	1	0.01	0.01	1	0.01	1
$H_5(z)$	$\lambda$	0.5	4	8	0.5	4	0.3
	$\eta$	1	0.01	0.01	0.5	0.01	1
$H_6(z)$	$\lambda$	0.5	6	6	0.2	4	0.8
	$\eta$	1	0.01	0.01	0.2	0.01	1
$H_7(z)$	$\lambda$	0.5	4	4	0.5	4	2
	$\eta$	1	0.01	0.01	0.5	0.01	1
$H_8(z)$	$\lambda$	0.5	0.1	3	0.5	3	0.8
	$\eta$	1	0.01	0.01	0.5	0.01	1
$H_9(z)$	$\lambda$	0.5	4	4	1	4	0.3
	$\eta$	1	0.01	0.01	0.2	0.01	1
$H_{11}(z)$	$\lambda$	0.5	4	8	0.5	4	0.8
	$\eta$	1	0.01	0.01	0.5	0.01	1
$H_{13}(z)$	$\lambda$	0.5	4	4	0.5	4	1
	$\eta$	1	0.01	0.01	0.5	0.01	1
$H_{14}(z)$	$\lambda$	2	4	4	0.3	4	0.8
	$\eta$	1	0.01	0.01	0.5	0.01	1

Table 5.2: Learning Parameters of Proposed RNN Based Equaliser Structures

Example 1: Channel  $H_1(z) = 1 + 0.5z^{-1}$

Figure 5.8 shows the the BER performance plots using the proposed RNN based equalisers for a channel characterised by  $H_1(z)$ . For this minimum phase channel, the proposed FRCS, HKFRCS and FZTUNRNN equalisers show a SNR gain of almost 0.7 db at a prefixed BER of  $10^{-4}$  over the conventional RNN one.

Example 2: Channel  $H_2(z) = 1 + 0.7z^{-1}$

Figure 5.9 depicts the simulation results of equalisation of a channel model  $H_2(z)$  using all the proposed structures based on RNN topology. The proposed FRCS, HKFRCS and RTCS equalisers show better BER performances over the conventional one by providing 1.3-1.7dB gain in SNR level at a prefixed BER of  $10^{-4}$ .

Example 3: Channel  $H_3(z) = 1 + 0.95z^{-1}$

For this typical channel (zero close to the unit circle), it is observed in Figure 5.10 that the conventional RNN equaliser provides a poor performance (it yields a BER of  $10^{-2}$  at 18 dB SNR condition), where as all the proposed RNN based structures exhibit significant improvement in BER performance (a BER level of  $10^{-4}$  can be achieved at almost 15-17dB SNR condition).

Example 4: Channel  $H_5(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$

For this non-minimum phase channel model  $H_5(z)$ , Figure 5.11 demonstrates that all the proposed equalisers have an edge over the conventional RNN structure performance wise at realistic SNR levels( >16 dB) though not in high noise conditions by providing about 1.5-1.8 dB SNR gain at a prefixed BER level of  $10^{-5}$ .

Example 5: Channel  $H_6(z) = 0.4084 + 0.8164z^{-1} + 0.4084z^{-2}$

Figure 5.12 demonstrates the enhancement in BER performances offered by proposed structures when applied to equalise a three tap channel with a coincident state, defined by  $H_6(z)$ . An impressive gain of almost 3.4 dB at a BER level of  $10^{-4}$  is noticed if the proposed RFCS structure is employed. Application of the HKFRCS, RTCS and FZTUNFNN equalisers demonstrate SNR gains of more than 1 dB at a prefixed BER level of  $10^{-4}$  over the conventional RNN one.

Example 6: Channel  $H_7(z) = 1 - 2z^{-1} + z^{-2}$

Further, equalisation of a partial response channel characterized by  $H_7(z)$  has been attempted employing all the proposed structures in the RNN framework and the corresponding BER plots shown in Figure 5.13 prove the supremacy of the proposed equaliser structures in terms of improvement in SNR level.

Example 7: Channel  $H_8(z) = 0.407 - 0.815z^{-1} - 0.407z^{-2}$

Further, all the proposed structures reported in this chapter are compared with the CRNN structure for equalisation of a channel  $H_8(z)$  in Figure 5.14. Though the proposed RFCS and FRCS equalisers exhibit almost the same performance as that of a conventional RNN one, the HKFRCS, RTCS and FZTUNRNN equalisers show distinct SNR gains of about 4-5 dB at a prefixed BER level of  $10^{-4}$  which is quite encouraging.

Example 8: Channel  $H_9(z) = 0.7255 + 0.584z^{-1} + 0.3627z^{-2} + 0.0724z^{-3}$

Figure 5.15 presents the BER performance comparison of all the proposed RNN based equaliser structures with a conventional RNN one for a four tap channel described by  $H_9(z)$ . For high noise conditions the performances of all the proposed structures are similar to the conventional RNN one. At a 16dB SNR condition, the proposed RTCS, HKFRCS and FRCS equalisers can reach a BER of  $10^{-5}$  in comparison with a BER of  $10^{-3}$  obtained using the conventional RNN one.

Example 9: Channel  $H_{11}(z) = -0.2052 - 0.5131z^{-1} + 0.7183z^{-2} + 0.3695z^{-3} + 0.2052z^{-4}$

Figure 5.16 shows a performance comparison of all the proposed equalisers based on RNN topology considering a channel defined by  $H_{11}(z)$ . All the proposed equalisers performances are similar to that of the CRNN one except the RTCS equaliser structure. It offers better performance by attaining a prefixed BER of  $10^{-4.5}$  at 18 dB SNR condition.

Example 10: Channel  $H_{14}(z) = (1 + 0.5z^{-1}) - 0.9(1 + 0.5z^{-1})^3$

Lastly, Figure 5.17 show performance curves for the equalisation performed on a nonlinear channel,  $H_{14}(z)$ . For this example the proposed FRCS, HKFRCS, FZTUNRNN and RTCS equalisers result significant 1.7dB gain in SNR level at a prefixed BER of  $10^{-5}$  over the CRNN equaliser which clearly justifies their application for such type of channel.

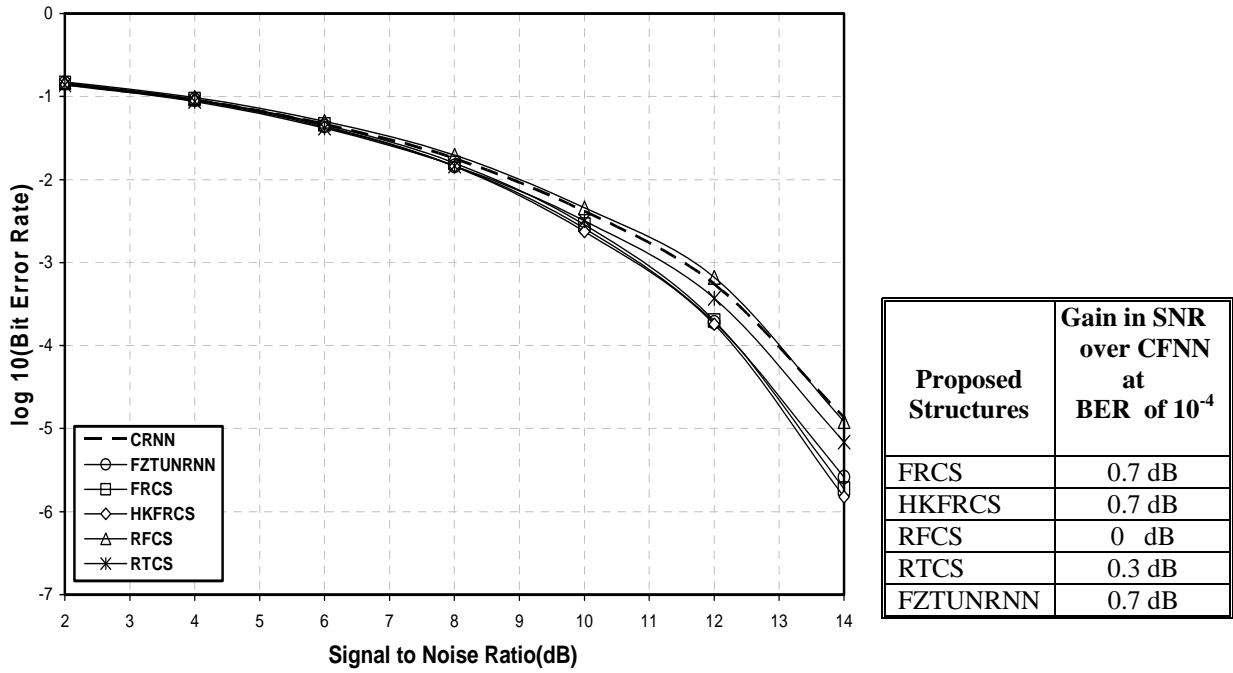


Figure 5.8 : BER performance comparison of proposed RNN based equalisers with conventional RNN for Channel  $H_1(z)$

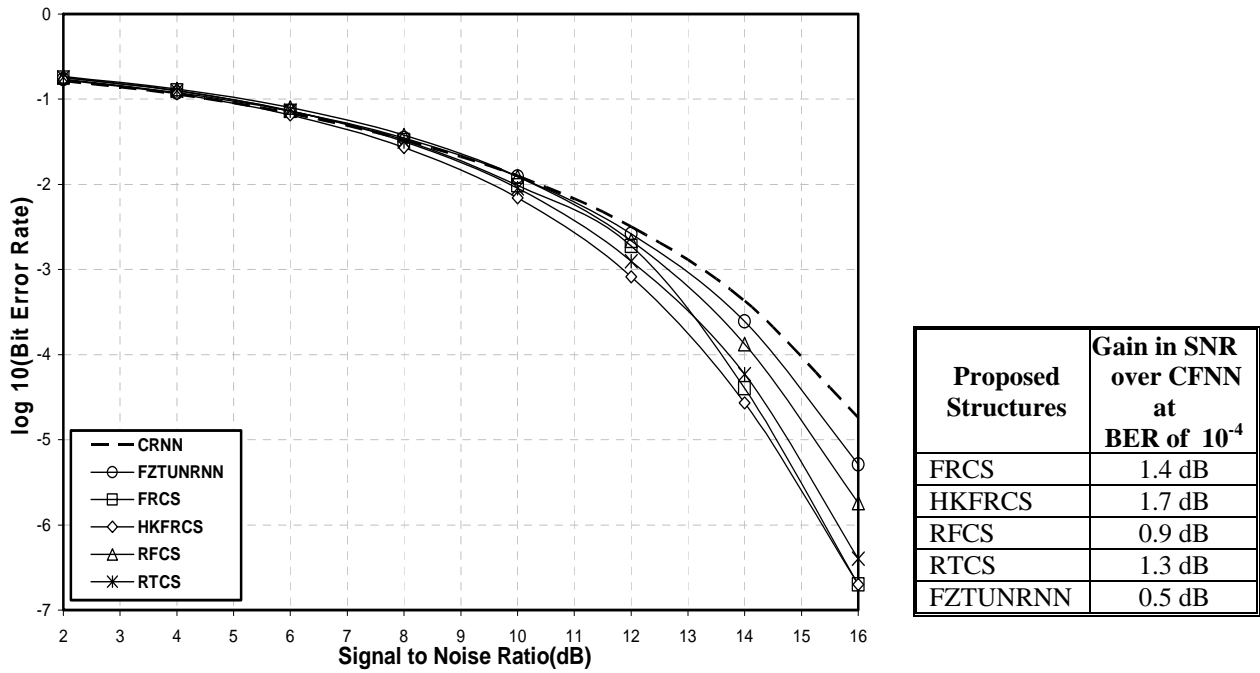


Figure 5.9 : BER performance comparison of proposed RNN based equalisers with conventional RNN for Channel  $H_2(z)$



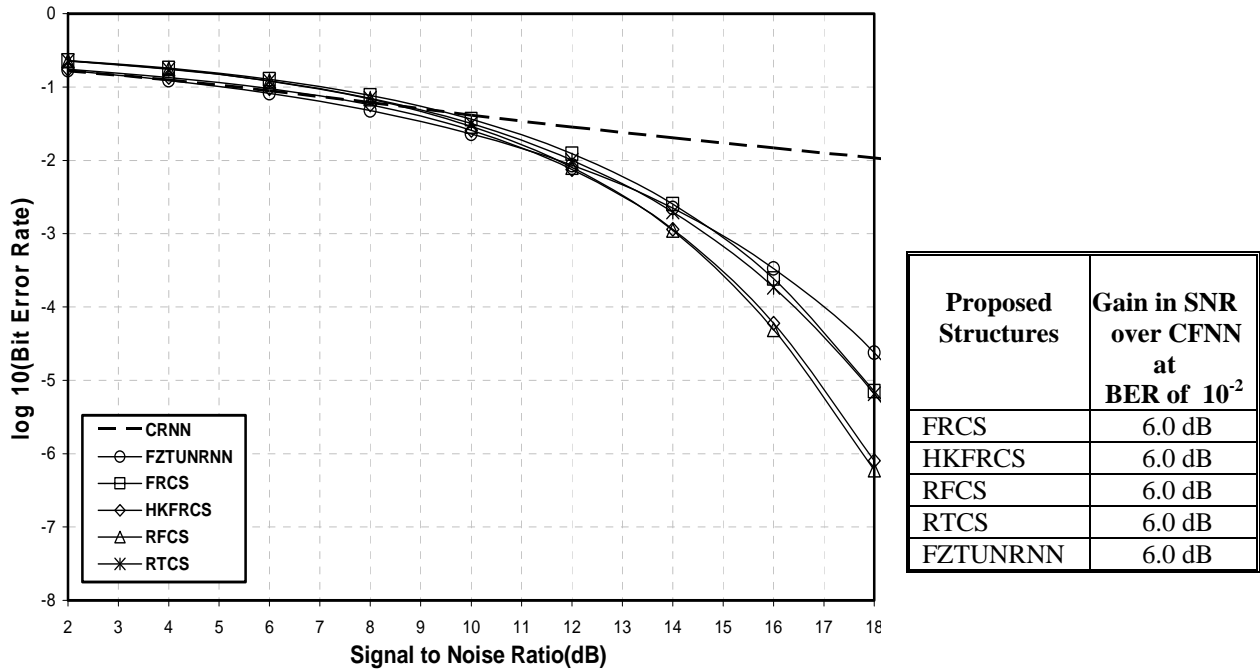


Figure 5.10 : BER performance comparison of proposed RNN based equalisers with conventional RNN for Channel  $H_3(z)$

