Wavelet Analysis and Artificial Intelligence for Diffuse Indoor Optical Wireless Communications

Robert James Dickenson

A thesis submitted in partial fulfilment
of the requirements of the
University of Northumbria at Newcastle
for the degree of
Doctor of Philosophy

Research undertaken in the
School of Computing, Engineering and Information Sciences

June 2007

Wavelet Analysis and Artificial Intelligence for Diffuse Indoor Optical Wireless Communications

Robert James Dickenson

PhD

2007

Abstract

This thesis investigates the use of Wavelet Analysis and Artificial Intelligence as elements of a diffuse indoor infrared (IR) optical wireless communications receiver. The work presented employs Matlab™ enabled simulations to explore the effects of inter symbol interference (ISI) and fluorescent light interference on a receiver using these techniques. The results are compared with those obtained from traditional receiver architectures.

IR devices have been commonplace in most households as remote control handsets for domestic entertainment equipment for many years. More recently, IR communication systems have been deployed in mobile phones, laptop computers and computer peripheral devices largely for the purpose of short range point-to-point data transfers. Since the late 1970's there has been consistent interest and research in the use of the IR part of the spectrum for short-range Wireless Local Area Networks (WLAN). IR offers a number of potential advantages over radio frequency systems such as unregulated and re-usable bandwidth, inherent security, resistance to multipath fading and the availability of mass produced, low cost emitters and detectors. However, significant problems still persist to impede the widespread and popular deployment of IR enabled LANs. The work presented in this thesis focuses on the use of the largely software enabled techniques of Wavelet Analysis and Artificial Intelligence (Wavelet-AI) as novel alternatives to mitigating the difficulties associated with diffuse indoor IR communication systems.

Indoor IR wireless links usually have to operate in the presence of intense noise generated by ambient light sources. The source can be natural sunlight from doors and windows, or artificial light from incandescent and fluorescent fittings. In addition to contributing to the generation of shot noise, artificial light sources can also impose a periodic interference signal that can significantly impair the performance of an optical link. Electrical high pass filtering is a typical mitigating technique that is effective at reducing the interference signal. Unfortunately it also introduces a performance degrading phenomenon known as baseline wander. Using well established interferer models the results of original Wavelet-AI inclusive simulations are presented and compared with those of typical receiver architectures with and without electrical high pass filtering. The performance of Wavelet-AI based receiver was found to be superior to traditional unfiltered receiver architecture in the presence of artificial light interference. At low to medium data rates the Wavelet-AI receiver was also found to outperform all but one case of the traditional receiver architecture employing electrical high pass filtering. The results of baseline wander simulations with On-and-Off keying (OOK) modulation and the Wavelet-AI receiver architectures is presented and shows that the Wavelet-AI architecture is far more tolerant to baseline wander.

In diffuse or non-directed links multipath propagation induced ISI can impose a significant performance penalty for data rates above approximately 10 Mb/s. Typical compensation techniques include the use of equalisers such as the zero-forcing equaliser (ZFE), the minimum mean square equaliser (MMSE) and the decision feedback equaliser (DFE), usually implemented as digital filters. The results of original Wavelet-AI inclusive simulations are presented and compared with those of typical receiver architectures with and without filtering and equalisers. In all cases the simulation results show that the Wavelet-AI receiver architecture performance is superior to the non-equalised traditional receiver. The Wavelet-AI receiver results show a very similar performance to the equalised traditional OOK receiver and the equalised Level 4 Pulse Position Modulation (4-PPM) receiver. However, between a 2 dB and 4 dB optical power penalty was incurred for the 8-PPM Wavelet-AI case. This result may not reflect the true potential performance of the system as time dictated that limited effort was expended on the 8-PPM case. In all instances, further development could potentially enhance the performance of the Wavelet-AI system beyond that revealed by simulations undertaken in this work.

Contents

List of Figures

List of Tables

Glossary of Abbreviations

AC	Alternating current
ADC	Analogue to digital converter
ANN	Artificial neural network
APD	Avalanche photodiode
AWGN	Additive white Gaussian noise
BER	Bit error rate
CDMA	Code division multiple access
CWT	Continuous wavelet transform
DAPPM	Differential amplitude pulse position modulation
DC	Direct current
DFE	Decision feedback equaliser
DFT	Discrete Fourier transform
DH-PIM	Dual header pulse interval modulation
DPIM	Digital pulse interval modulation
DPPM	Differential pulse position modulation
DSP	Digital signal processing
DSSS	Direct sequence spread spectrum
DTRIC	Dielectric totally internally reflecting concentrator
DWT	Discrete wavelet transform
EPM	Edge position modulation
ETSI	European telecommunication standards institute
FBI	Federal bureau of investigation
FFT	Fast Fourier transform
FIR	Finite impulse response
FOV	Field of view
FPGA	Field programmable gate array
GPS	Global positioning system
HPF	High pass filter
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IM/DD	Intensity modulation with direct detection
IR	Infrared
IRED	Infrared emitting diode
IrDA	Infrared Data Association
ISI	Inter-symbol interference

ISM	Industrial, scientific and medical
JPEG	Joint picture experts group
LAN	Local area network
LD	Laser diode
LED	Light-emitting diode
LOS	Line of sight
MAP	Maximum a posteriori
MIMO	Multiple input multiple output
MMSE	Minimum mean square equaliser
MLSD	Maximum likelihood sequence detector
MPPM	Multiple pulse position modulation
MRA	Multi-resolution analysis
MSE	Mean square error
NRZ	Non return to zero
OMIMO	Optical multiple input multiple output
OOK	On-off keying
OOK-NRZ	On-off keying using non return to zero signalling
OOK-RZ	On-off keying using return to zero signalling
PC	Personal computer
PDA	Personal digital assistant
PIN	Positive-intrinsic-negative
PPM	Pulse position modulation
PPM(MAP)	PPM using a maximum a posteriori detector
PPM(TH)	PPM using a threshold detector
PSD	Power spectral density
RADAR	Radio aircraft detection and recognition
RF	Radio frequency
RMS	Root mean square
RZ	Return to zero
SDR	Software defined radio
SNR	Signal to noise ratio
STFT	Short term Fourier transform
SWAP	Size weight and power
TDMA	Time division multiple access
TH	Threshold
Wavelet-AI	Wavelet and Artificial Intelligence
WBSD	Wavelet binary symbol detector
WSSD	Wavelet soft slot detector

WSD	Wavelet slope detector
WTD	Wavelet threshold detector
WLAN	Wireless local area network
WPAN	Wireless personal area network
U-NII	Unlicensed national information infrastructure
ZFE	Zero forcing equaliser

Glossary of Symbols

A	Pulse amplitude
a	Constant related to RMS delay spread
$\boldsymbol{a}$	Filter coefficient array
A_1	Constant which relates low-frequency interference amplitude with I_B
A_2	Constants which relates high-frequency interference amplitude with I_B
a_n	Fourier coefficients or filter coefficients
b_n	Filter coefficients
B	Bandwidth
b_n	Fourier coefficients
$C(f)$	Channel response
C_Ψ	Admissibility constant
$d(k)$	MMSE desired value
D_{mean}	Mean delay of channel
D_{RMS}	Channel RMS delay spread
D_T	Normalised delay spread
e	Base of natural logarithms
$e(k)$	MMSE error value
E	Expected value
f	Frequency
$f(t)$	Function of time
f_c	High-pass filter cut-on frequency
f_{high}	Electronic ballast switching frequency
F_a	Pseudo-frequency
F_c	Wavelet centre frequency
$F(\omega)$	Fourier transform
$G(f)$	Amplitude spectrum
$G_e(f)$	ZFE response
$g(t)$	Impulse response of high-pass filter
$g_{out}(t)$	Output of high-pass filter
H	Height
$H_r(f)$	Normalised gain response
$H(z)$	Transfer function
$h(t)$	Impulse response of channel
I_B	DC photocurrent generated by background radiation
j	Complex number notation

L	Order of PPM
M	Total number of bits over a 20 ms interval
$m_{fl}(t)$	Interference photocurrent from fluorescent lamp
$m_{high}(t)$	High-frequency component of $m_{fl}(t)$
m_k	Fluorescent light interference signal sample
$m_{low}(t)$	Low-frequency component of $m_{fl}(t)$
$n(t)$	Shot noise
N_0	Noise power spectral density (single-sided)
$\mathbf{p}$	Cross correlation vector
$p(t)$	Impulse response of transmitter filter
P_{avg}	Average optical power
$P_{e,bit,OOK}$	Probability of bit error for OOK
P_{max}	Maximum optical transmit power
P_n	Band limited white noise
$P(f)$	Fourier transform of pulse shape
q	Electron charge
$\boldsymbol{q}$	ZFE output array
$Q(\ )$	Marcum's Q-function
R	Photodetector responsivity
$\mathbf{R}$	Correlation matrix
$r(t)$	Impulse response of receiver filter
R_b	Bit rate
RC	Filter time constant
s	Scale or seconds
$S_{c,PPM}(f)$	Continuous component of electrical PSD of PPM
$S_{d,PPM}(f)$	Discrete component of electrical PSD of PPM
$S_{OOK\text{-}NRZ}(f)$	Electrical PSD of OOK-NRZ
$S_{OOK\text{-}RZ(\gamma=0.5)}(f)$	Electrical PSD of OOK-RZ with $\gamma = 0.5$
$S_{PPM}(f)$	Electrical PSD of PPM
t	Time
T	Pulse duration or sample interval
T_b	Bit duration
T_s	Slot duration
T_{symb}	Symbol duration
$u(t)$	Unit step function
$\mathbf{w}$	MMSE filter coefficient matrix
$\mathbf{w}_{opt}$	MMSE optimum filter coefficients
w_n	MMSE filter coefficients

$x(i)$ or $x(k)$	Signal samples
$x(t)$	Intensity or power of optical source or a function of time
$\mathbf{x}$	MMSE sample matrix
$X(k)$	Discrete Fourier transform values
$\boldsymbol{X}$	Sample point matrix
$y(i)$	Output sample (post processing)
y_k	Output of digital filter
$y(k)$	Output of MMSE
$y(t)$	Photocurrent
$\delta(\,)$	Dirac delta function
γ	Pulse duty cycle
λ	Wavelength
φ_i	Phase of odd harmonics of 50 Hz
ϕ_i	Phase of even harmonics of 50 Hz
θ_j	Phase of switching frequency harmonics
Φ_i	Amplitude of odd harmonics of 50 Hz
Ψ_i	Amplitude of even harmonics of 50 Hz
Ψ	Mother wavelet
Γ_j	Amplitude of switching frequency harmonics
ω	Radians per second or short term Fourier transform window
∞	Infinity
Ω	First harmonic frequency
τ	Translation
Δ	Sampling period

Acknowledgements

The work presented in this thesis was started with Optical Communications Research Group within the School of Engineering at Sheffield Hallam University. The work continued with the Optical Communications Research Group within the School of Computing, Engineering and Information Sciences at Northumbria University when my Director of Studies, Professor Z. Ghassemlooy was appointed Associate Dean of Research for the School. The work was undertaken on a part-time basis with some time being allowed for study by my employers. The work was also largely funded by my employers and I am grateful for their support and for them allowing me to take the opportunity to engage in such work.

I would like to thank my director of studies, Professor Z. Ghassemlooy, for his advice, guidance, support and constant enthusiasm throughout this work. I would also like to thank him for proof reading this thesis, and for giving me the opportunity to undertake my studies in the first place.

I would also like to thank those people I have met along the way, colleagues and staff at both universities and numerous academics at seminars that have taken the time to listen and talk to me.

Finally, I would like to say a special thank you to my wife, Tina, for her constant support, patience, understanding and encouragement not just through the duration of this work, but also during my previous academic endeavours through the years.

Declaration

I declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work.

Robert James Dickenson

June 2007

Chapter 1

Introduction

Today's society is becoming increasingly dependent on wireless connectivity with widespread use of numerous, disparate, but continuously converging technologies. WLANs are becoming commonplace in the home for sharing computer access, Internet connections and linking wireless enabled devices. Wireless routers and wireless network cards predominantly use the radio frequency (RF) IEEE 802.11 [1] standards, with 802.11n [6] currently in development. At the time of writing the most commonly deployed standards are IEEE 802.11b [2] and IEEE 802.11g [3], supporting maximum data rates of 11 Mb/s and 54 Mb/s respectively at a frequency of 2.4 GHz in the Industrial Scientific and Medical (ISM) band. Unfortunately this band is subject to loss of performance due to local interferers in the same frequency range, such as microwave ovens, cordless telephones and Bluetooth devices.

In contrast to IEEE 802.11, the European Telecommunications Standard Institutes' (ETSI) High Performance Radio LAN (HIPERLAN) [4] standard operates in the 5 GHz band and does not suffer from the same interferer limitations [5]. The HIPERLAN/2 standard supports higher data rates and introduces further features, however, at the time of writing it has largely failed in terms of market penetration compared to IEEE 802.11 enabled devices.

In January 2006 the IEEE announced the formation of a task group to draft the IEEE 802.11n standard to increase the throughput of the IEEE 802.11 wireless LAN standard [6]. The standard offers a projected typical data rate of 200 Mb/s with theoretical rates up to 540 Mb/s utilising both the ISM and 5 GHz Unlicensed National Information Infrastructure (U-NII) bands. The standard achieves its high speed by using multiple data streams and multiple input / multiple output (MIMO) antennas [7]. The standard will work with both 20 MHz and 40 MHz channels and will be backward compatible with previous standards. However, the standard is not due for approval until September 2007 although some companies are releasing products based on IEEE 802.11n drafts.

Bluetooth is a wireless personal area network (WPAN) based on the IEEE 802.15.1 standard operating in the ISM band that has achieved widespread domestic penetration, with applications being found in the personal computer and mobile device market. Arguably, Bluetooth devices have seen most success in the mobile market where such interfaces are implemented on mobile phones, car radios, Personal Digital Assistants (PDAs) and Global Positioning System (GPS) based navigation devices. Bluetooth can be thought of as a cable replacement system with a relatively short range and an initial data rate of 1 Mb/s; however, recent improvements to the standard have increased this figure to 3 Mb/s [1]. The use of Bluetooth is therefore limited to relatively low bandwidth applications such as real time voice or non-time constrained data transfers.

An emerging WPAN standard is ZigBee, championed by the ZigBee alliance [8], an industrial working group developing standardised application software on top of the IEEE 802.15.4 wireless standard [9]. The system operates in the 868 MHz, 915 MHz and 2.4 GHz bands with peak data rates of 20 kb/s, 40 kb/s and 250 kb/s respectfully with a range of 10 to 75 metres. The protocol is aimed largely at home and business automation

A still growing area of wireless technology is that of cellular based mobile phones, migrating from dedicated voice through 2nd generation (2G) Global System for Mobile Communication (GSM) standards (GSM 850, GSM 900, E-GSM, R-GSM, GSM 1800, GSM 1900) to dedicated to integrated voice and data packet based 3rd generation (3G) systems. Numerous phone models incorporate WPAN technologies, particularly Bluetooth and IR capabilities for device-to-device communications.

There is a growing market in converged devices incorporating the functionality of a mobile phone and PDA in one package. These products employ a whole range of protocols including WLAN and WPAN for home networking and point-to-point connectivity respectfully.

Whilst the above would indicate near total dominance of wireless technologies by radio frequency (RF) techniques and associated protocols, interest and deployment of IR capabilities is substantial. Indeed, IR technologies span wide area network (WAN), WLAN and WPAN applications with by far the largest market penetration being

WPAN products. Arguably, aside from the ubiquitous home entertainment appliance remote control, the largest group of deployed IR capable devices is mobile phones that use the IR medium for point-to-point data transfers. Shipments of Infrared Data Association (IrDA) transceivers reached over 297 million units in 2005 an increase of 17% compared with 2004 [10]. Mobile phones represented over 80% of the total IrDA market in 2005.

Given the volumes and relentless introduction of new models, the mobile phone and PDA market continues to attract new IR development. A recently introduced protocol for mobile phones being IRSimpleShot (IrSS) [11], a 16 Mb/s protocol aimed at transferring photographic images from camera phones to printers and printer kiosks. Another standard is Infrared Financial Messaging (IrFM) [12], targeting point of sales applications using IrFM enabled mobile phones and PDAs.

IR WLAN products are inherently limited in their use by the very nature of light propagation. Light cannot pass through doors and walls therefore its use is restricted to the room of origin; a centrally sited transmitter and receiver cannot practically service an entire house or range of rooms. In certain circumstances this limitation becomes an advantage, particularly where security issues are of paramount importance.

Military applications require enhanced security; U.S., U.K. and NATO installations are tested to TEMPEST [13] standards. TEMPEST is a codename for a set of standards limiting electromagnetic radiation from military equipment. The standard is available in US and NATO formats and is widely adopted, including by the U.K. MoD. The exact details of TEMPEST testing are still classified, however short excerpts of the American standard, NSTISSAM TEMPEST/1-92 are publically available. Battlefield network security and the ability for ad-hoc networking continues to be a security issue to military and government organisations throughout the world. Ad-hoc networking ability is one of the primary problems the U.K. MoD wants to solve and is looking at providing sponsorship for innovators in the field [14].