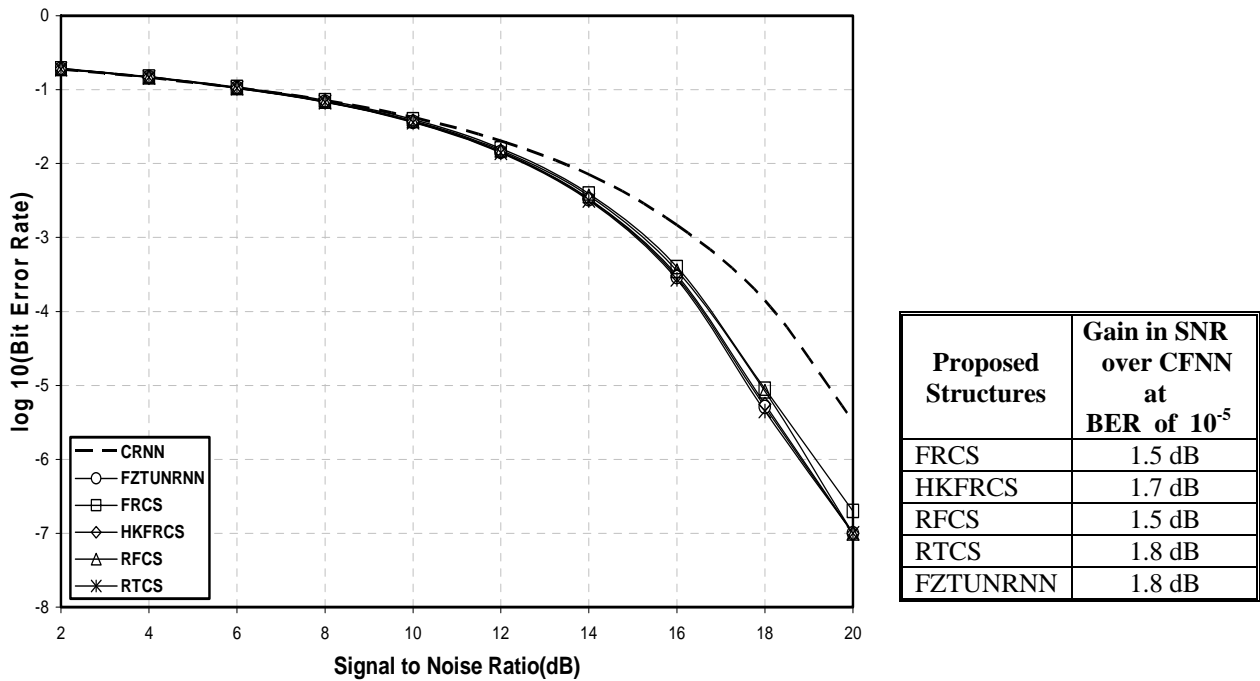


Figure 5.11: BER performance comparison of proposed RNN based equalisers with conventional RNN for Channel  $H_5(z)$

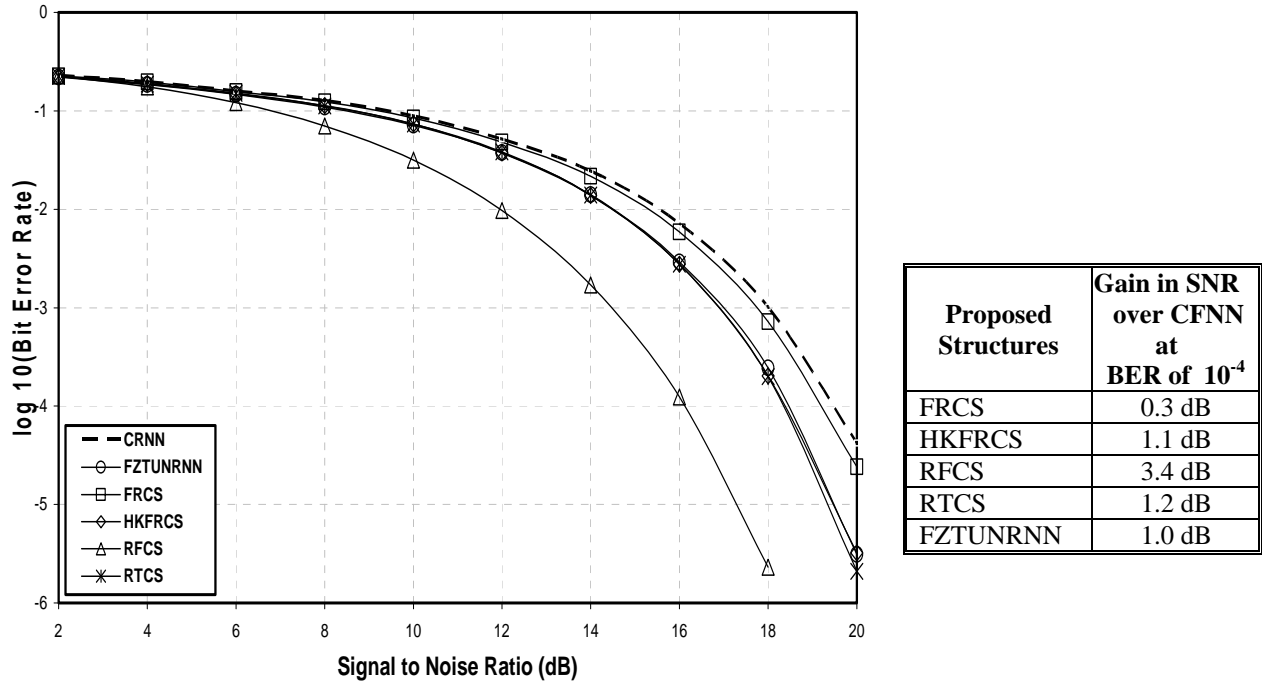


Figure 5.12: BER performance comparison of proposed RNN based equalisers with conventional RNN for Channel  $H_6(z)$

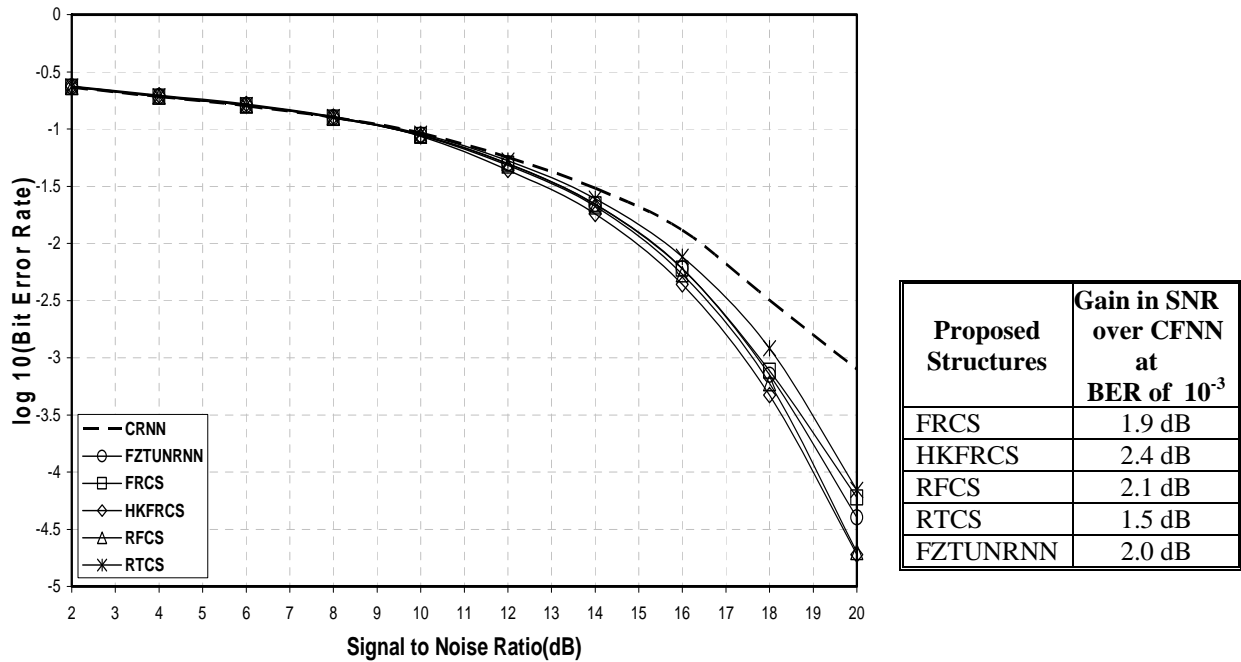


Figure 5.13: BER performance comparison of proposed RNN based equalisers with conventional RNN for Channel  $H_7(z)$

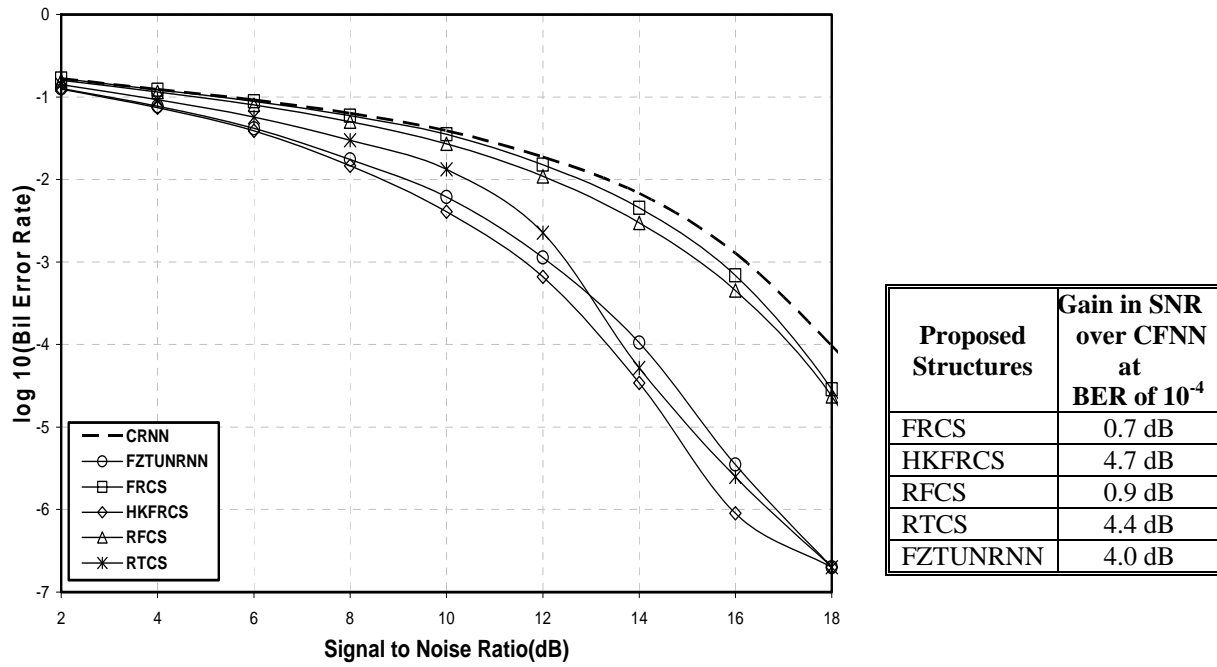


Figure 5.14: BER performance comparison of proposed RNN based equalisers with conventional RNN for Channel  $H_8(z)$

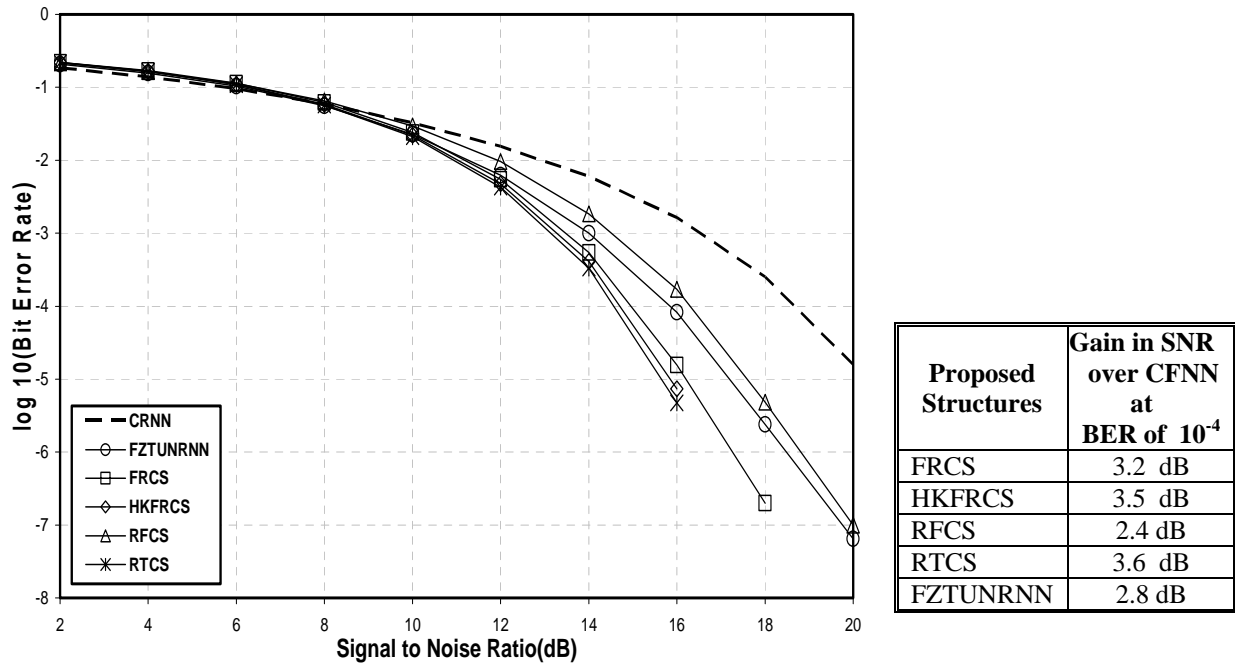


Figure 5.15: BER performance comparison of proposed RNN based equalisers with conventional RNN for Channel  $H_9(z)$

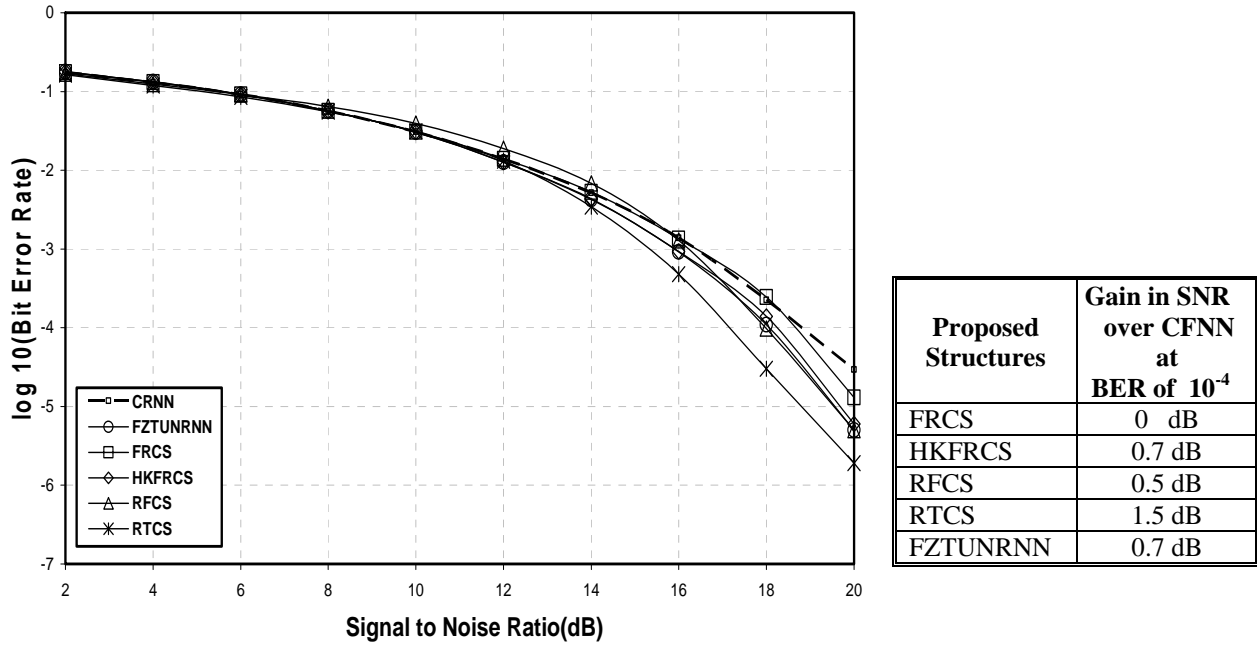


Figure 5.16: BER performance comparison of proposed RNN based equalisers with conventional RNN for Channel  $H_{11}(z)$