IR technologies can also be used for networking medical equipment, both in equipment to equipment and equipment to network modes without the associated conducted interference or susceptibility risk of RF technologies; an operating room scenario is discussed by Hagihira *et al* in [15]. Mobile medical applications may also have restrictive power requirements that prohibit the use of RF. Shin *et al* [16] propose a low power hybrid IR/RF system for infant incubators that addresses this problem; whilst Jivkov and Jivkova [17] propose a very low power compact Electroencephalograph (EEG) data transfer link.

Recently, white light emitting diode (LED) lighting fixtures for domestic use have become popular, both from a decorative and power efficiency point of view. The use of such lighting for optical communications has received recent research interest [18, 19]. The use of domestic wiring for establishing a link between a central server and the LED fitments within individual rooms could potentially overcome the restrictive nature of optical transmissions as discussed in [20].

Whilst the IR medium may currently be overshadowed by the plethora of RF devices deployed for wireless networking in the home and office, it still has significant potential, perhaps even through hybrid Wired/RF/IR systems. Further to this, the unique properties of the IR medium may also make it suitable for more demanding specialist applications such as military and medical uses that are not as sensitive to the usual commercial constraints.

The wide bandwidth of the IR medium has lead researchers to pursue implementations that can realise the potentially high data rates available. However, there are significant obstacles to this, even when the transmitter and receiver are located within the same room. Ambient light interference from natural and artificial light sources are potentially much stronger than the transmitted signal. In addition to this, a signal can take many paths from the transmitter to the receiver resulting in ISI. Both of these phenomena currently limit achievable data rates. Currently, IR receivers employ detection techniques largely developed for RF systems that may not provide the best solution for the diffuse IR environment.

This work focuses on a novel, largely software deployable technique as an alternative and possible improvement to traditional RF inspired methods for the detection of signals transmitted over the unique indoor IR channel. The technique is potentially deployable on any optical wireless system and has the inherent ability to be adapted for virtually any optical channel and numerous modulation schemes. The technique can also be paired with coding schemes, such as convolutional coding, to achieve higher data rates. It is arguable that the processing power required for such a technique is beyond the capability of readily available hardware; however, the relentless increase of commercially available processing power shows no sign of abating. In addition, military appetite for digital signal processing (DSP) is growing for such applications as radio aircraft detection and recognition (RADAR) and software defined radio (SDR), pursuing size weight and power (SWAP) reductions for embedded systems [21]. Field programmable gate arrays (FPGAs) are now becoming a popular device for the deployment of DSP algorithms and are well suited to for very high speed multiply and accumulate functions [21]. The advances in technology made by space and military research inevitably yield improvements in the commercial sector that will ultimately allow more demanding software algorithms to become realisable.

1.1 Research Objectives

The primary aim of this research is to explore the possibility of using wavelet analysis and artificial intelligence as the fundamental elements in a receiver architecture for detecting signals transmitted over a diffuse indoor infrared channel. The research aims to determine whether such architectures have the potential to yield an improvement in performance in comparison to traditionally employed receivers. In order to achieve this a number of research objectives have been outlined as follows:

- Review the fundamental properties of indoor optical wireless communication systems, understand the characteristics of the diffuse indoor IR channel and understand the constraints imposed by the channel on receiver architecture.

- Review the concepts involved in signal feature identification, particularly with respect to wavelet analysis. Examine its uses, including its possible employment and potential advantages when deployed as part of an IR receiver detecting signals transmitted over a diffuse channel.

- Review the concept of artificial intelligence. Examine its uses, including its possible partnering with Wavelet Analysis techniques and potential advantages when deployed as part of an IR receiver detecting signals transmitted over a diffuse IR channel.

- Examine the performance of a Wavelet-AI receiver architecture in the presence of artificial light interference emanating from a fluorescent light driven by electronic ballast. Compare the performance of the Wavelet-AI receiver with that of a traditional receiver architecture for both OOK and PPM transmissions over a diffuse IR channel.

- Examine the performance of a Wavelet-AI receiver architecture in the presence of multipath induced ISI. Compare the performance of the Wavelet-AI receiver with that of a traditional receiver architecture for both OOK and PPM transmissions over a diffuse channel.

1.2 Organisation of Thesis

This thesis is divided into eight chapters. Following this introduction, chapter 2 provides a general introduction of indoor optical wireless links. In this chapter the properties of the indoor infrared channel is described along with link configurations and some of their associated advantages and disadvantages. Channel characterisation and channel modelling are reviewed followed by a brief discussion on optical component selection. The chapter includes a review of performance limiting interfering light sources and work done to model these to date. The chapter also includes a short review of modulation techniques regarded as suitable for infrared systems, with an examination of properties that are considered important when considering modulation schemes for such systems.

Chapter 3 considers signal feature identification. The chapter commences by considering the general concepts and traditional techniques associated with the subject. Wavelet analysis is introduced and in particular the continuous wavelet transform (CWT) which is compared and contrasted to usual Fourier techniques. The chapter continues with the introduction of the discrete wavelet transform (DWT) and wavelet packet analysis. Finally a simulation of direct detection using wavelets analysis on a multipath IR channel is introduced and discussed.

Chapter 4 provides a review of general artificial intelligence techniques including artificial neurons, feed-forward back-propagation networks, radial basis function networks and recurrent networks. A scheme for the direct detection of OOK return to zeros (RZ) transmissions over a diffuse infrared channel is considered, providing results of exploratory simulations. The detection scheme is extended to include filtering and later, down sampling and the simulation results compared. The chapter goes on to consider a proposed Wavelet-AI receiver architecture and progresses with architecture simplification to reduce computational burden.

In Chapter 5 the error performance of both traditional and Wavelet-AI receiver architectures is examined in the presence of noise emanating from artificial light sources. Evidently, fluorescent lamps driven by electronic ballast are potentially the most degrading; therefore a model of this type is introduced and used for simulations. The effect of artificial light interference without electrical high pass filtering is considered for both OOK and PPM and the results of simulations presented. The chapter introduces the Wavelet-AI Binary Symbol Detector (WBSD) and the Wavelet-AI Soft Slot Detector (WSSD) for PPM transmissions and presents simulation results. The effect of baseline wander using OOK modulation for both the Wavelet-AI and traditional receiver architecture is discussed and simulation results presented. Finally, the chapter considers the effect of fluorescent light interference with electrical high pass filtering for traditional OOK and PPM systems and compares the simulation results with those obtained from the Wavelet-AI architectures.

Chapter 6 investigates the error performance of both traditional and Wavelet-AI receiver architectures when receiving a signal subjected to multipath distortion. The effects of ISI on a simple threshold receiver structure are presented. Following this, common mitigation techniques used to overcome multipath induced inter-symbol interference are considered including: filtering, coding, equalisers and diversity. A channel model for the simulations is described and the simulation results of the unequalised performance of both traditional and Wavelet-AI receivers for OOK and PPM modulation is presented. Finally the simulated equalised performance of the traditional receiver structures is presented and contrasted with the Wavelet-AI receiver.

In chapter 7 concluding comments are made and major elements of work within the thesis are outlined.

Finally, chapter 8 discusses areas where research can be continued.

1.3 Original Contributions

During the course of this work the author has:

1. Developed novel receiver architectures incorporating Wavelet Analysis and artificial intelligence as an alternative to traditional techniques for detecting signals transmitted over a diffuse IR channel as detailed in chapters 4 and 5. Schemes for OOK RZ and PPM were developed including the WBSD and WSSD receivers of chapters 5 and 6.

2. Developed Matlab™ simulation models for the Wavelet-AI receiver architectures to enable performance comparisons with traditional receiver models also developed using Matlab™ as detailed in chapters 3 through 6 and listed in the appendices.

3. Investigated the error performance of the Wavelet-AI receiver architecture in the presence of fluorescent light interference driven by high frequency ballast. Investigated the ability of the Wavelet-AI architecture to reject the effects of baseline wander in comparison to traditional receiver types. Compared the performance of the Wavelet-AI receiver with that of traditional receiver models under the same conditions both with and without mitigation filtering, all as detailed in chapter 5.

4. Investigated the performance of the Wavelet-AI receiver architecture in the presence of mutli-path induced ISI caused by the diffuse infrared channel. Compared the performance of the Wavelet-AI receiver with that of traditional

receiver models under the same conditions both with and without equalisation techniques, all as detailed in chapter 6.

1.4 List of Publications

The work presented in this thesis has resulted in the following publications, which are listed in reverse chronological order:

R. J. Dickenson & Z. Ghassemlooy, "A Software Based Receiver Kernel for Diffuse Indoor IR Communication Systems," *Presentation to: The Optics Of Free Space Optical Communications Meeting, Institute of Physics*, London, December 2006.

R. J. Dickenson, Z. Ghassemlooy, "Wavelet-AI equalization and detection for indoor diffuse infrared wireless systems," *International Journal of Communication Systems*, vol. 18, iss. 3, pp. 247-266, April 2005.

R. J. Dickenson, Z. Ghassemlooy, "BER performance of 166 Mbit s OOK diffuse indoor IR link employing wavelets and neural networks," *Electronics Letters*, vol. 40, iss. 12, June 2004.

R. J. Dickenson and Z. Ghassemlooy, "An Experimental Receiver Architecture For Diffuse Indoor Infrared Environments," *Optical Wireless Communications Technical Symposium*, Warwick University, September 2003.

Robert J. Dickenson and Z. Ghassemlooy, "A Feature Extraction and Pattern Recognition Receiver Employing Artificial Intelligence For Signal Detection In Diffuse Optical Wireless Communications," *IEEE Wireless Communications Magazine*, vol. 10, iss. 2, April 2003.

R. J. Dickenson and Z. Ghassemlooy, "Application Of Wavelet Analysis Analysis And Artificial Intelligence For Signal Detection In Optical Wireless Communications," *Proceedings of the 3rd Postgraduate Symposium on The Convergeance of Telecommunications, Networking and Broadcasting*, John Moores University, Liverpool, pp. 63-67, June 2002.

Chapter 2

Indoor Optical Wireless Links

2.1 Introduction

This chapter aims to provide an overview of indoor optical wireless links, particularly with respect to the aspects that are covered later in the thesis. In section 2.2 the general properties of wireless links are reviewed and the limitations on system design highlighted. The major indoor IR configurations are considered in section 2.3 and constraints of each type are briefly reviewed. In addition to noise, channel characteristics impose severe limitations on link performance and much research effort is expended in finding ways to increase the speed of data transfer, this topic is discussed in section 2.4. Optical component selection must be considered when designing a particular type of link; however, component selection does not play a major part in this work and therefore section 2.5 provides a brief overview. Section 2.6 discusses the effects of both natural and artificial light sources. Modulation schemes are discussed in section 2.7 as they have a significant influence on link performance in both a positive and negative sense depending on potentially conflicting requirements imposed by channel characteristics and IR devices. Finally, section 2.8 summarises the chapter.

2.2 Properties of Indoor Optical Wireless Links

Intensity modulation and direct detection (IM/DD) is the de-facto method of implementing indoor optical wireless systems principally due to cost and complexity [22, 23, 24]. The drive current of an optical source is modulated which in turn varies the transmitted optical power $x(t)$. The receiver employs a photodetector which generates a photocurrent $y(t)$ that is proportional to the instantaneous optical power incident upon it. An IM/DD based optical wireless system has an equivalent baseband model that hides the high frequency nature of the optical carrier [23]. The model is shown in Fig. 2.1 [22], where R is the photodetector responsivity, $h(t)$ is the baseband channel impulse response and $n(t)$ is the signal independent shot noise, modelled throughout the work described here as additive white Gaussian noise (AWGN).

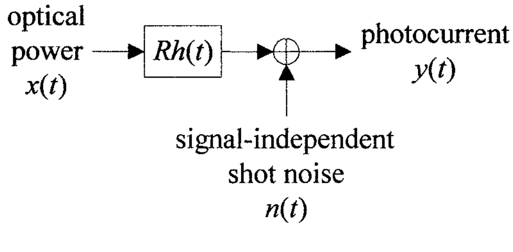


Fig. 2.1: Equivalent baseband model of an optical wireless system employing IM/DD [22].

Optical wireless links are subject to the effects of multipath propagation in the same way as RF systems and these effects are more pronounced for non-directional transmitter receiver pairs. This type of link can suffer form severe multipath induced performance penalties and is the subject of a large part of this work. Multipath propagation causes the electric field to suffer from severe amplitude fades on the scale of a wavelength. The detector would experience multipath fading if the detector size (surface area) was proportional to one wavelength or less. Fortunately, IR wireless receivers have detectors which are typically millions of square wavelengths. In addition, the total photocurrent generated is proportional to the integral of the optical power over the entire photodetector surface; this provides an inherent spatial diversity as shown in Fig. 2.2. [22].

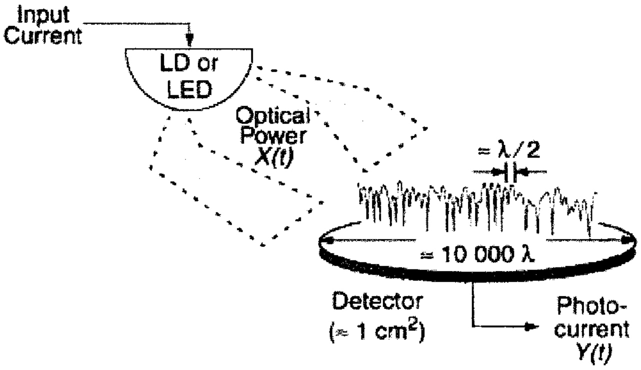


Fig. 2.2: Spatial diversity [22].

Although indoor IR links do not suffer from the effects of multipath fading, they do suffer from the effects of dispersion which manifests itself in a practical sense as ISI. Dispersion is modelled as a linear baseband channel impulse response $h(t)$. The channel characteristic of an IR link is fixed for a given position of transmitter, receiver and intervening reflecting objects. The channel characteristic only changes when these components are moved by distances in the order of centimetres [22]. Due to high bit rates, and the relatively slow movement of objects and people within a room, the channel will vary only on the time scale of many bit periods, and may therefore be considered as quasi-static [25].

Wireless IR links usually operate in domestic or office environments and are subject to intense ambient light, emanating from both natural (sunlight) and artificial (fluorescent and incandescent) light sources. The average combined power of this background radiation generates a DC photocurrent I_B in the photodetector, producing shot noise $n(t)$ that has a single-sided power spectral density N_o given as [4]:

$$N_o = 2qI_B, \qquad (2.1)$$

Where q is the electron charge.

Optical filtering can be used to reject light at wavelengths outside of those being used to transmit data, even so the received signal power is much lower than power from ambient light sources; typically 25 dB lower [22]. I_B is therefore usually much larger than the photocurrent generated by the signal and so shot noise can be considered as white Gaussian and independent of the received signal [27]. Evidently, receiver preamplifier noise is the dominant noise source in a diffuse receiver if little ambient light is present, this can also be considered as signal independent and Gaussian [23].

Artificial light sources also contribute a periodic interference to the signal which must be added to $n(t)$. Given the above, the equivalent baseband model of an optical wireless link can be summarised by [23]:

$$y(t) = Rx(t) \otimes h(t) + n(t), \qquad (2.2)$$

Where $y(t)$ is the convolution of the transmitted optical power $x(t)$ with the channel impulse response $h(t)$, scaled by the photodetector responsivity R, plus additive noise

$n(t)$. In practice photodetector responsivity is lower than unity; however, as the subject of this work deals mainly with comparative simulations, R is taken as unity in all later cases considered in this thesis.

There is a major difference between an optical wireless system and its RF contemporary since in equation (2.2), $x(t)$ represents power rather than amplitude. This places two constraints on the transmitted signal. Firstly, $x(t)$ must be non-negative:

$$x(t) \geq 0 . \tag{2.3}$$

Secondly, for the wavelengths of interest, ocular safety requirements limit the average amount of power which may be transmitted and $x(t)$ must not exceed a specified value P_{max} [22]:

$$P_{\max} \geq \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} x(t)\, dt . \tag{2.4}$$

In RF systems the time averaged power of a signal is given by $|x(t)|^2$. These differences have a significant effect on system design. On RF channels the signal to noise ratio (SNR) is proportional to the average received power. On optical channels the SNR is proportional to the square of the average received optical signal power, implying that relatively high optical transmit powers are required and significant path loss cannot be tolerated [23].

Limited average transmission powers indicate that modulation techniques with a high peak-to-mean power ratio would be most suitable for optical wireless links. However, such techniques suggest shorter pulse durations requiring progressively wider bandwidths. When shot noise is dominant however, the SNR is proportional to the photodetector area, favouring the use of large area receivers for single element detectors. Unfortunately, as the detector area increases so does its capacitance, this in turn limits the receiver bandwidth. This conflicts with the increased bandwidth requirement of power efficient modulation schemes and therefore a trade-off exists between these two factors.

2.3 Link Configuration

There are numerous ways that an optical link can be physically configured. These are typically grouped into four system configurations [28]. These configurations are: directed line of sight (LOS), non-directed LOS, diffuse and tracked. However, hybrid variations on these configurations are also possible as shown in Fig. 2.3 [23].