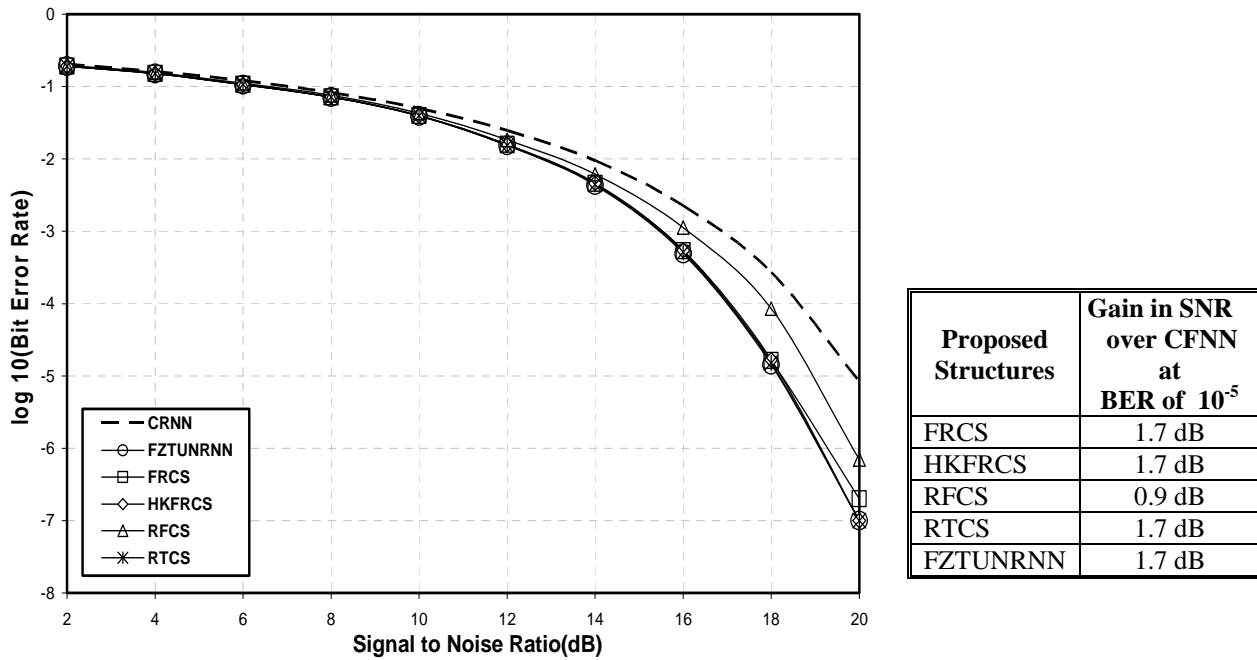
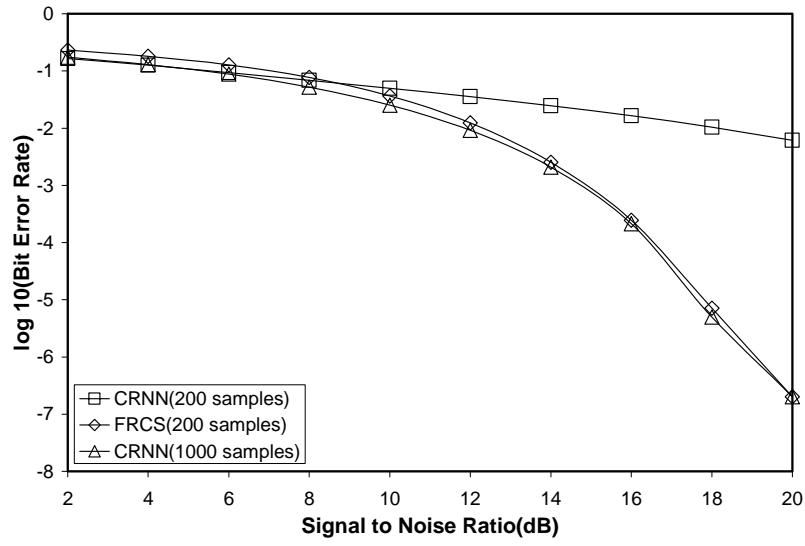


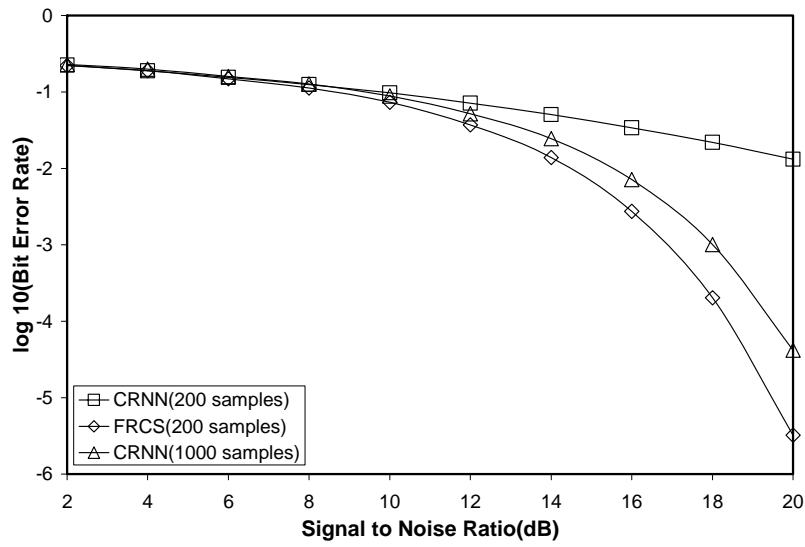
Figure 5.17: BER performance comparison of proposed RNN based equalisers with conventional RNN for Channel  $H_{14}(z)$

All the proposed equalisers in RNN domain require fewer samples in training phase for satisfactory BER performance. Simulation results demonstrate this advantages offered by these structures. Figure 5.18 shows the effect of change of length of training sequence on the BER performance obtained using the conventional RNN equaliser. It is observed that in channel  $H_3(z)$ , the CRNN equaliser (trained with 1000 samples) is able to attain the BER performance level of the proposed FRCS equaliser (trained using 200 samples only), where as for channel  $H_6(z)$  its performance is still inferior. In Figure 5.19, the BER performance of proposed HKFRCS equaliser (exposed to 200 training samples) is compared with conventional one. It is shown for channels  $H_3(z)$  and  $H_8(z)$  that increasing the length of learning phase from 200 to 1000 samples, the CRNN equaliser still could not achieve the BER performance level of the HKFRCS equaliser. Further, the proposed RFCS equaliser (trained with 200 samples) BER performance is compared in Figure 5.20 with that of a conventional RNN equaliser (increasing the training samples to 1000). Though the CRNN equaliser is able to provide the same performance level as that of RFCS structure for channel  $H_7(z)$ , no significant improvement in case of channel  $H_6(z)$  is observed. The proposed RTCS equaliser (trained using 200 samples) shows better performance even if the CRNN based equaliser is presented with 1000 training samples for channel  $H_8(z)$  and  $H_9(z)$  as shown in Figure 5.21. It is noticed in Figure 5.22 that the conventional RNN based equaliser can achieve the BER performance obtained using the proposed FZTUNRNN equaliser (trained using 200 samples) by increasing the length of the learning phase sequence to 1000 samples for examples of channels  $H_3(z)$  and  $H_9(z)$ .

Thus it is concluded from the exhaustive simulation study that BER performances of the proposed equalisers in RNN domain are superior and all these structures learn much faster in comparison with the conventional RNN one. Several types of channel models, used as examples, also demonstrate the robustness of the proposed equaliser structures and verify the efficacy of the new techniques applied.

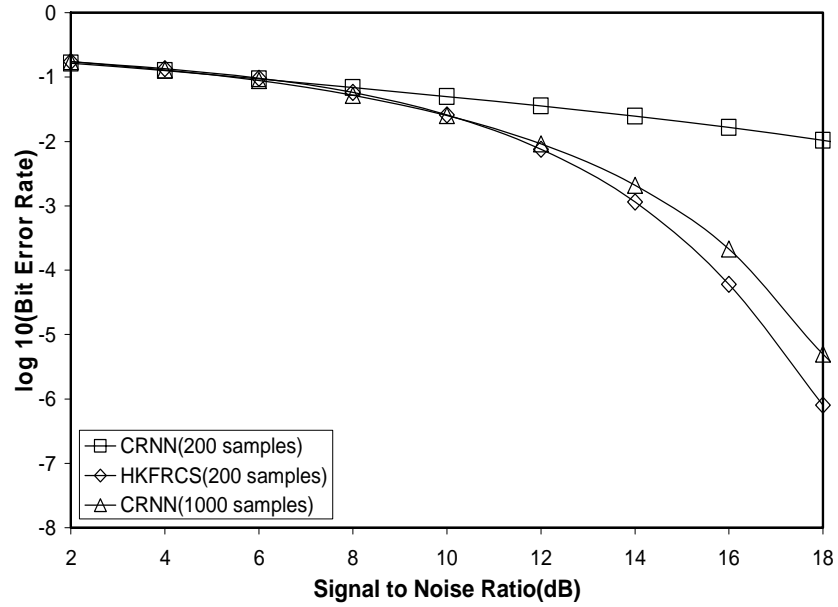


(a)

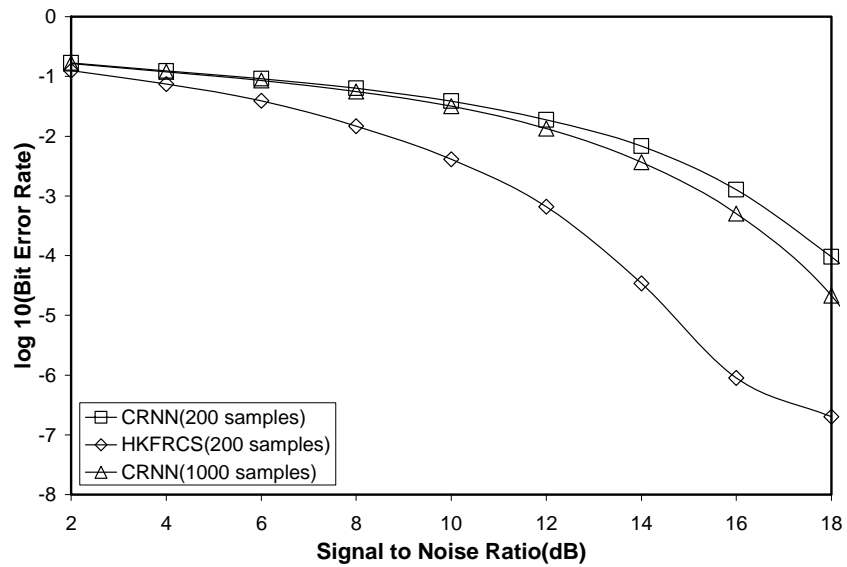


(b)

Figure 5.18 : BER performance comparison of FRCS equaliser with CRNN w.r.t. training samples for Channels (a)  $H_3(z)$  and (b)  $H_6(z)$

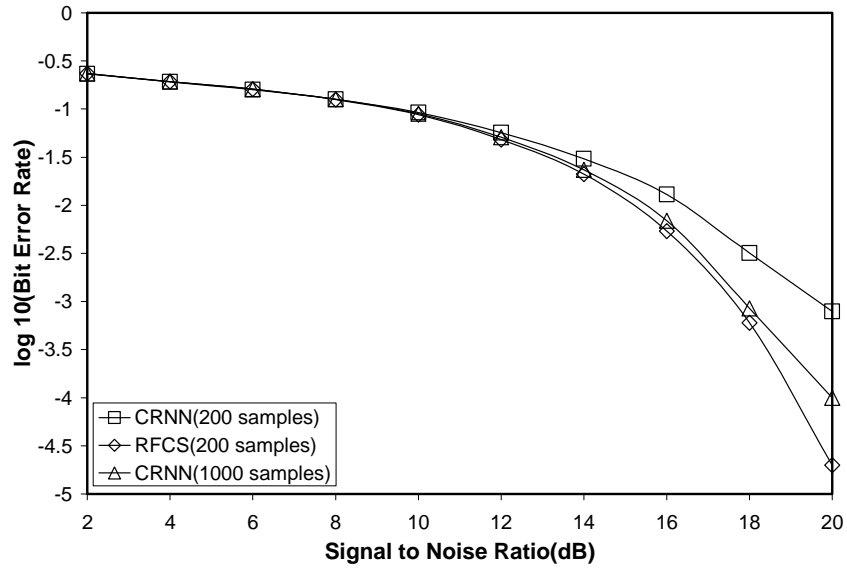


(a)

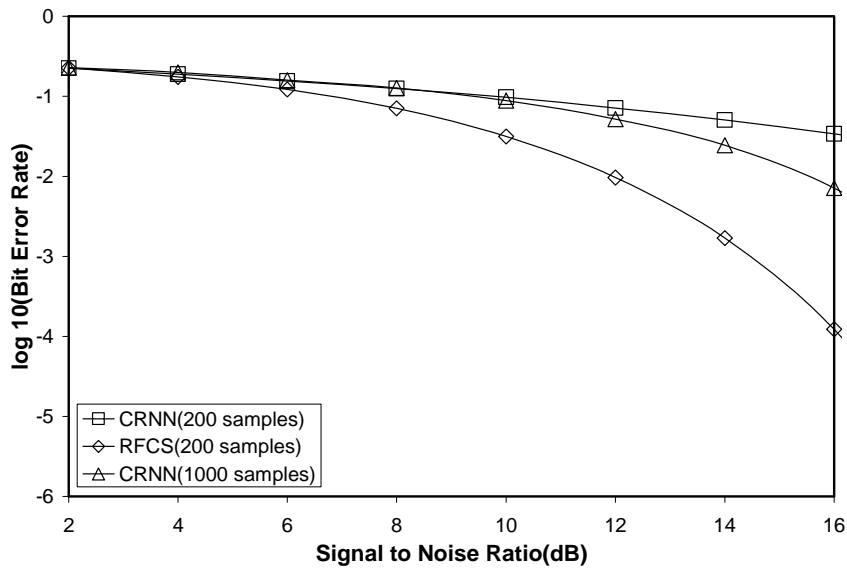


(b)

**Figure 5.19 : BER performance comparison of HKFRCS equaliser with CRNN w.r.t. training samples for Channels(a) $H_3(z)$  and (b) $H_8(z)$**



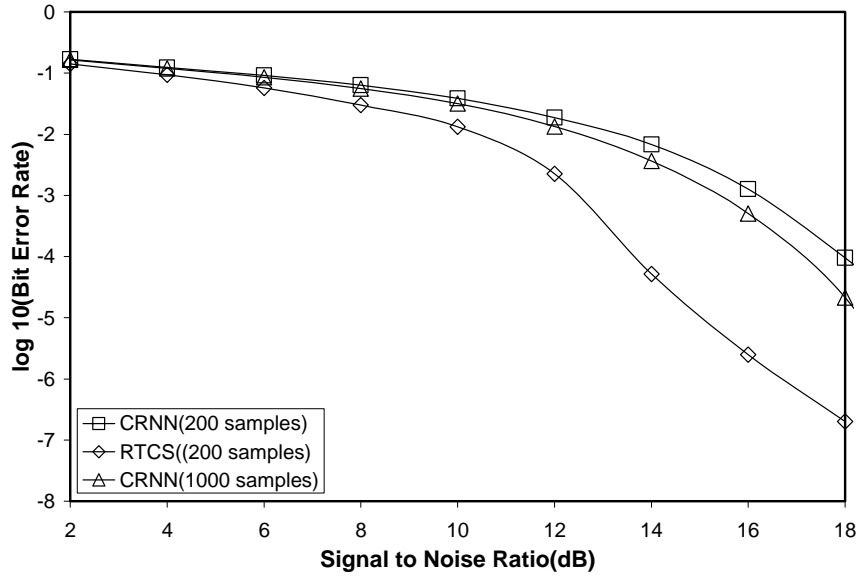
(a)