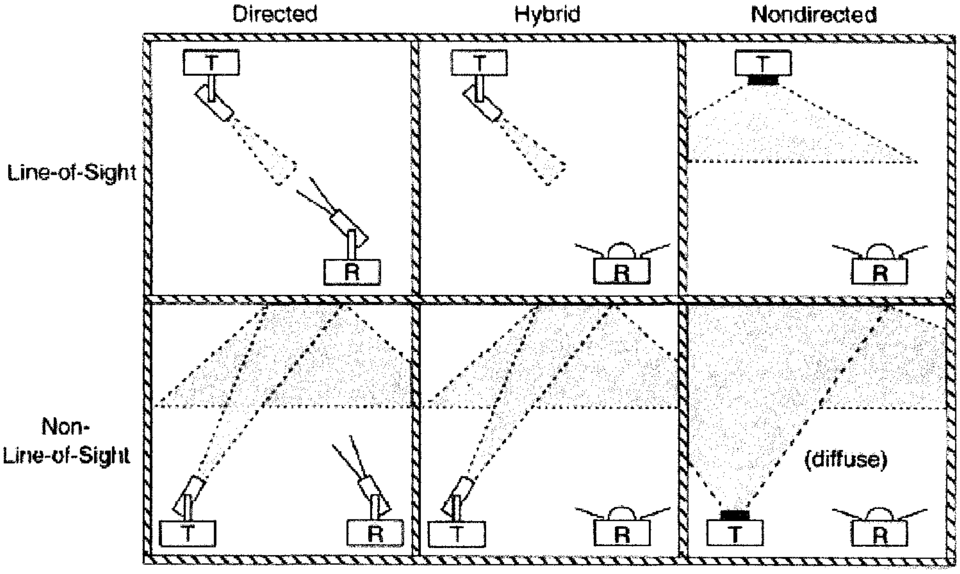


Fig. 2.3: Link configurations [23].

Directed LOS links are power efficient, and do due to their point-to–point configuration do not suffer from the effects of multipath propagation; noise form ambient light sources is also largely rejected. The data rate is therefore limited by free space path loss rather than the effects of multipath dispersion [28, 30]. However, there are disadvantages. Directed LOS links must be accurately 'pointed' or aligned before use and require an uninterrupted LOS path making them susceptible to beam blocking. As an aside, in an outdoor environment they are also subject to the effects of atmospheric conditions causing scintillations which defocus and deflect the beam as shown in Fig. 2.4 [30].

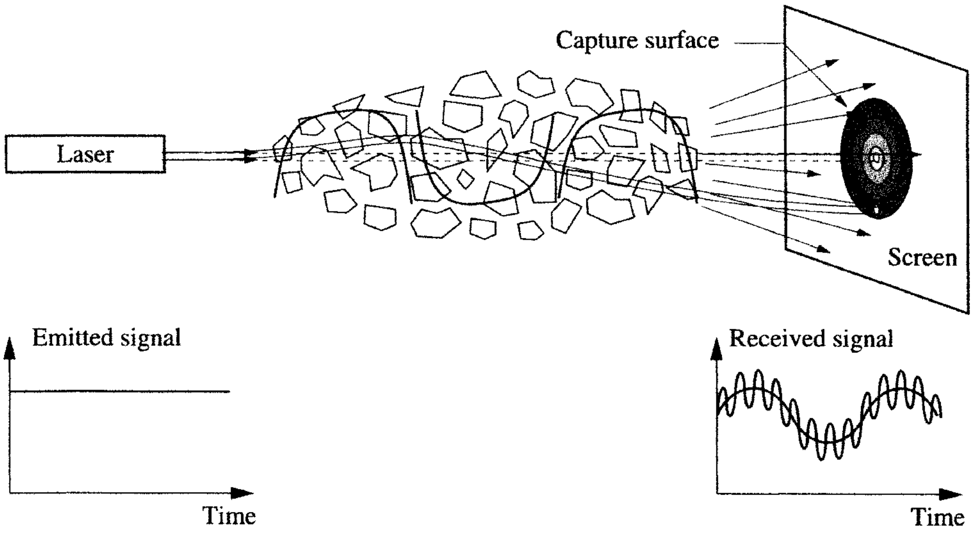


Fig. 2.4: The effects of scintillation [30].

Whilst high data rate directed links can be used for building-to-building communication [31], their most well known uses is in low bit rate, simplex remote control applications largely for domestic entertainment equipment.

Nondirected LOS configurations use wide beam transmitters and wide field of view (FOV) receivers to achieve a broader coverage area without the need to accurately align the devices. This type of link suffers from reduced irradiance for a given transmission power and may suffer from the effects of multipath propagation. As most of the received power is received from the LOS path such links are still prone to being blocked. Nondirected links are suitable for point to multipoint broadcast applications, an example being JVC's VIPSLAN-10 [32]. This product offers IR access points either in ceilings or walls of a room with a cell radius of 10 m, and a data rate of 10 Mb/s shared between all users in the cell.

The diffuse configuration, proposed in [33, 34] typically consists of a transmitter that points directly towards the ceiling emitting a wide IR beam. The receiver has a wide FOV and collects the signal after it has undergone one or more reflections from objects within the room. Such reflections attenuate the signal with typical reflection coefficients being between 0.4 to 0.9 [34]. The received signal can also suffer from severe multipath dispersion. However, from a user's point of view this link architecture is the most convenient as it does not require any alignment of the transmitter or receiver, does not rely on LOS and is almost immune to beam blocking. The diffuse configuration is also flexible and can be employed in both infrastructure and ad-hoc networks [28]. Unfortunately, diffuse links experience high path loss, typically 52 - 70 dB for a horizontal separation of 5 m [35]. These links also suffer from shadowing, where a person or object blocks the main path between transmitter and receiver, further increasing path loss [35].

Performance in diffuse systems can be improved by the use of diversity techniques and enhanced optical design such as the recently proposed dielectric totally internally reflecting concentrator (DTRIC) optical antenna [36]. Such methods can be used to partner other techniques to provide a combined improvement as in [37], where angle diversity is employed in conjunction with code division multiple access (CDMA).

Angle diversity receivers consist of multiple receiver elements effectively pointing in different directions either because of their physical orientation or due to imaging concentrators [38]. The photocurrent generated by each receiver element can be processed in a number of ways, including the use of digital Wavelet-AI techniques. This type of receiver can reduce the effects of ambient noise and multipath distortion due to the fact that these unwanted contributions are usually received from different directions to that of the desired signal [42]. The performance of such schemes have been analysed in [43 – 49]. Another diversity technique is to employ a multi-beam transmitter, or quasi-diffuse transmitter. This type of transmitter employs multiple narrow beams pointing in different directions [38 - 41]. The performance of some of these schemes is presented in [38, 42, 50 – 54].

Tracked systems potentially offer high data rates and high power efficiency, BT conducted experiments that achieved a data rate of 1 Gb/s using mechanical steerable optics [57]. Unfortunately, such systems are expensive to realise and solid state systems have been proposed which are conceptually similar to angle diversity techniques. More recently, a single channel imaging receiver has been proposed [55], which could be described as a hybrid tracked and angle diversity system. Optical multiple input multiple output (OMIMO) systems [56] improve channel quality due to spatial diversity, or system speed using multiplexing. Diversity techniques can also potentially implement space division multiplexing providing the advantage that multiple users can communicate without loss of per-user capacity, since each user would be located in a different cell [10].

2.4 Channel Characterisation

Characterisation of the indoor optical wireless channel must be considered if the best performing link design is to be implemented and it has been a subject of considerable interest [25, 35, 58 - 70]. There are two main factors when considering power penalties associated with the indoor channel, these being path loss and multipath dispersion [25, 58]. For directed configurations path loss due to reflections do not represent a major concern and the loss can be calculated from information on the transmitter beam divergence, receiver size and transmitter receiver separation distance. However, for diffuse configurations the optical path loss is much more difficult to predict as it depends on factors such as room size, reflectivity of surfaces, position and orientation of transmitter and receiver, plus other contributing factors and many different modelling strategies have been pursued.

In addition to reflective path loss, higher data rates incur a power penalty due multipath dispersion. Although the optical power associated with two or more reflections is much smaller than the first reflection or the LOS component, the higher order reflections arrive much later and therefore cannot be ignored when considering high data rate transmissions due to the induced ISI. Bary *et al* presented recursive methods for evaluating the impulse response of the indoor channel with Lambertian reflectors [59]. Lomba *et al* published work that considers the approximation of emitter pattern, propagation environment and receiver pattern. The authors introduced 'time delay agglutination and time and space indexed tables' to improve the efficiency of simulation for the impulse response of the optical channel [60]. In [61] an iterative site based modelling technique is discussed as a method that can efficiently account for multiple reflections of any order. The model can also take account of interior features such as furniture and room partitions. Jungnickel *et al* [62] introduce a model of the indoor IR channel where the signal is modelled as the combination of LOS and diffuse components. The authors use Ulbricht's integrating sphere for calculating diffuse contributions and indicate that the LOS contribution is dependent upon the Rician factor [62]. Carruthers and Kahn [25, 58] developed a ceiling bounce model for the diffuse channel and suggested that realistic multipath IR channels can be characterised by only two parameters. Namely the optical path loss and the root mean square (RMS) delay spread. The model is based on two stages; initially only reflections from the ceiling of an infinitely large room are considered, corrections are then made with respect to the position of the transmitter receiver pair within the room. This model is considered more fully in later chapters and is the basis of all the simulated multipath channel characteristics within this work.

2.5 Optical Component Selection

The majority of components for IR applications have a wavelength between 780 and 950 nm. They are typically low cost devices used in mobile phone, computer peripheral and home entertainment remote control devices. Electromagnetic radiation in this band can potentially cause damage to the human eye and safety precautions must be taken; however, the latest edition of the IEC standard, IEC 60825-1 Amd. 2, effective January 2001 [71] allowed fairly large increases in exposure limits for LEDs and infrared emitting devices (IREDs). It is difficult for a normal IRED to exceed these limits under normal operating conditions. In addition IrDA standards for the physical layer also limit the amount of power that a device can transmit and still be compliant. Whilst ocular safety considerations have diminished with the later IEC standards, compliance with

IrDA [11] for interoperability, power handling of IR devices and power consumption for portable appliances are still factors when considering suitable transmission power for a particular link design.

Laser diodes can be used as an optical source for indoor applications and offer a number of advantages over LEDs including wide modulation bandwidth and narrow spectral width; however, they require more complex drive circuitry and they are point source devices and must be diffused in some way to make them eye safe [72]. The two options for photodetector construction are the positive-intrinsic-negative (PIN) photodiode and the avalanche photodiode (APD). Unfortunately, APDs degrade link performance in high ambient noise conditions and it is the PIN photodiode that is generally employed in such systems [22]. Commercial IREDs can be produced from a range of III-V compounds, however, for commonly used wavelengths (800 nm – 1000 nm) the materials are limited to GaAs and mixed crystal GaAlAs [73].

Indoor optical wireless receivers usually employ optical filtering in planar or hemispherical forms. These can also be configured in such a way as to provide band pass effects. The choice of filter has an influence on the FOV and careful consideration should be given to these effects on the performance of non-directed links. Many commercial receiver components are delivered complete with some form of concentrator and filter in place suitable for their target application. In [22] analysis of hemispherical concentrators and optical filters are considered for an OOK system operating at 100 Mb/s

2.6 Ambient Light Sources

The degrading effects of ambient light sources cannot be neglected in optical link design. The main sources of ambient light are sunlight, incandescent lamps and fluorescent lamps. The optical spectra of these are shown in Fig. 2.5 [23].

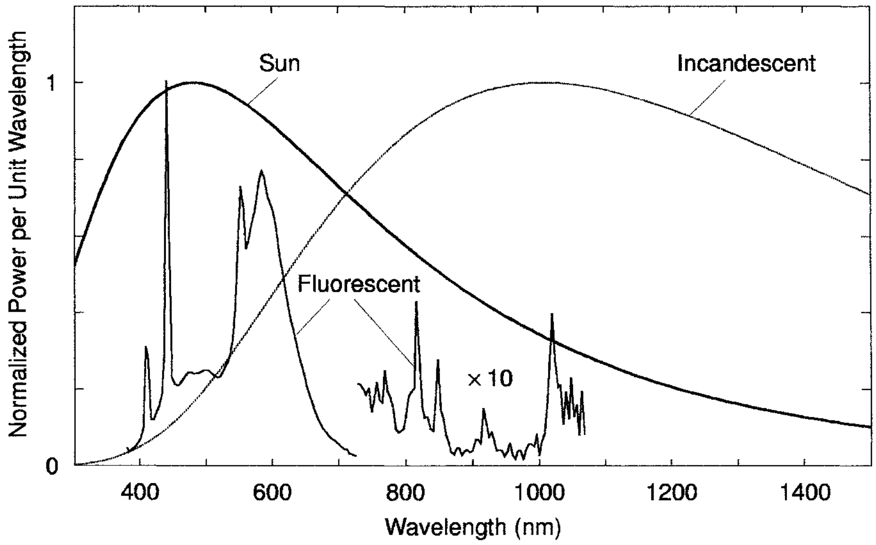


Fig. 2.5: Optical spectra [23].

The spectra in the figure have been scaled to have equal maximum value, and the longer wavelength of the fluorescent lamp spectrum has been amplified by a factor of 10 to make it more clearly visible. Sunlight is typically much stronger than the other two sources and represents an unmodulated source of ambient light with wide spectral width and maximum power spectral density located around 500 nm.

Moreira *et al* carried conducted extensive measurements on a variety of ambient light sources and produced associated models to describe the interference signals [74, 75]. In addition to the characterisation of ambient light sources significant effort has been expended analysing the effect of ambient light on link performance [26, 76, 77, 78-84].

Artificial light sources are modulated either by the mains frequency or in the case of some fluorescent lamps by a high frequency switching signal. Incandescent lamps have a maximum power spectral density around 1 μm and produce an interference signal consisting of a near perfect sinusoid at a frequency of 100 Hz [74]. Fluorescent lamps driven by conventional ballast produce an interference signal represented by a distorted 100 Hz sinusoid, and the electrical spectrum contains harmonics into tens of kHz. Fluorescent lamps driven by electronic ballasts typically have switching frequencies in the 20 – 40 kHz range. The detected electrical spectrum contains harmonics of the mains and switching frequencies [72]. Harmonics of the switching frequency can extend into the MHz range and therefore present a much more serious impairment to optical wireless receivers [76]. Aspects of fluorescent light interference driven by electronic ballast are investigated further later in this thesis.

2.7 Modulation Schemes

The transmission power employed in indoor IR configurations is limited by numerous factors including, ocular safety, physical device limitations and power consumption. Limitations on power transmission favour modulation schemes with high peak to mean power characteristics. Such schemes by their very nature have short duration pulses with short rise and fall times that lead to wider spectral bandwidth requirements. The bandwidth of high data rate systems is limited due to the capacitance constraints of large area photodiodes and therefore a compromise between power and bandwidth requirements must be pursued.

In addition to this, mutlipath channel induced ISI is more severe for shorter pulse duration schemes due to the short rise and fall times required. Baseband or near baseband schemes like OOK are therefore more tolerant to the effects of the multipath channel. In contrast high-pass filtering is commonly employed to reduce the effects of periodic inference from artificial light sources. Such filtering introduces a phenomenon known as baseline wander, which is more severe for modulation schemes with a significant amount of power located near DC [85]. These conflicting constraints have resulted in a great deal of research effort being expended on power efficient modulation schemes with minimum bandwidth requirement, or with resistance in some way to the adverse effects of the channel characteristics [26, 72, 82, 83, 85 - 91, 94 - 106].

Farrell and Wong [88] propose complementary inverse keying, a binary direct sequence spread spectrum (DSSS) technique that reduces the affect of ISI to achieve higher data rates. Sethaskaset and Gulliver [89] suggest a differential amplitude pulse position modulation (DAPPM) scheme that offers advantages over PPM in terms of bandwidth efficiency and peak-to-average power ratio. Edge position modulation (EPM) [91] is a further scheme that reduces the bandwidth requirement; this is achieved primarily by using the rising edge of a pulse to convey data. Time is divided into discrete slots greater than the rise time and jitter of a pulse; however, the pulse width can be wider than one time slot thus allowing more information to be transmitted than for a comparable PPM scheme.

In [87] digital pulse interval modulation (DPIM) is presented, in this technique the information is represented by the number of empty slots between pulses, potentially allowing higher data rates and improvements in power efficiency compared to OOK and PPM. Aldibbiat *et al* [86] propose dual-header pulse interval (DH-PIM), this variation on PIM reduces the number of 'empty' slots, and therefore symbol length, by introducing a second pulse at the start of the information symbol. The technique offers a trade-off between the lower bandwidth requirement of the longer pulse and the subsequent higher average optical power requirement. At higher bit rates the scheme is both bandwidth and power efficient compared to PPM. Unlike the fixed symbol length of PPM, DPIM and DH-PIM are anisochronous, but offer symbol synchronisation due to the pulse always being at the start of the symbol.

An additional strategy to finding the best compromise for the modulation scheme is to find a receiver architecture that is tolerant to some or a combination of the limiting aspects of channel characteristics and receiver electronics, this thesis explores one of these avenues. Further to this, modulation schemes such as those proposed in [86, 87, 91] are potential candidates for use with wavelet enabled receiver architectures due to the ability of wavelet analysis to potentially isolate temporal discontinuities. Given this, the Wavelet-AI receiver may yield further improvements in performance for these schemes. However, OOK and PPM are well understood and well cited in diffuse indoor IR research and all further work in this thesis relates to these modulation types.