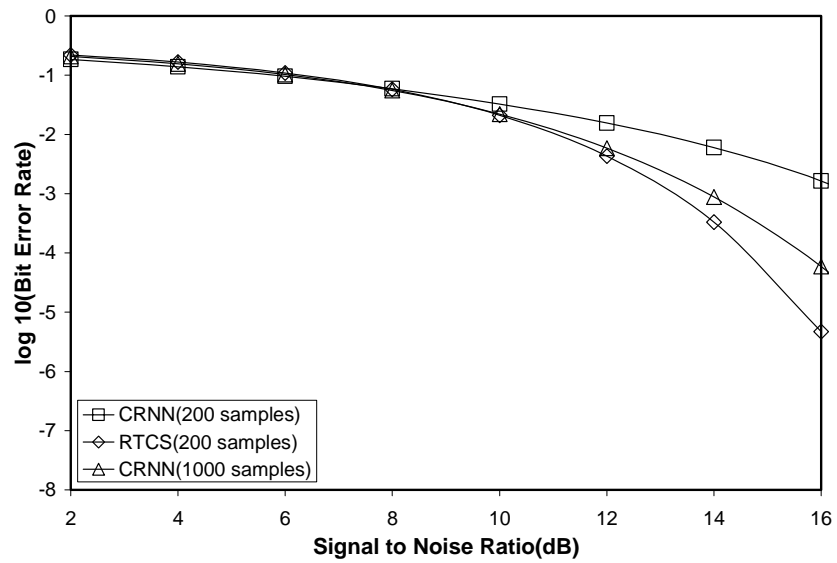
(b)

Figure 5.20: BER performance comparison of RFCS equaliser with CRNN w.r.t. training samples for Channels(a)  $H_7(z)$  and (b)  $H_6(z)$



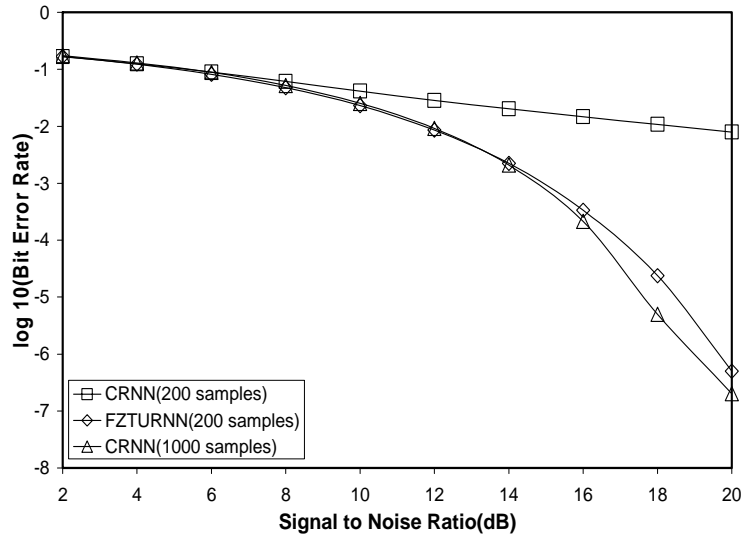


(a)

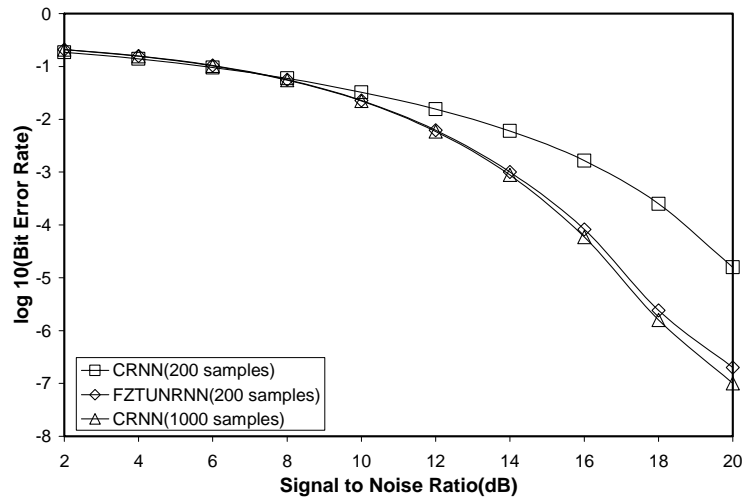


(b)

Figure 5.21 : BER performance comparison of RTCS equaliser with CRNN w.r.t. training samples for Channels (a) $H_8(z)$  and (b) $H_9(z)$



(a)



(b)

Figure 5.22 : BER performance comparison of FZTURNN equaliser with CRNN w.r.t. training samples for Channels (a)  $H_3(z)$  and (b)  $H_9(z)$

## 5.7 Conclusion

In this chapter hybrid configurations using cascaded modules of RNN and FNN have been proposed. Further the FNN module is replaced with a transform block in the hybrid structure of RNN-FNN cascaded equaliser and a new configuration has been proposed. As the sole aim of this research work is to develop reduced network configurations for ease of real time implementation in all practical applications, the overall structural complexities have been kept within limits by restricting the selection of number of nodes in both RNN and FNN modules while the cascading technique is employed. Different training algorithms have been developed for the weight adaptation in the proposed equalisers looking into their hybrid structural configurations. The equivalence approach based weight updation developed in this present work is a novel concept for such hybrid structures as discussed in Section 5.1.2 and helps to employ the existing RTRL and BP algorithms. Also the technique followed in the training algorithm given in Section 5.4.1 to back propagate the output error through the transform block, proves to be highly efficient from the performance point of view. Further, tuning of the sigmoid slope of the neurons in conventional RNN equaliser using fuzzy logic concept explained in Section 5.5.1 helps to improve performance noticeably by increasing the network adaptability, although there is no visible structural modifications. It is observed from the exhaustive simulation study that while all the equalisers proposed have resulted faster learning and encouraging BER performances, the gains obtained are entirely channel dependent. For example, for channel  $H_3(z)$ , all the proposed RNN based cascaded equalisers perform better than a conventional RNN where as for  $H_6(z)$ , only the proposed RFCS equaliser yields improved BER performance. For channel  $H_8(z)$ , the HKFRCS and RTCS and FZTUNRNN equalisers perform better, but FRCS and RFCS equalisers achieve the same BER performance in comparison to conventional RNN one.

---

## CHAPTER 6

### Conclusion

---

The work described in the thesis is primarily concerned with the development of novel adaptive equalisers for communication channels using ANN techniques. In particular, the main focus of this research work is to design novel neural equalisers of reduced structural complexity based on both conventional FNN and RNN topologies. Further, certain modifications in the existing Back-Propagation and Real-Time-Recurrent-Learning algorithms have been incorporated to update the connection weights of the proposed structures. The training algorithms, developed to adapt the Orthogonal basis function based FNN equaliser structure and Transform domain based FNN equaliser structure, are important contributions of this thesis work. Especially the concept of the proposed equivalence approach employed in the RNN based cascaded structures for weight adaptation adds a new dimension to research. This thesis work also has contributed to the proper selection of the key parameters associated with the equaliser structures. Along with the design of efficient structures, a different approach has been suggested to choose the key parameters involved with the equalisers for optimising the BER performance. However, in this study the major thrust is given in comparing the performances of the proposed equalisers with their conventional counterparts (either FNN or RNN based equaliser), considering the examples of various linear and nonlinear communication channel models.

In Section 6.1 a summary of the undertaken research is presented, followed by highlighting the specific achievements accomplished in this work. Section 6.2 discusses the limitations of the present work and new direction for future work is proposed in Section 6.3.

#### 6.1 Summary and achievement of the thesis

The work presented in this thesis can be divided into two distinct parts. In the first part, the factors influencing the performance of optimal symbol-by-symbol equalisers are discussed and the importance of proper selection of the key design parameters (feedforward order ' $m$ ', feedback order ' $n_b$ ' and decision delay ' $d$ ') is analysed, which has led to the formulation of certain empirical relations. Secondly,

efficient equaliser structures in FNN and RNN domains with reduced structural complexity are proposed. Particularly, some contributions have also been made in the development of training algorithms for faster convergence. Even though the BP algorithm remains as the backbone of the proposed FNN based equaliser structures, certain modifications have been incorporated for evolving new training algorithms to adapt the proposed structure. Specific emphasis has been provided to suitably alter the conventional BP algorithm for adapting the connection weights of the proposed OBFNN and TDFNN based equalisers due to the positioning of the OBF block and transform block in the conventional structures respectively.

Chapter-3 of the thesis provides a detailed study on various factors influencing the performance of an optimal symbol by symbol (Bayesian) equaliser. In case of an ideal channel (no ISI), even though the optimal decision boundary is linear, the BER performance degrades at high additive noise level as evident from Figure 3.1e. Further, the influence of ISI makes the optimal decision boundary nonlinear as can be seen in Figure 3.3c for channel  $H_1(z)$  and the severity of additive noise level also plays a major role in misclassification of received symbols. Hence, it is inferred from the simulation study that both ISI and additive noise level in a channel influence the BER performance of the equalisers appreciably only in realistic SNR range, as illustrated in Figure 3.4. The number of noise-free channel states close to the optimal decision boundary varies with decision delay significantly and if it is more, then the probability of misclassification of observed samples increases when noise is present in the communication system. It is observed that increasing feedforward order of the equaliser improves the performance, but certain drawbacks are encountered as the structural complexity increases. However, the feedforward order has been restricted to the length of channel impulse response, maintaining a trade-off between performance and complexity, as discussed in Section 3.3. There has been an interesting observation that in a channel having coincident channel states, even if the feedforward order is increased to a large value, BER performance gain obtained is not satisfactory at all as seen in Figure 3.8d for a channel defined by  $H_6(z)$ . This necessitates the inclusion of decision feedback technique, which eliminates the existence of coincident states while reducing the number of channel states. Improvement of BER performance in the DFE structures is observed only if all the parameters like feedforward order ' $m$ ', feedback order ' $n_b$ ' and decision delay ' $d$ ' are properly chosen, otherwise equalisers without decision feedback and with a proper decision delay yield better BER performance as shown in Figure 3.16a-e. An exhaustive analysis regarding the selection of the key design parameters has given an insight for developing a new approach. For equalisers without decision feed back, empirical relations and logical explanations have been presented for optimal selection of the delay order in Section 3.2.1. For equalisers with decision feedback, Section

3.2.2 provides a procedure and its interpretation for the proper selection of the key design parameters. Separate studies and BER plots for examples of various channel models (both symmetric and asymmetric types) prove the accuracy of the derived mathematical relationships. This approach opens up a new direction for selecting the parameters directly from the channel tap coefficients.

Chapter-4 is entirely devoted to one of the major components of the contributions of the work undertaken relating to the design of novel equaliser structures in the FNN domain along with the development of suitable training algorithms to adapt the network weights during the learning phase. The extensive simulation study shows that the proposed equaliser structures are superior in performance and require less number of training samples for satisfactory BER performance in comparison with the conventional FNN equalisers. The justifications behind the performance enhancement using the proposed equalisers have been emphasised below.

- Section 4.1 discusses about the hierarchical knowledge reinforced FNN (HKFNN) equalisers. The improvement in results for such equalisers can be attributed to the fact that the information is fed from one layer to the next layer for consolidation of the knowledge base and hence this equaliser configuration yields improved performance. The proposed structure has been designed with only one node per layer restricting the structural configuration. The opinions of the experts (node outputs) are passed on hierarchically to the subsequent experts so that the knowledge base gets more refined at the output node due to sequential processing operation in each node layer by layer. Thus in HKFNN structure, the output node is fed with more information (the original information from the input layer and the expert opinions of all the preceding nodes) in comparison to the Conventional FNN. The performance improvement of the proposed HKFNN structure in comparison to a conventional one can be seen in the Table 6.1. Further, it is observed from Figure 4.23 that even exposing the CFNN structure to a sequence of 2000 training samples instead of 1000 samples, the performance improvement is not significant w.r.t. HKFNN for channels  $H_7(z)$  and  $H_{10}(z)$ .

Sl.No.	Channels	Gain in SNR over conventional FNN	Figure No.
1.	$H_1(z)$	1.8 dB at BER of $10^{-4}$	4.9
2.	$H_6(z)$	1.3 dB at BER of $10^{-5}$	4.14
3.	$H_7(z)$	1.4 dB at BER of $10^{-3}$	4.15
4.	$H_8(z)$	1.8 dB at BER of $10^{-4}$	4.16
5.	$H_{10}(z)$	2.7 dB at BER of $10^{-2.5}$	4.18
6.	$H_{11}(z)$	1.4 dB at BER of $10^{-4}$	4.19
7.	$H_{12}(z)$	1.7 dB at BER of $10^{-4}$	4.20
8.	$H_{14}(z)$	1.1 dB at BER of $10^{-5}$	4.21
9.	$H_{15}(z)$	1.0 dB at BER of $10^{-5}$	4.22

**Table 6.1 : Performance analysis of proposed HKFNN equaliser**

- Another FNN based equaliser termed as the orthogonal basis function (OBFNN) equaliser based on self-breeding genetic framework, is presented in Section 4.2. Such an equaliser has resulted in encouraging BER performances in comparison to Conventional FNN. The gain in performance is basically due to the evolution concept, where the decision at a node (expert's opinion) instead of being directly conveyed to the next node undergoes a two dimensional orthogonal expansion. While one output preserves the information of the current generation to take part in the final decision, the other one is allowed to pass on the information to the next generation to generate a new expert opinion. Such configuration generally reinforces the information base of the final decision node to yield a better estimate as compared with a conventional FNN structure. The performance gain of this equaliser excited by 1000 training samples is shown in Table 6.2. And it is also observed from Figure 4.24 in channels  $H_3(z)$  and  $H_8(z)$  that increasing the length of learning phase to 2000 samples, though a CFNN equaliser has resulted an improved BER performance but the proposed OBFNN equaliser trained with 1000 samples only is still superior.

Sl.No.	Channels	Gain in SNR over conventional FNN	Figure No.
1.	$H_1(z)$	1.6 dB at BER of $10^{-4}$	4.9
2.	$H_2(z)$	1.3 dB at BER of $10^{-5}$	4.10
3.	$H_3(z)$	1.0 dB at BER of $10^{-4}$	4.11
4.	$H_8(z)$	2.4 dB at BER of $10^{-4}$	4.16
5.	$H_{10}(z)$	1.5 dB at BER of $10^{-2.5}$	4.18
6.	$H_{11}(z)$	1.0 dB at BER of $10^{-4}$	4.19
7.	$H_{15}(z)$	1.2 dB at BER of $10^{-5}$	4.22

**Table 6.2 : Performance analysis of proposed OBFNN equaliser**

- The TDFNN equaliser employing a DCT block with power normalisation in cascade with a FNN module comprising of a single layer is another example of a new variant of equaliser based on FNN topology and is described in Section 4.3. So far as the choice of real-valued transform is considered, DCT is a clear winner over its competitors like DHT, DST, etc. based on performance study. The gain achieved with such hybrid configuration in comparison with a simple two layer CFNN is primarily due to the fact that the transform block at the output end performs further de-correlation of the already preprocessed information from the FNN module. Table 6.3 provides the performance gain attained by this equaliser for various channels. Further, it is observed for channels  $H_7(z)$  and  $H_{10}(z)$  shown in Figure 4.25 that if the CFNN equaliser is presented with 2000 training samples instead of 1000 samples, then its BER performance improved but the result remained inferior in comparison to that of the proposed TDFNN equaliser.