OOK is by far the most simple and well known modulation technique, which is also one of the easiest to implement in an optical IM/DD system. A 'one' is represented by transmitting a pulse of constant optical power for a duration of up to one bit, and a 'zero' by transmitting nothing during the bit period. Both RZ and 'non-return to zero' (NRZ) schemes can be employed. OOK NRZ has a pulse that spans the entire bit period (T_b), whilst OOK RZ has a pulse of less than T_b. For an average optical power (P_{avg}), the transmitted waveforms for OOK NRZ and OOK RZ with a duty cycle (γ) of 0.5 are shown in Figs. 2.6(a) and (b), respectively [72].

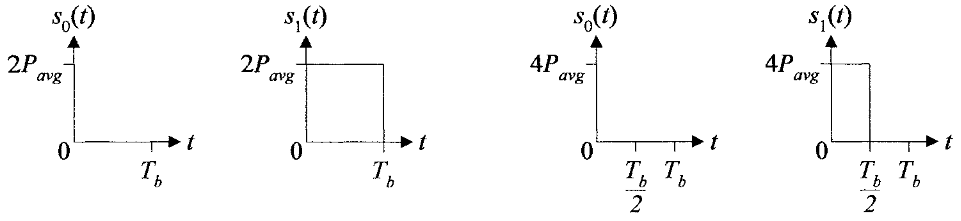


Fig.2.6: Transmitted waveforms for OOK: (a) NRZ and (b) RZ (γ = 0.5).

The simplicity of OOK has precipitated its use in commercial optical wireless systems including IrDA links operating below 4 Mb/s [93]. The IrDA standard employs RZ 'inverted' signalling, where a pulse signifies a zero rather than a one.

The Power spectral density (PSD) of an OOK-NRZ waveform is given by [92]:

$$S_{OOK-NRZ}(f)=(P_{avg}R)^2 T_b\left(\frac{\sin \pi f T_b}{\pi f T_b}\right)^2\left[1+\frac{1}{T_b}\delta(f)\right], \tag{2.5}$$

Where R is the photodetector responsivity and δ is the Dirac delta function.

For OOK-RZ (γ = 0.5) the PSD is given by [92]:

$$S_{OOK-RZ(\gamma=0.5)}(f)=(P_{avg}R)^2 T_b\left(\frac{\sin(\pi f T_b/2)}{\pi f T_b/2}\right)^2\left[1+\frac{1}{T_b}\sum_{n=-\infty}^{\infty}\delta\left(f-\frac{n}{T_b}\right)\right]. \tag{2.6}$$

The PSDs of both OOK modulation types are shown in Fig 2.7 [72]. The power axis has been normalised to the average electrical power multiplied by the bit duration, and the frequency axis is normalised to the bit rate $R_b = 1/T_b$.

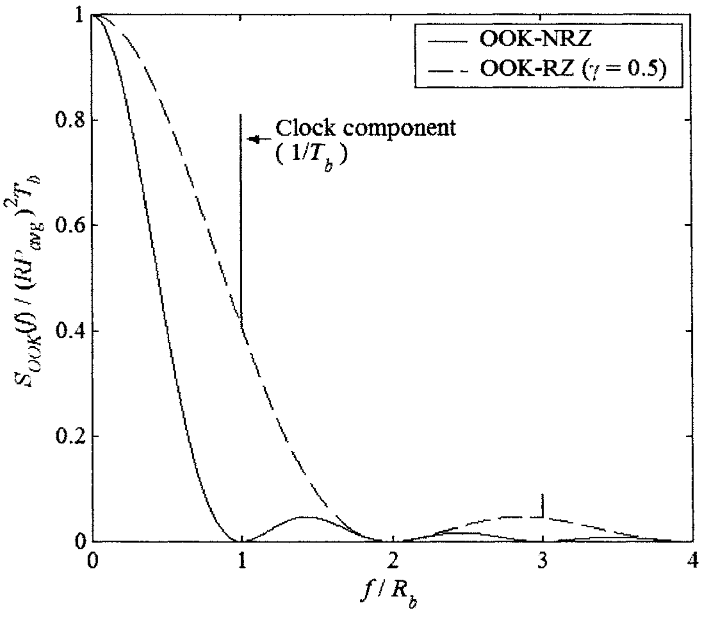


Fig. 2.7: PSD of OOK-NRZ and OOK-RZ (γ = 0.5)

For baseband techniques, the bandwidth requirement is generally defined as the span from DC to the first null in the PSD of the transmitted signal [72]. As can be seen, OOK-RZ (γ = 0.5) has twice the bandwidth requirement of OOK-NRZ. Both have significant power content at or near DC. Such characteristics incur the penalties imposed by baseline wander when electrical high pass filtering is employed to reduce the effects of artificial light interference as discussed more fully in Chapter 5.

PPM is an orthogonal modulation technique [90] that improves the power efficiency of OOK at the expense of bandwidth and complexity [83, 86]. In this scheme each Log$_2L$ data bits is mapped onto one of the L possible symbols. Usually, the notation L-PPM is used to indicate the order. Every symbol consists of a pulse of constant power occupying one slot, along with $L-1$ empty slots. The slot duration (T_s) is related to the bit duration by [72]:

$$T_s = T_b \log_2 L/L. \tag{2.7}$$

Assuming an average optical power of P_{avg} the transmitted waveforms for 4-PPM are shown in Fig. 2.8.

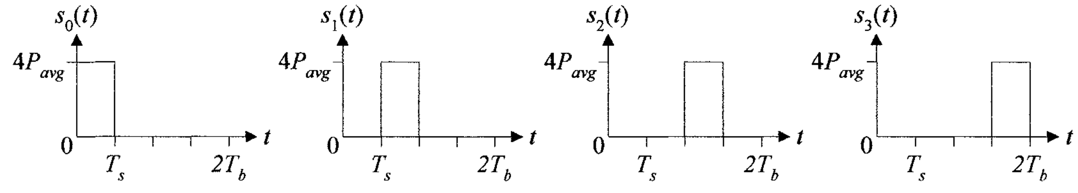


Fig. 2.8: Transmitted waveforms for 4-PPM

In comparison to OOK, PPM incurs the additional complexity required to perform both slot and symbol synchronisation to enable demodulation of the signal. However, due to its power efficiency PPM is a popular modulation scheme and has been adopted in the IrDA physical standard for data links operating at 4 Mb/s [93]. The detected electrical power spectrum of L-PPM is given as [94]:

$$S_{PPM}(f) = |P(f)|^2 \left[S_{c,PPM}(f) + S_{d,PPM}(f) \right], \tag{2.8}$$

Where $S_{c,PPM}(f)$ and $S_{d,PPM}(f)$ are the continuous and discrete components respectively, which are given as:

$$S_{c,PPM}(f) = \frac{1}{T_{symb}} \left[\left(1 - \frac{1}{L}\right) + \frac{2}{L} \sum_{k=1}^{L-1} \left(\frac{k}{L} - 1\right) \cos\left(\frac{k 2\pi f T_{symb}}{L}\right) \right], \tag{2.9}$$

$$S_{d,PPM}(f) = \frac{2\pi}{T_{symb}^{\ 2}} \sum_{k=-\infty}^{\infty} \delta\left(f - \frac{kL}{T_{symb}}\right), \tag{2.10}$$

$P(f)$ is the Fourier transform of the pulse shape and T_{symb} is the symbol duration, which is given as $T_{symb} = T_b \log_2 L$.

The PSD of PPM for $L = 4$, 8 and 16 is shown in Fig. 2.9 [72]. The three curves were constructed using the same average optical power, using rectangular pulse shapes occupying the full slot duration. The power axis is normalised to the average electrical power multiplied by the duration. The frequency axis is normalised to the bit rate R_b.

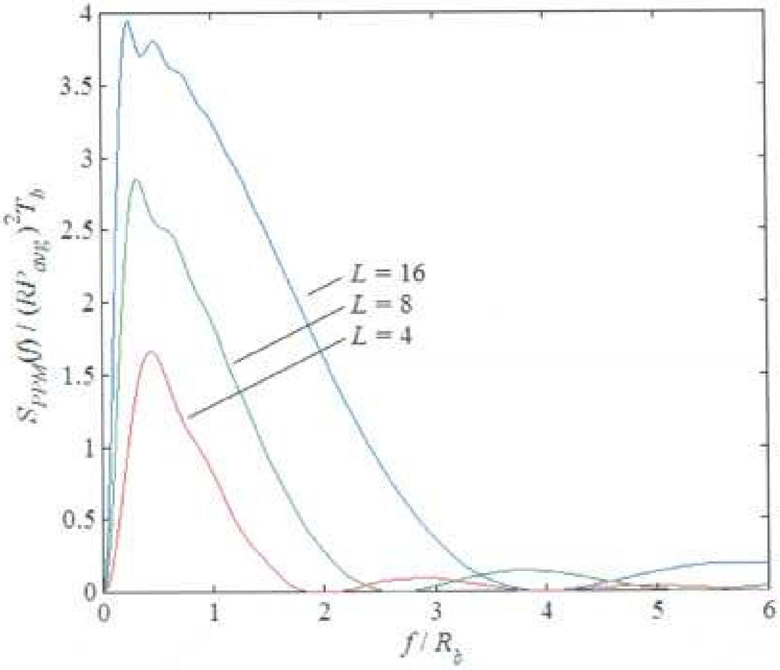


Fig. 2.9: PSD of PPM for L=4, 8 and 16 [72].