Sl.No.	Channels	Gain in SNR over conventional FNN	Figure No.
1.	$H_1(z)$	1.0 dB at BER of $10^{-4}$	4.9
2.	$H_3(z)$	1.0 dB at BER of $10^{-4}$	4.11
3.	$H_6(z)$	1.1 dB at BER of $10^{-5}$	4.14
4.	$H_7(z)$	2.5 dB at BER of $10^{-3}$	4.15
5.	$H_8(z)$	1.6 dB at BER of $10^{-4}$	4.16
6.	$H_{10}(z)$	2.9 dB at BER of $10^{-2.5}$	4.18
7.	$H_{11}(z)$	1.2 dB at BER of $10^{-4}$	4.19
8.	$H_{12}(z)$	2.7 dB at BER of $10^{-4}$	4.20
9.	$H_{14}(z)$	1.2 dB at BER of $10^{-5}$	4.21
10.	$H_{15}(z)$	1.1 dB at BER of $10^{-5}$	4.22

**Table 6.3 : Performance analysis of proposed TDFNN equaliser**

- Section 4.4 presents Fuzzy tuned FNN (FZTUNFNN) equaliser designed on an FNN platform which is a conventional FNN structure with a reduced structure, where fuzzy logic concept is employed to tune the slope ( $\phi$ ) of the sigmoid activation functions at all the nodes. The adaptation of the slope parameter increases the degrees of freedom in the weight space of the conventional FNN configuration and thus provides a better non-linear mapping between the input and output. Application of this technique makes the existing CFNN structure more adaptable and hence significant performance gain can be expected from such situations even though the proposed structure has not undergone any structural modifications. The performance enhancement of the proposed structure is presented in Table 6.4. However, it is inferred from Figure 4.26 for channels  $H_1(z)$  and  $H_8(z)$  that if 2000 training samples are provided to a CFNN structure instead of 1000 samples, its BER performance certainly improves, but still the proposed FZTUNFNN equaliser (trained with 1000 samples only) shows better BER performance.

Sl.No.	Channels	Gain in SNR over conventional FNN	Figure No.
1.	$H_1(z)$	1.8 dB at BER of $10^{-4}$	4.9
2.	$H_4(z)$	1.0 dB at BER of $10^{-4}$	4.12
3.	$H_8(z)$	1.7 dB at BER of $10^{-4}$	4.16
4.	$H_{10}(z)$	2.2 dB at BER of $10^{-2.5}$	4.18
5.	$H_{11}(z)$	1.0 dB at BER of $10^{-4}$	4.19
6.	$H_{12}(z)$	1.5 dB at BER of $10^{-4}$	4.20
7.	$H_{15}(z)$	1.1 dB at BER of $10^{-5}$	4.22

**Table 6.4 : Performance analysis of proposed FZTUNFNN equaliser**



The sole objective of the proposed research work is focussed on the development of reduced structure based efficient neural equalisers, so the Recurrent Neural Network (RNN) platform has emerged as an attractive alternative in this research work. Although in RNN framework many hybrid configurations have been chosen employing cascading technique, it is ensured that under no circumstances the main objective of developing equalisers on reduced structure framework be defeated. Analysis of such configurations in RNN platform and their training algorithms has been presented in Chapter-5.

- The first structure proposed in Section 5.1 is FNN-RNN cascaded (FRCS) equaliser, which consists of two modules, a FNN module is followed by an RNN one. Both the FNN and RNN modules have one processing unit each, in contrast to the conventional RNN having two processing units constraining the structural complexity. A novel ‘*equivalence approach*’ has been applied in RNN framework to evaluate its node errors, which cannot be explicitly defined if alone the RTRL algorithm is used. The development of new training algorithms of the cascaded structures is mainly based on this technique. The enhancement in BER performance using the proposed equalisers over the existing conventional RNN equaliser is given in Table 6.5. The improvement in result of the proposed structures in comparison to a conventional RNN based one, is due to the self pseudo-decision feedback strategy which is an inbuilt phenomenon within an RNN framework. Here, the signal is pre-processed in the FNN block before being fed to the RNN module cascaded to it and hence gain in the BER performance is achieved.

Sl.No.	Channels	Gain in SNR over conventional RNN	Figure No.
1.	$H_2(z)$	1.4 dB at BER of $10^{-4}$	5.9
2.	$H_3(z)$	6.0 dB at BER of $10^{-2}$	5.10
3.	$H_5(z)$	1.5 dB at BER of $10^{-5}$	5.11
4.	$H_7(z)$	1.9 dB at BER of $10^{-3}$	5.13
5.	$H_9(z)$	3.2 dB at BER of $10^{-4}$	5.15
6.	$H_{14}(z)$	1.7 dB at BER of $10^{-5}$	5.17

**Table 6.5 : Performance analysis of proposed FRCS equaliser**

- The concept of hierarchical reinforcement in HKFRCS structure is mentioned in Section 5.2. Basically, this structure is identical to the previous one (FRCS), except that the RNN nodes are fed with the original input information in order to strengthen the knowledge base of the RNN processing units (refining expert opinions), which deliver the final output. Thus it seems logical that such a structure will yield a better performance in comparison to Conventional RNN due to strengthening of knowledge base at the output node. The BER performance

improvement using the proposed equaliser over the existing CRNN is observed in Table 6.6.

Sl.No.	Channels	Gain in SNR over conventional RNN	Figure No.
1.	$H_2(z)$	1.7 dB at BER of $10^{-4}$	5.9
2.	$H_3(z)$	6.0 dB at BER of $10^{-2}$	5.10
3.	$H_5(z)$	1.7 dB at BER of $10^{-5}$	5.11
4.	$H_7(z)$	2.4 dB at BER of $10^{-3}$	5.13
5.	$H_8(z)$	4.7 dB at BER of $10^{-4}$	5.14
6.	$H_9(z)$	3.5 dB at BER of $10^{-4}$	5.15
7.	$H_{14}(z)$	1.7 dB at BER of $10^{-4}$	5.17

**Table 6.6 : Performance analysis of proposed HKFRCS equaliser**

- The structure FRCS explained earlier has motivated to develop another hybrid configuration by swapping the FNN module with the RNN and vice-versa and has been explained in Section 5.3. The number of processing units and the external input in both remain the same. The RFCS structure provides an intermediate decision feedback while the FRCS one employs a pseudo decision feedback concept. The superiority in the performance of the proposed one over the CRNN equaliser is shown in Table 6.7.

Sl.No.	Channels	Gain in SNR over conventional RNN	Figure No.
1.	$H_3(z)$	6.0 dB at BER of $10^{-2}$	5.10
2.	$H_5(z)$	1.5 dB at BER of $10^{-5}$	5.11
3.	$H_6(z)$	3.4 dB at BER of $10^{-4}$	5.12
4.	$H_7(z)$	2.1 dB at BER of $10^{-3}$	5.13
5.	$H_9(z)$	2.4 dB at BER of $10^{-4}$	5.15

**Table 6.7 : Performance analysis of proposed RFCS equaliser**

- For the RTCS structure as mentioned in Section 5.4, the number of processing units remains the same as the CRNN equaliser. After the input signal is preprocessed in the RNN module, it is fed to the DCT transform block for further processing. As expected, such a proposed structure performs better than a CRNN due to the further signal de-correlation in the transform block followed by power normalisation as illustrated in Table 6.8.

Sl.No.	Channels	Gain in SNR over conventional RNN	Figure No.
1.	$H_2(z)$	1.3 dB at BER of $10^{-4}$	5.9
2.	$H_3(z)$	6.0 dB at BER of $10^{-2}$	5.10
3.	$H_5(z)$	1.8 dB at BER of $10^{-5}$	5.11
4.	$H_6(z)$	1.2 dB at BER of $10^{-4}$	5.12
5.	$H_7(z)$	1.5 dB at BER of $10^{-3}$	5.13
6.	$H_8(z)$	4.4 dB at BER of $10^{-4}$	5.14
7.	$H_9(z)$	3.6 dB at BER of $10^{-4}$	5.15
8.	$H_{11}(z)$	1.5 dB at BER of $10^{-4}$	5.16
9.	$H_{14}(z)$	1.7 dB at BER of $10^{-5}$	5.17

**Table 6.8 : Performance analysis of proposed RTCS equaliser**

- In the RNN framework another configuration has been discussed in Section 5.5 which employs fuzzy logic technique to a conventional RNN structure to tune the slope of the sigmoid activation function of the RNN nodes. This is basically aimed at incorporating another degree of freedom, thus increasing the adaptability of the conventional RNN and hence the proposed one is expected to perform better as observed in Table 6.9.

Sl.No.	Channels	Gain in SNR over conventional RNN	Figure No.
1.	$H_3(z)$	6.0 dB at BER of $10^{-2}$	5.10
2.	$H_5(z)$	1.8 dB at BER of $10^{-5}$	5.11
3.	$H_5(z)$	1.0 dB at BER of $10^{-4}$	5.12
3.	$H_7(z)$	2.0 dB at BER of $10^{-3}$	5.13
4.	$H_8(z)$	4.0 dB at BER of $10^{-4}$	5.14
5.	$H_9(z)$	2.8 dB at BER of $10^{-4}$	5.15

**Table 6.9 : Performance analysis of proposed FZTUNRNN equaliser**

- In the simulation study for analysing hybrid structures in RNN framework less training samples (200 only) have been used for training those networks. It has been observed from Figure 5.18, Figure 5.19, Figure 5.20, Figure 5.21 and Figure 5.22 that even if the CRNN based equaliser is trained using 1000 samples (exactly five times that of the training samples employed for all the proposed equalisers) its BER performance is almost comparable with the proposed ones, for which 200 training samples are required only.

Finally, the general inference derived from the simulation study carried out on various communication channels clearly indicates that the proposed neural equaliser structures are highly efficient in terms of structural complexity, BER performance and faster learning in comparison to their conventional counterparts (FNN and RNN based structures).

## 6.2 Limitations of the work

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

This thesis is generally concerned with the development of novel equaliser structures in neural domain for communication systems. In all the proposed structures the equaliser feedforward order ' $m$ ' has been restricted to the channel order ' $n_a$ '. Even though it was observed that increasing ' $m$ ' to a higher order can result in enhancement of BER performance, the selection of ' $m$ ' is constrained to a specific value such that the objective of the proposed work is preserved. Further, it is observed that because of such limitation on the value of ' $m$ ', the optimal performance is not achieved. In this

research work achieving the optimal Bayesian performance is not the major criterion, but development of novel neural equalisers with reduced structural complexity has been the main emphasis all along, even though it results in some performance loss. However, minimal performance degradation is noticed and a trade-off between structural complexity and performance is maintained in all the proposed structures.

The other limitation of the work pertains to analysis of stationary channel models only and time-varying channels and multi-channels have not been analysed in the present simulation study. Further, all the proposed equaliser structures and training algorithms developed in this research work are tested for applications using 2-PAM signalling only.

### **6.3 Scope of future work**

To conclude the thesis, following are some pointers for further work.

The first suggested area in which research can be undertaken follows from the limitation of the work presented in this chapter. Efforts can be put for utilizing the proposed equalisers developed in the present research work for time-varying channels and multipath fading channels. Attempts can be made to apply the proposed neural equalisers in FNN and RNN framework for blind equalisation of mobile communication channels and their suitability for combating co-channel interference can be found. New architectures based on ANN techniques and fuzzy techniques can be attempted even in a reduced structural complexity frame work to achieve near optimal performance for real-time applications. As bit error rate is the performance criterion of equalisation, more efficient training algorithms minimising an error function, which is a direct measure of BER, can be tried instead of the common gradient-based approaches used in either conventional BP or RTRL algorithms for faster convergence as well as optimal weight adaptation for improving the performance. This research work also can be extended to other efficient modulation schemes like 4-level PAM, QPSK etc., to increase transmission speed with limited channel bandwidth.

---

## REFERENCES

---

- [01] R.W. Lucky, "Automatic Equalisation of Digital Communication", *Bell system Technical Journal*, vol.44, pp. 547-588, April 1965.
- [02] R.P. Lippmann, "An Introduction to Computing with Neural Nets", *IEEE Acoustics, Speech and Signal Processing Magazine*, vol.4, pp. 4-22, April 1987.
- [03] B. Widrow and M.A. Lehr, "30 Years of Adaptive Neural Networks : Perceptron, Madaline, and Backpropagation", *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415-1442, September 1990.
- [04] Y.H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison Wesley, Reading, MA, 1989.
- [05] D.R. Hush and B.G. Horne, "Progress in Supervised Neural Networks-What's New Since Lippmann ?", *IEEE Signal Processing Magazine*, vol. 10, pp. 8-36, January 1993.
- [06] S. Haykin, *Neural networks – A Comprehensive Foundation*, New York : Macmillan, 1994.
- [07] C.F.N. Cowan, "Non-Linear Adaptive Equalisation", *Sixth International Conference on Digital Processing of Signals in Communication*, pp. 1-5, 2-6 September 1991.
- [08] 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 of Communication*, vol. 142, no. 3, pp. 165-171, June 1995.
- [09] C.F.N. Cowan, "Communications Equalisation using Adaptive Techniques", *IEE colloquium on Circuit Theory and DSP*, pp. 6/1-6/3, 18 February 1992.
- [10] C.F.N. Cowan, "Equalisation Using Non-Linear Adaptive Clustering", *IEE colloquium on Advances in Neural Networks for Control and Systems*, pp.17/1-17/3, 25-27 May 1994.
- [11] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, 1973.
- [12] S. Chen, G. J.Gibson and C.F.N. Cowan, "Adaptive Channel Equalisation using a Polynomial Perceptron Structure", *IEE Proceedings-I on Communication, Speech and Vision*, vol.137, no.5, pp. 257-264, October 1990.

- 
- [13] G.J. Gibson, S. Siu, S. Chen, C.F.N. Cowan and P.M. Grant, "The Application of Nonlinear Architectures to Adaptive Channel Equalisation", *IEEE International Conference on Communications, ICC'90 (Atlanta, USA)*, vol. 2, pp. 649-653, 16-19 April 1990.
- [14] 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, no. 8, pp. 1877-1884, August 1991.
- [15] R.P. Lippmann, "Pattern Classification Using Neural Networks", *IEEE Communications Magazine*, vol. 27, pp. 47-64, November 1989.
- [16] S.K. Pal and S. Mitra, "Multilayer Perceptron, Fuzzy Sets and Classification", *IEEE Transactions on Neural Networks*, vol. 3, pp.683-697, September 1992.
- [17] R.P. Lippmann, "A Critical Overview of Neural Network Pattern Classifiers", *Proceedings of IEEE Workshop on Neural Network for Signal Processing, (Princeton, NJ, USA)*, pp. 266-275, 30 September-2 October 1991.
- [18] G.J. Gibson, S. Siu and C.F.N. Cowan, "Multilayer Perceptron Structures Applied to Adaptive Equalisers for Data Communications", *International Conference on ASSP, ICASSP 89*, vol. 2, pp. 1183-1186, May 1989.
- [19] P. Power, F. Sweeney and C.F.N. Cowan, "Non-Linear MLP Channel Equalisation", *IEE Colloquium on Statistical Signal Processing*, pp. 3/1-3/6, 6 January 1999.
- [20] G. Kechriotis and E.S. Manolakos, "A VLSI Array Architecture for the on-line Training of Recurrent Neural Networks", *25<sup>th</sup> Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 506-510, 4-6 November 1991.
- [21] Yuan Yang, T. Ihalainen, J. Alhava, M. Renfors, "DSP Implementation of Low-Complexity Equaliser for Multicarrier Systems", *Seventh International Symposium on Signal Processing and its Applications*, vol. 2, pp. 271-274, 1-4 July 2003.
- [22] C.T. Yen, Wan-de Weng, and Y.T. Lin, "FPGA Realisation of a Neural-Network-Based Nonlinear Channel Equaliser", *IEEE Transaction on Industrial Electronics*, vol. 51, no. 2, pp. 472-479, April 2004.
- [23] R. Tanner, D.G.M. Cruickshank, C.Z.W.H. Sweatman and B. Mulgrew, "Receivers for Non-linearly Separable Scenarios in DS-CDMA", *Electronics Letters*, vol. 33, pp. 2103-2105, December 1997.
- [24] S.W. McLaughlin, "Shedding Light on the Future of Signal Processing for Optical Recording", *IEEE Signal Processing Magazine*, vol. 15, pp. 83-94, July 1998.
- [25] J.G. Proakis, "Equalisation Techniques for High Density Magnetic Recording", *IEEE Signal Processing Magazine*, vol. 15, pp. 73-82, July 1998.
- [26] S. Siu, G.J. Gibson and C.F.N. Cowan, "Decision Feedback Equalisation Using Neural Network Structures and Performance Comparison with

- Standard Architecture”, *IEE Proceedings-I on Communications, Speech and Vision*, vol. 137, no. 4, pp. 221-225, August 1990.
- [27] K. Raivio, O. Simula and J. Henriksson, “Improving Decision Feedback Equaliser Performance using Neural Network”, *Electronics Letters*, vol. 27, pp. 2151-2153, November 1991
- [28] M. Meyer and G. Pfeiffer, “Multilayer Perceptron Based Decision Feedback Equalisers for Channels with Intersymbol Interference”, *IEE Proceedings-I on Communication, Speech and Vision*, vol. 140, no. 6, pp. 420-424, December 1993.
- [29] S. Chen, B. Mulgrew and S. Mc Laughlin, “Adaptive Bayesian Equaliser with Decision Feedback”, *IEEE Transactions on Signal Processing*, vol. 41, pp. 2918-2927, September 1993.
- [30] M.J. Bradley and P. Mars, “A Critical Assessment of Recurrent Artificial Neural Networks as Adaptive Equalisers in Digital Communications”, *IEE colloquium on Applications of Neural Networks to Signal processing*, pp. 11/1-11/4, December 1994.
- [31] M.J. Bradley and P. Mars, “Application of Recurrent Neural Networks to Communication Channel Equalisation”, *International Conference on Acoustics, Speech and Signal Processing, ICASSP-95*, vol. 5, pp. 3399-3402, 9-12 May 1995.
- [32] G. Kechriotis, E. Zervas and E.S. Manolakos, “Using Recurrent Neural Networks for Adaptive Communication Channel Equalisation”, *IEEE Transaction on Neural Networks*, vol. 5, no. 2, pp. 267-278, March 1994.
- [33] J.D. Ortiz-Fuentes and M.L. Forcada, “A Comparison between Recurrent Neural Network Architectures for Digital Equalisation”, *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP-97*, vol. 4, pp. 3281-3284, 21-24 April 1997.
- [34] R. Parishi, E.D.D Claudio, G. Orlandi and B.D. Rao, “Fast Adaptive Digital Equalisation by Recurrent Neural Networks”, *IEEE Transactions Neural Networks*, vol. 45, no. 1, pp. 2731-2739, November 1997.
- [35] H. Nyquist, “Certain Topics in Telegraph Transmission Theory”, *Transactions of AIEE*, vol. 47, pp. 617-644, February 1928.
- [36] E.A. Lee and D.G. Messerschmitt, *Digital Communications*, Second Edition, Allied Publishers Limited, 1996.
- [37] J.G. Proakis, *Digital Communications*, New York: Mc Graw-Hill Inc.1995.
- [38] S. Haykin, *Digital Communications*, John Wiley and Sons, 1988.
- [39] B. Widrow and S.D. Stearns, *Adaptive Signal Processing*, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1985.
- [40] G.D. Forney, “Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference”, *IEEE Transactions on Information Theory*, vol. 18, pp. 363-378, May 1972.
- [41] G.D. Forney, “The Viterbi Algorithm”, *Proceedings of the IEEE*, vol. 61, pp. 268-278, March 1973.

- 
- [42] D.A. George, R.R. Bowen and J.R. Storey, "An Adaptive Decision Feedback Equaliser", *IEEE Transactions on Communication Technology*, vol. 19, no. 3, pp. 281-293, June 1971.
- [43] D. Godard, "Channel Equalisation Using Kalman Filter for Fast Data Transmission", *IBM Journal Research Development*, vol. 18, pp. 267-273, May 1974.
- [44] J. Makhoul, "A Class of All-Zero Lattice Digital Filters: Properties and Applications", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 4, pp. 304-314, August 1978.
- [45] J.R. Treichler, I. Fijalkow and C.R. Johnson, "Fractionally Spaced Equalisers – How Long Should They Really be?", *IEEE Signal Processing Magazine*, pp. 65-81, May 1996.
- [46] S.U.H. Qureshi, "Adaptive Equalisation", *Proceedings of the IEEE*, vol. 73, no. 9, pp. 1349-1387, September 1985.
- [47] S. Chen, G.J. Gibson, C.F.N. Cowan and P.M. Grant, "Reconstruction of Binary Signals using an Adaptive Radial Basis Function Equaliser", *Signal Processing (Eurasip)*, vol. 22, pp. 77-93, January 1991.
- [48] 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, no. 2, pp. 302-309, March 1991.
- [49] L.X. Wang and J.M. Mendel, "Fuzzy Adaptive Filters, with Application to Non-linear Channel Equalisation", *IEEE Transactions on Fuzzy Systems*, vol. 1, no.3, pp. 161-170, August 1993.
- [50] C.Z.W.H. Sweatman, B. Mulgrew and G.J. Gibson, "Two Algorithms for Neural-Network Design and Training with Application to Channel Equalisation", *IEEE Transactions on Neural Networks*, vol. 9, no. 3, pp. 533-543, May 1998.
- [51] B. Mulgrew, "Applying Radial Basis Functions", *IEEE Signal Processing Magazine*, vol. 13, pp. 50-65, March 1996.
- [52] S.K. Patra and B. Mulgrew, "Efficient Architecture for Bayesian Equalisation Using Fuzzy Filters", *IEEE Transactions on Circuits and Systems II : Analog and Digital Signal Processing*, vol. 45, no. 7, pp. 812-820, July 1998.
- [53] W.R. Kirkland and D.P. Taylor, "On the Application of Feedforward Neural Networks to Channel Equalisation", *Proceedings of International Joint Conference on Neural Networks (New York)*, vol. 2, pp. 919-924, 7-11 June 1992.
- [54] M. Peng, C.L. Nikias and J.G. Proakis, "Adaptive Equalisation with Neural Networks : New Multilayer Perceptron Structures and their Evaluation", *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'92*, vol. 2, pp. 301-304, 23-26 March 1992.
- [55] R.J. Williams and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks", *Neural Computation*, vol. 1, pp. 270-280, 1989.



- [56] P.R. Chang, B.F. Yeh and C.C.Chang, "Adaptive Packet Equalisation for Indoor Radio Channel Using Multilayer Neural Networks", *IEEE Transactions on Vehicular Technology*, vol. 43, no. 3, pp. 773-780, August 1994.
- [57] Q. Zhang and A. Benveniste, "Wavelet Networks", *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 889-898, November 1992.
- [58] P.R. Chang and B.F. Yeh, "Nonlinear Communication Channel Equalisation Using Wavelet Neural Networks", *Proceedings of IEEE International Conference on Neural Networks*, vol. 6, pp. 3605-3610, 27 June – 2 July 1994.
- [59] K.A. Al-Mashouq and I.S.Reed, "The Use of Neural Nets to Combine Equalisation with Decoding for Severe Intersymbol Interference Channels", *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 982-988, November 1994.
- [60] K.Y .Lee, S.Y. Lee and S. McLaughlin, " A Neural Network Equaliser with the Fuzzy Decision Learning Rule", *Proceedings of 7<sup>th</sup> IEEE Workshop on Neural Networks for Signal Processing, NNSP-97 (Amelia Island, FL, USA)*, pp. 551-559, 24-26 September 1997.
- [61] T. Adali, X. Liu and M.K. Sonmez, "Conditional Distribution Learning with Neural Network and Its Application to Channel Equalisation", *IEEE Transactions on Signal Processing*, vol. 45, no. 4, pp. 1051-1064, April 1997.
- [62] J.J. Xue and X.H. Yu, "A Mean Field Annealing Partially – Connected Neural Equaliser for Pan-European GSM System", *Proceedings of IEEE International Conference on Communications*, vol. 2 (Dallas, TX, USA), pp. 701-705, 23-27 June 1996.
- [63] L.O. Chua and L. Yang, "Cellular Neural Networks: Theory", *IEEE Transactions on Circuits and Systems*, vol. 35, no. 10, pp. 1257-1272, October 1988.
- [64] B.W. Lee and B.J. Sheu, "Parallel Hardware Annealing for Optimal Solutions on Electronic Neural Networks", *IEEE Transactions on Neural Networks*, vol. 4, no. 4, pp. 588-598, July 1993.
- [65] Z. Xiang, G. Bi and T. Le-Ngoc, "Polynomial Perceptrons and their Applications to Fading Channel Equalisation and Co-Channel Interference Suppression", *IEEE Transactions on Signal Processing*, vol. 42, no. 9, pp. 2470-2480, September 1994.
- [66] W.S. Gan, J.J. Soraghan, and T.S. Durrani, "A new Functional-link based Equaliser", *Electronics Letters*, vol. 28, no. 17, pp. 1643-1645, 13 August 1992.
- [67] 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, no. 11, pp. 1358-1362, November 1997.

- [68] J.C. Patra and R.N. Pal, "A Functional Link Artificial Neural Network for Adaptive Channel Equalisation", *Signal Processing (Eurasip)*, vol. 43, pp. 181-195, May 1995.
- [69] Z.J. Xiang, G.G. Bi and C.B. Schlegel, "A New Lattice Polynomial Perceptron and its Application to Fading Channel Equalisation and Adjacent – Channel Interference Suppression", *Proceeding of IEEE International Conference on Communications (Seattle, WA, USA)*, vol. 1, pp. 478-482, IEEE, 18-22 June 1995.
- [70] Zerguine, A. Shafi and M. Bettayab, "Multilayer Perceptron-Based DFE with Lattice Structure", *IEEE Transactions on Neural Networks*, vol. 12, no. 3, pp. 532-545, May 2001.
- [71] P. Power, F. Sweeney and C.F.N. Cowan, "EA Crossover Schemes for a MLP Channel Equaliser", *6<sup>th</sup> IEEE International Conference on Electronics, Circuits and Systems, ICECS'99*, vol. 1, pp. 407-410, 5-8 September 1999..
- [72] S. Chen, S.R. Gunn and C.J. Harris, "The Relevance Vector Machine Technique for Channel Equalisation Application", *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1529-1532, November 2001.
- [73] D.J. Sebald and J.A. Bucklew, "Support Vector Machine Techniques for Nonlinear Equalisation", *IEEE Transactions on Signal Processing*, vol. 48, no. 11, pp. 3217-3226, November 2000.
- [74] S. Chen, S. Gunn and C.J. Harris, "Decision Feedback Equaliser Design Using Support Vector Machines", *IEE Proceedings on Vision, Image and Signal Processing*, vol. 147, no. 3, pp. 213-219, June 2000.
- [75] S. Chen, B. Mulgrew, E.S. Chng, and G.J. Gibson, "Space Translation Properties and the Minimum-BER Linear-Combiner DFE", *IEE Proceedings on Communication*, vol. 145, no. 5, pp. 316-322, October 1998.
- [76] S. Chen, E.S. Chng, B. Mulgrew and G. Gibson, "Minimum-BER Linear Combiner DFE", *IEEE International Conference on Communications, ICC'96*, vol. 2, pp. 1173-1177, 23-27 June 1996.
- [77] S. Chen, B. Mulgrew and L. Hanzo, "Least Bit Error Rate Adaptive Nonlinear Equalisers for Binary Signalling", *IEE Proceedings on Communication*, vol. 150, no. 1, pp. 29-36, February 2003.
- [78] J. Choi, A.C. de C. Lima and S. Haykin, "Unscented Kalman Filter-Trained Recurrent Neural Equaliser for Time-Varying Channels", *IEEE International Conference on Communications, ICC'03*, vol. 5, pp. 3241-3245, May 2003.
- [79] R.L. Valcarce, "Realizable Linear and Decision Feedback Equalisers : Properties and Connections", *IEEE Transactions on Signal Processing*, vol. 52, pp. 757-773, March 2004.
- [80] N. Al-Dhahir, and J.M. Cioffi, "MMSE Decision-Feedback Equalisers : Finite-Length Results", *IEEE Transaction Information Theory*, vol. 41, no. 4, pp. 961-975, July 1995.

- [81] S. Chen, B. Mulgrew and L. Hanzo, "Asymptotic Bayesian Decision Feedback Equaliser Using a Set of Hyperplanes", *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 48, pp. 3493-3500, December 2000.
- [82] R. Jr Chen and Wen-Rong Wu, "Adaptive Asymptotic Bayesian Equalization Using a Signal Space Partitioning Technique", *IEEE Transactions on Signal Processing*, vol. 52, pp. 1376-1386, May 2004.
- [83] M.J. Lopez, K.Zangi, and J-F. Cheng, "Reduced-Complexity MAP Equaliser for Dispersive Channels", *52<sup>nd</sup> IEEE Vehicular Technology Conference, VTC 2000*, vol. 3, pp. 1371-1375, 24-28 Septemeber 2000.
- [84] P. Chandra Kumar, P. Saratchandran and N. Sundararajan, "Minimal Radial Basis Function Neural Networks for Nonlinear Channel Equalisation", *IEE Proceedings on Vision, Image and Signal Processing*, vol. 147, no. 5, pp. 428-435, October 2000.
- [85] C.N. Chen, K. H Chen and T.D. Chiueh, " Algorithm and Architecture Design for a Low Complexity Adaptive Equaliser", *IEEE International Symposium on Circuits and Systems, ISCAS'03*, vol. 2 , pp. 304-307, May 2003.
- [86] C.E. Siong, S. Chen and D. Rajan, "Optimum Delay Order Selection for Linear Equalisation Problems", *Proceedings IEEE, International Symposium of Intelligent Signal Processing and Communication Systems (Awaji Island, Japan)*, pp. 850-853, December 2003.
- [87] S. Choi and Te-Won Lee, "A New Equalisation Algorithm based on minimizing error negentropy", *58<sup>th</sup> IEEE Vehicular Technology Conference*, vol. 1, pp. 241-245, October 2003.
- [88] S. Haykin, *Adaptive Filter Theory*, Englewood Cliff, NJ, USA, Prentice Hall, 1991.
- [89] B. Farhang-Boroujeny, *Adaptive Filters : Theory and Applications*, John Wiley and Sons,1999
- [90] S. Chen, B. Mulgrew and P.M. Grant, "A Clustering Technique for Digital Communication Channel Equalisation Using Radial Basis Function Networks", *IEEE Transactions on Neural Networks*, vol.4, pp. 570-579, July 1993.
- [91] K. Abend and B.D. Fritchman, "Statistical Detection for Communication Channels with Intersymbol Interference", *Proceedings of the IEEE*, vol. 58, pp. 779-785, May 1970.
- [92] 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.
- [93] M.T. Ozeden, A.H. Kayran and E. Panayirei, "Adaptive Volterra Channel Equalisation with Lattice Orthogonalisation", *IEE Proceedings – Communication*, vol. 145, pp. 109-115, April 1998.

- [94] K.Y. Lee, "Fuzzy Adaptive Decision Feedback Equaliser", *Electronics Letters*, vol. 30, pp. 749-751, 12 May 1994.
- [95] P.A. Voois, I. Lee and J.M. Cioffi, "The Effect of Decision Delay in Finite-Length Decision Feedback Equalisation", *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 618-621, March 1996.
- [96] E.S. Cheng, B. Mulgrew, S. Chen and G. Gibson, "Optimum Lag and Subset Selection for a Radial Basis Function Equaliser", *Proceedings of 5<sup>th</sup> IEEE Workshop Neural Networks for Signal Processing (Cambridge, USA)*, pp. 593-602, 31 August – 2 September 1995.
- [97] J.K. Satapathy, S. Das and C.J. Harris, "Application of hierarchical knowledge in a neural network domain for designing a high performance adaptive digital channel equaliser", *International Conference on Signal Processing, Applications and Technology, ICSPAT'2000 (Dallas, Texas, USA)*, 16-19 October 2000.
- [98] J.K. Satapathy, S. Das and C.J. Harris, "A novel neural equaliser structure using an embedded self-breeding genetic concept in an orthonormal basis-function framework", *2<sup>nd</sup> International Conference on Information, Communications and Signal Processing, ICICS'99 (Singapore)*, pp.3E 4.6, 7-10 December 1999.
- [99] M.A. Shamma, "Improving the Speed and Performance of Adaptive Equalisers via Transform based Adaptive Filtering", *14<sup>th</sup> International Conference on Digital Signal Processing, DSP 2002*, vol. 2, pp. 1301-1304, July 2002.
- [100] J.K. Satapathy, S. Das and C.J. Harris, "Embedding discrete cosine transform in a neural network paradigm for performance enhancement of an adaptive channel equaliser", *Proceedings of the ICSC Symposia on NC'2000 (Berlin, Germany)*, 23-26 May 2000.
- [101] D.F. Marshall, W.K. Jenkins and J.J. Murphy, "The Use of Orthogonal Transforms for Improving Performance of Adaptive Filters", *IEEE Transactions on Circuits and Systems*, vol. 36, no. 4, pp. 474-484, April 1989.
- [102] F. Beaufays, "Transform-Domain Adaptive Filters: An Analytical Approach", *IEEE Transactions on Signal Processing*, vol. 43, no. 2, pp. 422-431, February 1995.
- [103] D. Driankov, H. Hellendoorn and M. Reinfrank, *An Introduction to Fuzzy Control*, Springer-Verlag, 1993.
- [104] M. Brown and C.J. Harris, *Neurofuzzy Adaptive Modelling and Control*, Prentice Hall, 1994.
- [105] A.F. Stronach, P. Vas and M. Neuroth, "Implementation of Intelligent Self-organising Controllers in DSP Controlled Electromechanical drives", *IEE Proceedings of Control Theory Application*, vol. 144, no. 4, pp. 324-330, July 1997.
- [106] K.L. Kientitz, "Controller Design using Fuzzy Logic-A case Study", *Automatica*, vol. 29, pp. 549-554, March 1993.

- 
- [107] J.K. Satapathy and C.J. Harris, "Application of Fuzzy-Tuned Adaptive Feedforward Neural Networks for Accelerating Convergence in Identification", *3<sup>rd</sup> International Conference on Industrial Automation*, pp. 6.1, June 1999.
- [108] J.S.R. Jang, *Neuro-fuzzy and Soft Computing*, Prentice-Hall International, Inc.
- [109] 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 1992.
- [110] J.M. Mendel, "Fuzzy Logic System for Engineering: A Tutorial", *Proceedings of the IEEE*, vol. 83, pp. 345-377, March 1995.
- [111] S. Sundaralingam and K. Sharman, "Genetic Evolution of Adaptive Filters", *DSP Proceedings, UK, London*, pp. 47-53, December 1997.
- [112] S. Lambotharan and J.A. Chambers, "A New Blind Equalisation Structure for Deep-Null Communication Channels", *IEEE Transactions on Circuits and Systems-II, Analog and Digital Signal Processing*, vol. 45, no. 1, pp. 108-114, January 1998.
- [113] M.T. Madeira da Silva and M. Gerken, "A RNN-LC Hybrid Equaliser", *XI European Signal Processing Conference (France)*, pp. 341-344, September 2002.

This appendix presents the **Back-Propagation (BP)** [06] algorithm.

### **A : Description of the Training Algorithm**

The weights and thresholds of a feedforward neural network are generally updated by using the *Back-Propagation* (BP) training algorithm, which is a stochastic gradient descent optimization procedure. The BP algorithm is an iterative algorithm and adjusts the weights so as to minimize any differentiable cost function such as the mean square error (MSE). In back propagation, the output value is compared with the desired output, resulting in an error signal. The error signal is fed back through the network connections and the weights are adjusted so as to minimise the error. The increments used in updating the weights,  $\Delta w_{ij}$  and threshold levels,  $\Delta th_j$  of the  $l^{\text{th}}$  layer can be accomplished by the following rules.

$$\Delta w_{ij}^{(l)}(n+1) = \eta \delta_j^{(l)}(n) y_j^{(l-1)}(n) + \alpha \Delta w_{ij}^{(l)}(n) \quad (\text{A.1})$$

and

$$\Delta th_j^{(l)}(n+1) = \beta \delta_j^{(l)}(n) \quad (\text{A.2})$$

where  $\eta$  is the *learning-rate parameter*,  $\alpha$  is the *momentum parameter*,  $\beta$  is the *threshold level adaptation gain* and layer  $l \in [1, 2, \dots, L]$ .

The error signal  $\delta_j^{(l)}(n)$  for layer  $l$  is calculated starting from the output layer  $L$

$$\delta_j^{(L)}(n) = \{d_j(n) - y_j^{(L)}(n)\} \{1 - y_j^{2(L)}(n)\} / 2 \quad (\text{A.3})$$

and recursively back propagating the error signal to the lower layers

$$\delta_j^{(l)}(n) = \{1 - y_j^{2(l)}(n)\} \sum_q \delta_q^{(l+1)}(n) w_{qj}^{(l+1)}(n) / 2 \quad (\text{A.4})$$

where  $l \in [1, 2, \dots, L-1]$ ,  $q$  is the overall neurons in the layer above neuron  $j$  and  $d_j(n)$  is the desired output.

This appendix presents the *Real-Time-Recurrent-Learning (RTRL) algorithm* [55].

**B : Description of the Training Algorithm**

The Recurrent neural network chosen here has  $nx$  external inputs and  $nr$  fully interconnected recurrent processing units. Thus input of RNN is a vector  $\mathbf{u}(n)$ ,  $l^{th}$  element of which  $u_l(n)$  is defined as

$$u_l(n) = \begin{cases} y_k(n), & 1 \leq k \leq nr \\ x_k(n), & 1 \leq k \leq nx \end{cases} \quad \text{for } 1 \leq l \leq (nx+nr) \quad (\text{B.1})$$

The output of  $k^{th}$  neuron of RNN at time  $n$

$$y_k(n) = \frac{1 - e^{-\phi c_k(n)}}{1 + e^{-\phi c_k(n)}}, \text{ where } c_k(n) = \sum_{l=1}^{nx+nr} w_{kl}(n) \cdot u_l(n), \quad 1 \leq k \leq nr \quad (\text{B.2})$$

Sigmoidal activation function with slope parameter  $\phi$  has been considered for each processing unit of RNN structure.  $\mathbf{W}$  denotes  $nr$  by  $(nx + nr)$  weight matrix of RNN.

The final output of the proposed equaliser structure is taken from the output of  $j^{th}$  neuron of RNN. By comparing  $y_j(n)$  with the desired value  $d(n)$ , the error  $e_j(n)$  is calculated.

$$e_j(n) = d(n) - y_j(n) \quad (\text{B.3})$$

The instantaneous sum of squared error at time  $n$

$$J(n) = \frac{1}{2} \sum_{j=1}^{nr} e_j^2(n) \quad (\text{B.4})$$

The objective here is to minimise a *cost function*, obtained by summing  $J(n)$  over all time  $n$ ; that is,

$$J_{total} = \sum_n J(n) \quad (\text{B.5})$$

To accomplish this objective the method of steepest descent is used, which requires knowledge of the gradient matrix, written as

$$\nabla_w J_{total} = \sum_n \frac{\partial J(n)}{\partial \mathbf{W}} \quad (\text{B.6})$$

In order to develop a learning algorithm for training the network in real time, an instantaneous *estimate* of the gradient is necessary following an approximation to the method of steepest descent.

For the case of a particular weight  $w_{kl}(n)$ , the incremental change  $\Delta w_{kl}(n)$  made at time index  $n$

$$\Delta w_{kl}(n) = -\lambda \frac{\partial J(n)}{\partial w_{kl}(n)}, \quad (\text{B.7})$$

where  $\lambda$  is a *learning -rate parameter* of RNN.

From Equations (B.6) and (B.7)

$$\frac{\partial J(n)}{\partial w_{kl}(n)} = -\sum_{j=1}^n e_j(n) \frac{\partial y_j(n)}{\partial w_{kl}(n)} \quad (\text{B.8})$$

Since  $e_j(n)$  is known at all times, determination of the partial derivative is required to implement RTRL algorithm.

A *sensitivity parameter* is described by a triply indexed set of variables  $\{p_{kl}^j\}$ , where

$$p_{kl}^j(n) = \frac{\partial y_j(n)}{\partial w_{kl}(n)}, \quad 1 \leq k \leq nr \text{ and } 1 \leq l \leq (nr+nx) \quad (\text{B.9})$$

The evaluation of this partial derivative is carried out as follows.

$$p_{kl}^j(n+1) = \mathbf{F}' \{c_j(n)\} \left[ \sum_{i=1}^{nr} w_{ji}(n) \cdot p_{kl}^i(n) + \partial_{kj} u_l(n) \right] \quad (\text{B.10})$$

with initial condition  $p_{kl}^j(0) = 0$

$$\text{The derivative } \mathbf{F}' \{c_j(n)\} = \{1 - y_j(n+1)^2\} (\phi / 2) \quad (\text{B.11})$$

$\partial_{kj}$  is a *Kronecker delta* defined as

$$\begin{aligned} \partial_{kj} &= 1 \text{ for } k = j \\ &= 0, \text{ otherwise.} \end{aligned} \quad (\text{B.12})$$

Updation of the connection weight  $w_{kl}(n)$  is carried out as per the following expressions.

$$\Delta w_{kl}(n) = \lambda \sum_{j=1}^{nr} e_j(n) \cdot p_{kl}^j(n), \quad 1 \leq k \leq nr \text{ and } 1 \leq l \leq (nx+nr) \quad (\text{B.13})$$

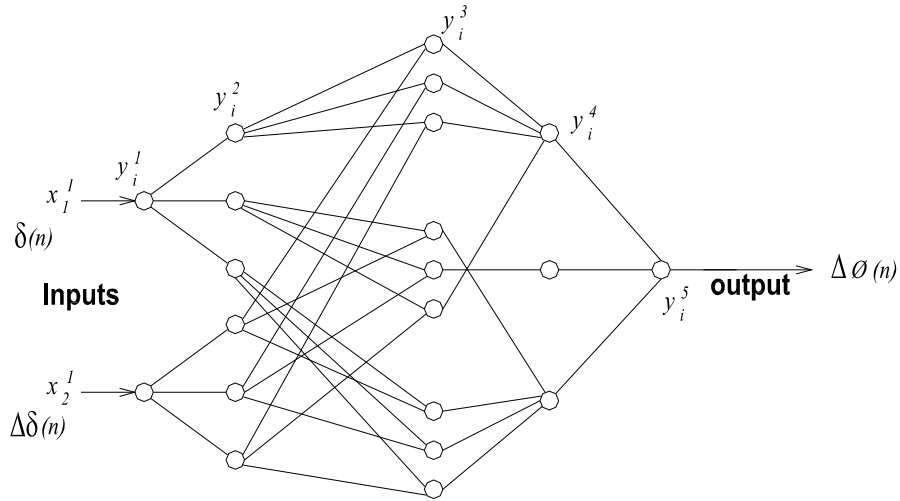
$$\text{Finally, } w_{kl}(n+1) = w_{kl}(n) + \Delta w_{kl}(n) \quad (\text{B.14})$$



A fuzzy controller block [107] chosen to adapt the slope of the sigmoid is represented by a network consisting of five layers with two inputs and a single output is discussed in this appendix.

**C : Fuzzy Controller Structure**

The construction of a five layer network given in Figure C.1 is described below.



**Figure C.1: A fuzzy controller structure**

The first layer is an input layer with one node for each controller variable. The node output is given by

$$y_i^1 = x_i^1 \tag{C.1}$$

for  $i = 1, \dots, n1$ , where  $n1$  is the total number of nodes in layer 1. The interconnecting weights between the first and the second layer are all unity. The second layer is made up of nodes representing Gaussian membership function. The total number of nodes is equal to the total number of fuzzy sets associated with the input variables. The node output of this layer is given by the expression

$$y_i^2 = \exp(-((y_i^1 - cn_i) / \sigma_i)^2) \tag{C.2}$$

for  $i = 1, \dots, n2$ , where  $n2$  is the total number of nodes in layer 2.

The interconnecting weights between the second and the third layer are all unity. The third layer consists of nodes implementing fuzzy AND operator. Each node in this layer represents a fuzzy rule and the output is given by

$$y_i^3 = \min(y_j^2) \quad (C.3)$$

for  $i = 1, \dots, n3$ , and  $j = 1, \dots, n2$ , where  $n3$  is the total number of nodes in layer 3. The interconnecting weights between the third and fourth layer are all unity.

The fourth layer consists of nodes realising the bounded sum form of the fuzzy OR operator. The number of nodes is equal to the number of fuzzy sets representing the controller variable output. The node output is given by

$$y_i^4 = \min(1, \sum y_j^3) \quad (C.4)$$

for  $i = 1, \dots, n4$ , where  $n4$  is the total number of nodes in layer 4. The fifth layer comprises of nodes implementing a centre-of-area (COA) defuzzification algorithm. The weights of interconnection between the nodes in the fourth and fifth layers are the products of the centre and width of the membership function associated with the fuzzy set for the controller output variables, which are given by

$$y_i^5 = \frac{\sum_{j=1}^{n4} cn_j \sigma_j y_j^4}{\sum_{j=1}^{n4} \sigma_j y_j^4} \quad (C.5)$$

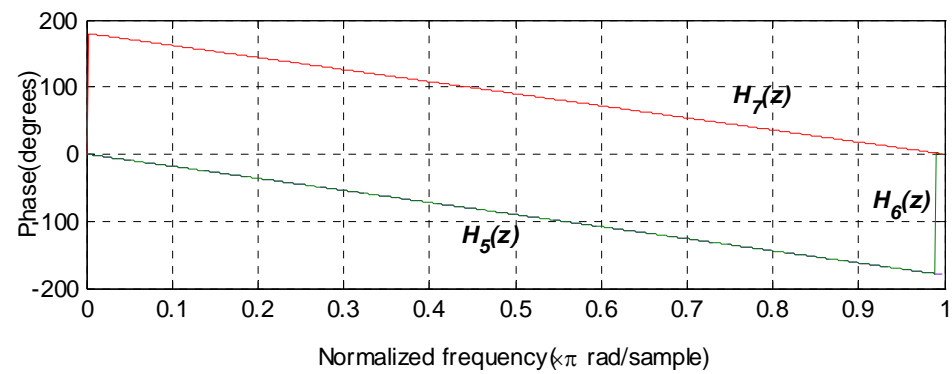
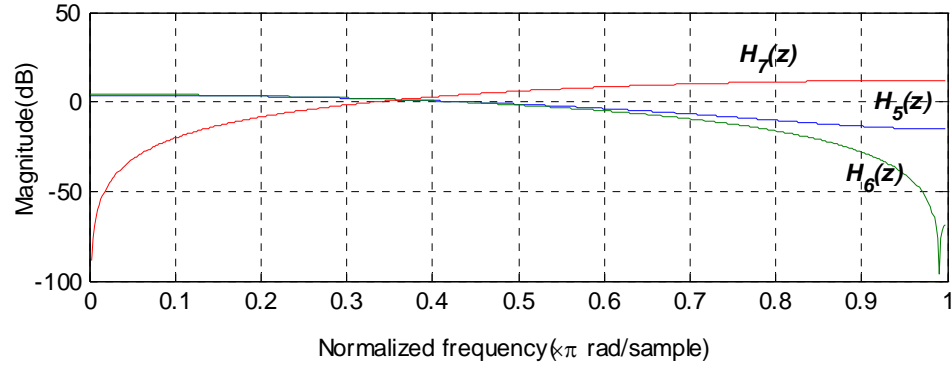
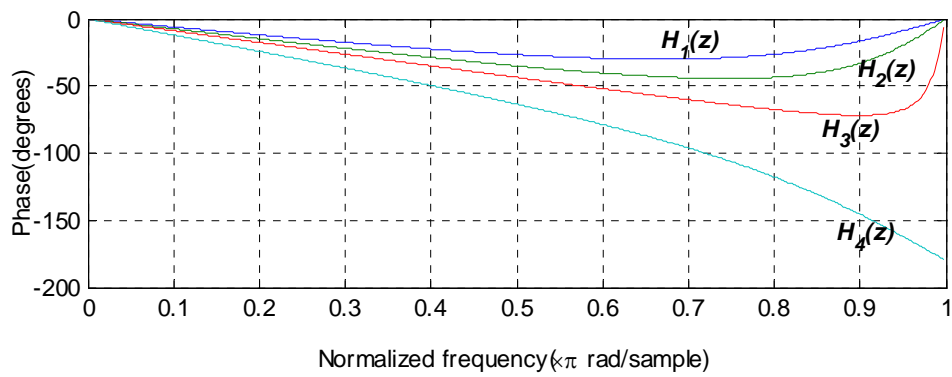
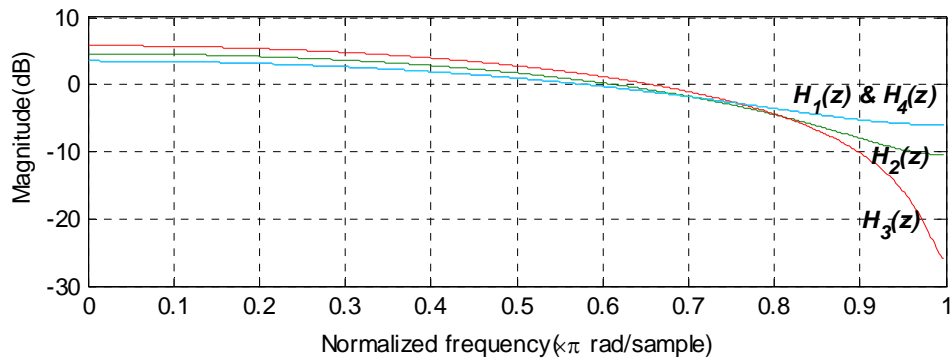
for  $i = 1, \dots, n5$ , where  $n5$  is the total number of nodes in layer 5.

## **APPENDIX-D**

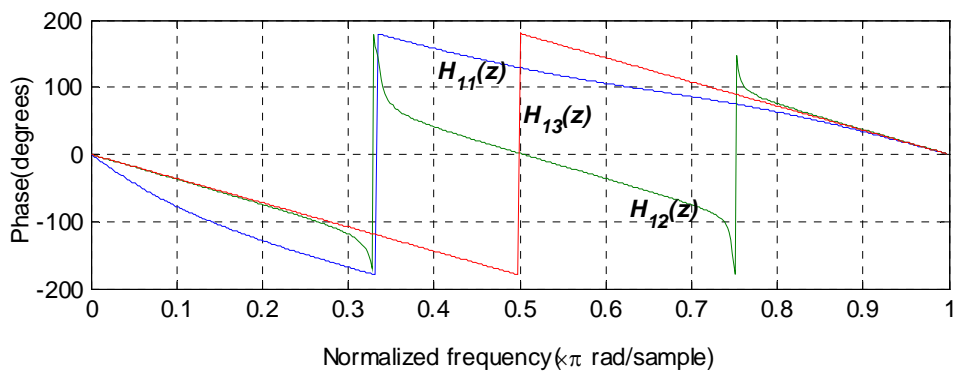
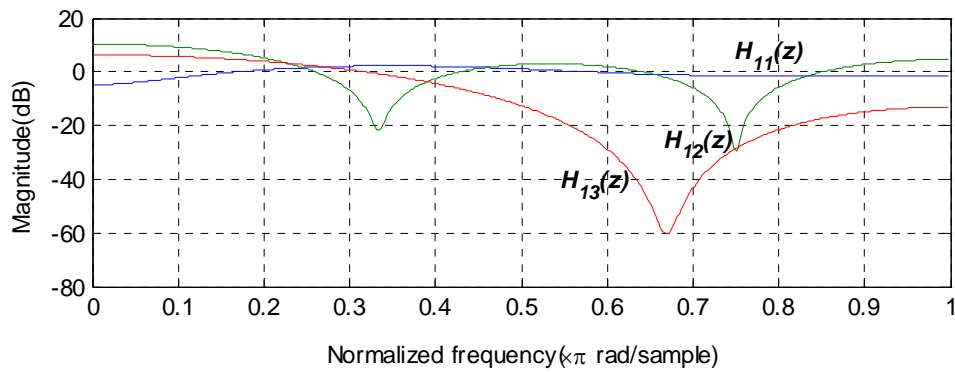
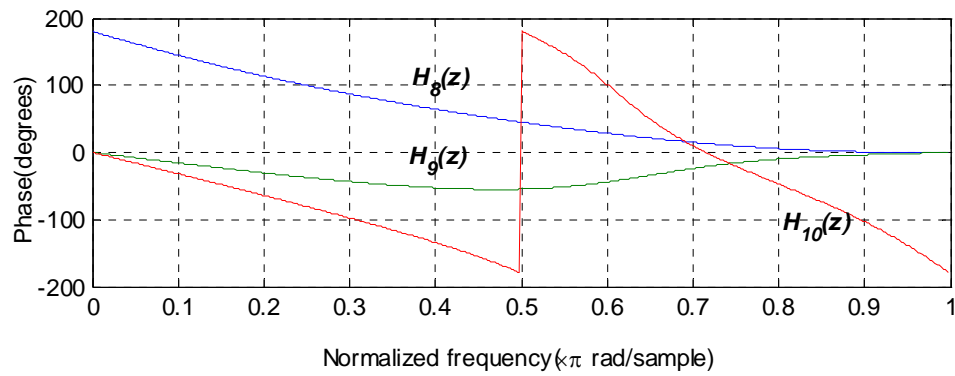
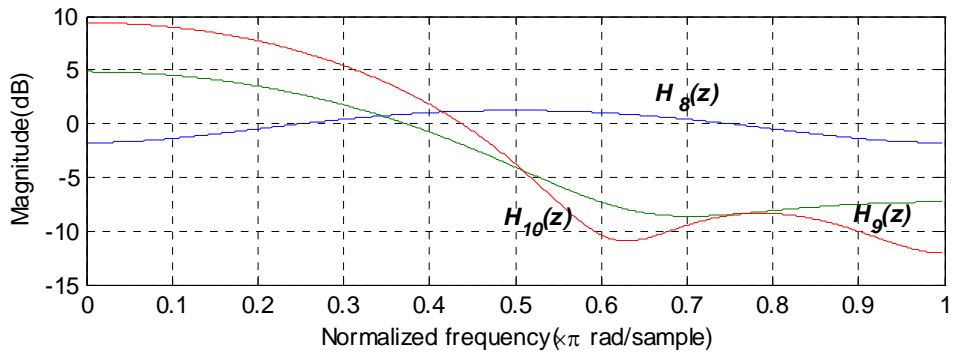
Following channels have been used as examples for BER performance evaluation of proposed FNN based and RNN based cascaded equalisers in the simulation study.

No.	Transfer functions of channels
$H_1(z)$	$1 + 0.5 z^{-1}$
$H_2(z)$	$1 + 0.7 z^{-1}$
$H_3(z)$	$1 + 0.95 z^{-1}$
$H_4(z)$	$0.5 + 1 z^{-1}$
$H_5(z)$	$0.3482 + 0.8704 z^{-1} + 0.3482 z^{-2}$
$H_6(z)$	$0.4084 + 0.8164 z^{-1} + 0.4084 z^{-2}$
$H_7(z)$	$1 - 2 z^{-1} + 1 z^{-2}$
$H_8(z)$	$0.407 - 0.815 z^{-1} - 0.407 z^{-2}$
$H_9(z)$	$0.7255 + 0.584 z^{-1} + 0.3627 z^{-2} + 0.0724 z^{-3}$
$H_{10}(z)$	$0.35 + 0.8 z^{-1} + 1 z^{-2} + 0.8 z^{-3}$
$H_{11}(z)$	$-0.2052 - 0.5131 z^{-1} + 0.7183 z^{-2} + 0.3695 z^{-3} + 0.2052 z^{-4}$
$H_{12}(z)$	$0.9413 + 0.3841 z^{-1} + 0.5684 z^{-2} + 0.4201 z^{-3} + 1 z^{-4}$
$H_{13}(z)$	$0.227 + 0.46 z^{-1} + 0.688 z^{-2} + 0.46 z^{-3} + 0.227 z^{-4}$
$H_{14}(z)$	$(1+0.5 z^{-1}) - 0.9 (1+0.5 z^{-1})^3$
$H_{15}(z)$	$(0.3482 + 0.8704 z^{-1} + 0.3482 z^{-2}) + 0.2 (0.3482 + 0.8704 z^{-1} + 0.3482 z^{-2})^2$

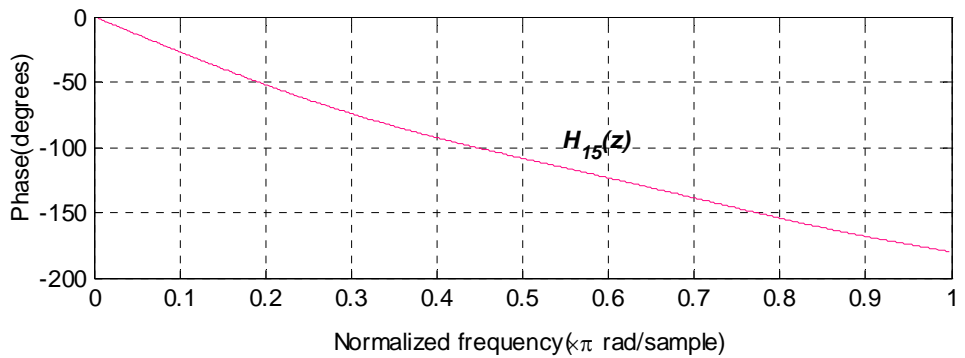
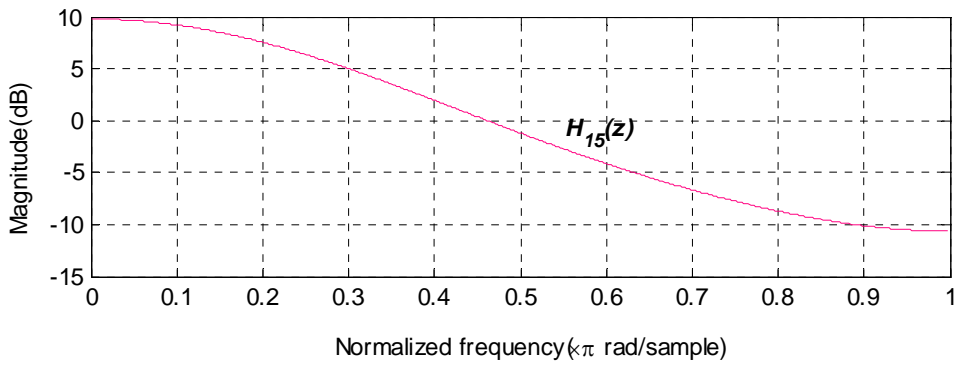
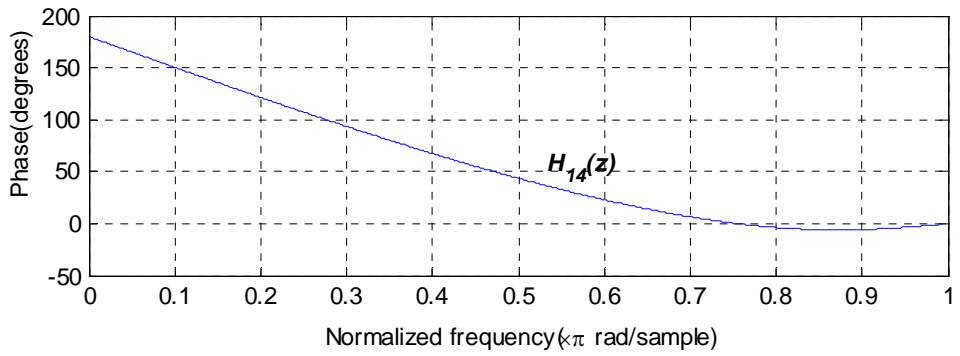
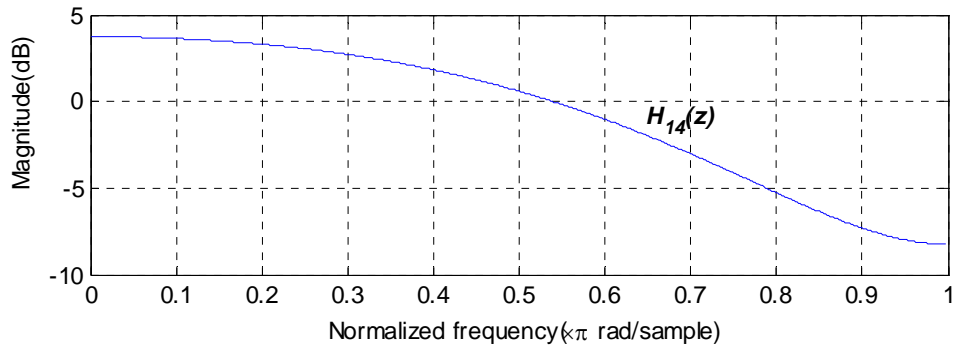
**Frequency Responses of the Channels**



### Frequency Responses of the Channels



### Frequency Responses of the Channels



## **List of Papers Published**

- [01] J.K. Satapathy, G. Panda and S. Das, “Development of a novel digital communication channel equaliser using real time implementable multilayer neural structure”, *International Conference on Signal Processing, Applications and Technology*, ICSPAT’96, Boston, USA, pp.1378-1381, October 1996.
- [02] J.K. Satapathy, S. Das and C.J. Harris, “Embedding discrete cosine transform in a neural network paradigm for performance enhancement of an adaptive channel equaliser”, *Proceedings of the ICSC Symposia on NC’2000 (Berlin, Germany)*, 23-26 May 2000.
- [03] J.K. Satapathy, S. Das and C.J. Harris, “Application of hierarchical knowledge in a neural network domain for designing a high performance adaptive digital channel equaliser”, *International Conference on Signal Processing, Applications and Technology, ICSPAT’2000 (Dallas, Texas, USA)*, 16-19 October 2000.
- [04] J.K. Satapathy, S. Das and C.J. Harris, “A novel neural equaliser structure using an embedded self-breeding genetic concept in an orthonormal basis-function framework”, *2<sup>nd</sup> International Conference on Information, Communications and Signal Processing, ICICS’99 (Singapore)*, pp.3E 4.6, 7-10 December 1999.

## **List of Papers to be submitted**

- [01] Susmita Das and J.K. Satapathy, “An overview of various critical factors influencing the Bayesian equaliser’s performance”, *to be submitted to IEEE Transactions on Signal Processing*.
- [02] Susmita Das and J.K. Satapathy, “A new approach for selection of key design parameters for an optimal symbol-by-symbol equaliser”, *to be submitted to IEEE Transactions on Communications*.
- [03] Susmita Das and J.K. Satapathy, “Hierarchical knowledge reinforced neural network configurations for designing efficient adaptive equalisers”, *to be submitted to IEEE Transactions on Neural Networks*.
- [04] Susmita Das and J.K. Satapathy, “Application of orthogonal basis function expansion technique in a neural network paradigm for equalisation of communication channels”, *to be submitted to IEEE Transactions on Circuits and Systems-II : Analog and Digital Signal Processing*.

- [05] Susmita Das and J.K. Satapathy, "Development of transform domain based hybrid neural equalisers for improved BER performance", *to be submitted to IEEE Transactions on Signal Processing*.
- [06] Susmita Das and J.K. Satapathy, "Adaptive equalisation of communication channels using a novel concept of cascaded structures on ANN framework", *to be submitted to IEE Proceedings on Vision, Image and Signal Processing*.
- [07] Susmita Das and J.K. Satapathy, "High performance equaliser design in a neural network domain using an adaptive sigmoidal activation function based on fuzzy logic approach", *to be submitted to IEE Proceedings on Communication*.