From Fig. 2.9 it is easily seen that unlike OOK, the PSD of PPM falls to zero at DC for all values of L. This phenomenon provides an increased resistance to baseline wander over DC schemes and allows the use of higher cut on frequencies when mitigating high pass filtering is employed to combat artificial light interference. However, higher orders of L require increased bandwidth over OOK and therefore exacerbate the effects of the multipath channel.

2.8 Summary

Numerous ways exist to configure an indoor optical link, each offering advantages and disadvantages that must be considered with reference to the deployed application. Limitations imposed by path loss and dispersion are largely dependent on link configuration. LOS and directed links suffer limited effects from path loss and multipath propagation; however, the mobility of such links is limited without employing complex and expensive tracking systems. Nondirected, or diffuse links are far less complex and subsequently cheaper, offering the user a greater degree of flexibility in terms of positioning transmitter and receiver equipment. Unfortunately such links have to overcome high path loss and the effects of multipath propagation.

Indoor optical wireless links usually operate in the presence of intense ambient light. Such light contributes to shot noise generation, and with artificial light sources a periodic interference signal as well. However, fluorescent lights driven by electronic ballasts potentially impose the most severe degrading effects. The extent to which ambient light sources effect system performance again depends on link configuration. Directional and LOS links can be configured to reject a considerable amount of background radiation, whilst nondirected links by their very nature are more susceptible.

Regardless of the link configuration, the indoor optical channel is unique, combining the filtered Gaussian noise characteristics of conventional wire based channels with the IM/DD constraints of fibre-optic systems [22]. The diffuse configuration potentially provides the most flexible, convenient and cost effective end user solution whilst suffering the most degrading channel characteristics. The diffuse channel therefore presents the greatest challenge to implementing reliable high-speed communications. The remainder of this thesis concentrates solely on the diffuse channel.

Chapter 3

Signal Feature Identification

3.1 Introduction

This chapter aims to provide a brief introduction to the topic of wavelet analysis in its major forms and its suitability as a candidate for inclusion into an alternative diffuse IR receiver based around feature extraction and classification. In section 3.2 an outline of traditional Fourier analysis is presented and its limitations in feature extraction highlighted. In section 3.3 wavelet analysis in its major forms is introduced and contrasted with Fourier analysis highlighting its advantages for the type of application required in this work. Sections 3.2 and section 3.3 are inextricably linked as their properties are contrasted and compared in the text. Section 3.4 discusses model architecture and simulation of a basic OOK RZ threshold receiver based on wavelet coefficient values, the results of the simulations are compared with those obtained from a comparable traditional threshold receiver under the same conditions. Finally, section 3.5 summarises and concludes the chapter.

3.2 Fourier Analysis

The majority of frequency analysis techniques can be traced back to the fundamental work undertaken by Jean-Baptiste-Joseph Fourier during the nineteenth century and his resulting 'Theorie analytique de la chaleur' (The mathematical theory of heat). Indeed, Fourier analysis remains an essential mathematical tool in many branches of science and engineering. However, Fourier based frequency analysis has inherent fundamental limitations in terms of its ability to temporally isolate localised events, a problem which is largely solved in wavelet analysis.

The Fourier series involves representing the periodic signal of interest *f(t)*, with a period *T*, as a series of sine waves and cosine waves as in (3.1). The Fourier coefficients a_o, a_n, b_n etc., scale the amplitude of the trigonometric function

The trigonometric Fourier series is given by [107] as:

$$f(t) = \frac{1}{2}a_o + \sum_{n}^{\infty}(a_n \cos n\omega_o t + b_n \sin n\omega_0 t) \tag{3.1}$$

Where: $\omega_0 = 2\pi/T$.

The Fourier series therefore uses sine and cosine waves as basis functions, and it is possible to synthesize any periodic signal from the trigonometric Fourier series. The

Fourier series can be further written in harmonics and complex forms. The first harmonic is commonly called the fundamental component as it has the same period as the function $f(t)$. A periodic signal can therefore be represented as a series of sine waves with different amplitudes and frequencies. This is commonly referred to as the signals spectrum.

The Fourier series approach must be modified for non-periodic signals, an important example of this case being a single rectangular pulse. The Fourier transform is used for this purpose and is given by [107]:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}\,d\omega \tag{3.2}$$

It is possible to transform the frequency domain into the time domain by using the inverse Fourier transform [107]:

In the majority of practical applications spectral analysis is carried out by a digital computer rather than by analogue processing and the Fourier transform must again be modified to function with the discrete sample points obtained by analogue to digital conversion (ADC) circuitry. Given a sample sequence $\{x(nT)\} = x(0) + x(T)+,\ldots,+x[(N-1)T]$, where n is the sample number and T is the sample interval, the Discrete Fourier Transform (DFT) of $x(nT)$ is defined as a sequence of complex values $\{X(k\Omega)\} = X(0), X(\Omega), \ldots, X[N-1]\Omega$ in the frequency domain. The DFT values $X(k)$ are given by [108]:

$$X(k) = F_D[x(nt)] = \sum_{n=0}^{N-1} x(nT)e^{-jk\Omega nT} \tag{3.3}$$

Where $k = 0, 1, \ldots, N - 1$. and $\Omega = 2\pi/nT$

Again it is also possible to covert from the frequency domain to the time domain by using the inverse DFT.

In applications such as spectral analysis, the transform is generally used as a one-way process from the time to the frequency domain. In addition, these transformations are performed by a variety of general purpose or dedicated computer hardware. In these circumstances the decimation in time fast Fourier transform, usually now abbreviated to FFT, is employed as it considerably reduces the number of calculations required and makes real time, or near real time analysis of dynamic signals possible. The first decimation in time algorithm was due to Cooley and Tukey (1965) [108]. The mathematical derivation of the FFT algorithm can be found it texts such as [108]. Whilst there are many other transforms available to the mathematician and engineer, it is the Fourier transform and its derivatives that find the most use, especially in electronic engineering.

The inherent temporal limitations of Fourier analysis can be understood by considering 2 signals of interest. Fig. 3.1 shows a signal consisting of 3 sinusoids at different frequencies disjointed in time. Fig. 3.2 illustrates a signal consisting of 3 mixed sinusoids of the same frequencies as shown in Fig. 3.1. Their corresponding FFTs generated from them by the use of Matlab™ can be seen in Figs. 3.3 and 3.4, respectfully.

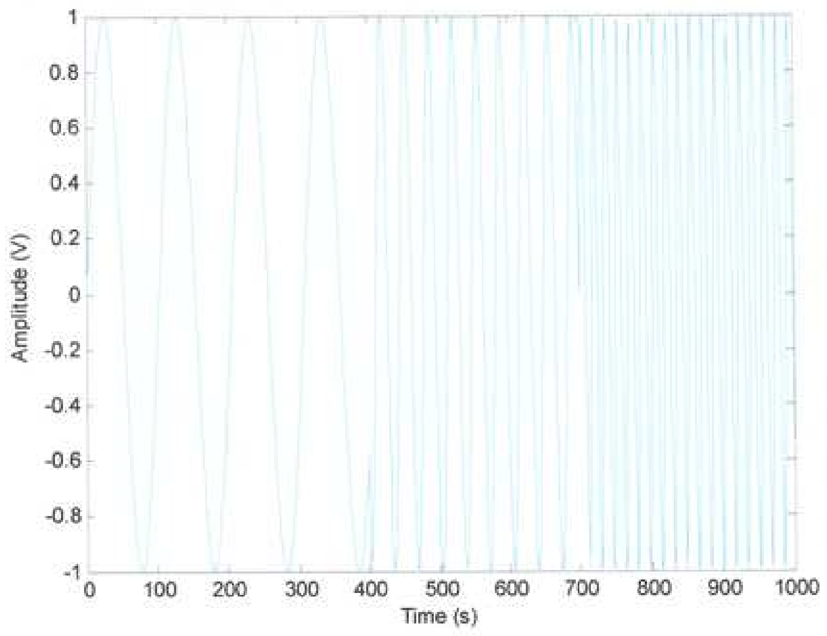


Fig. 3.1: 3 Sinusoids disjointed in time

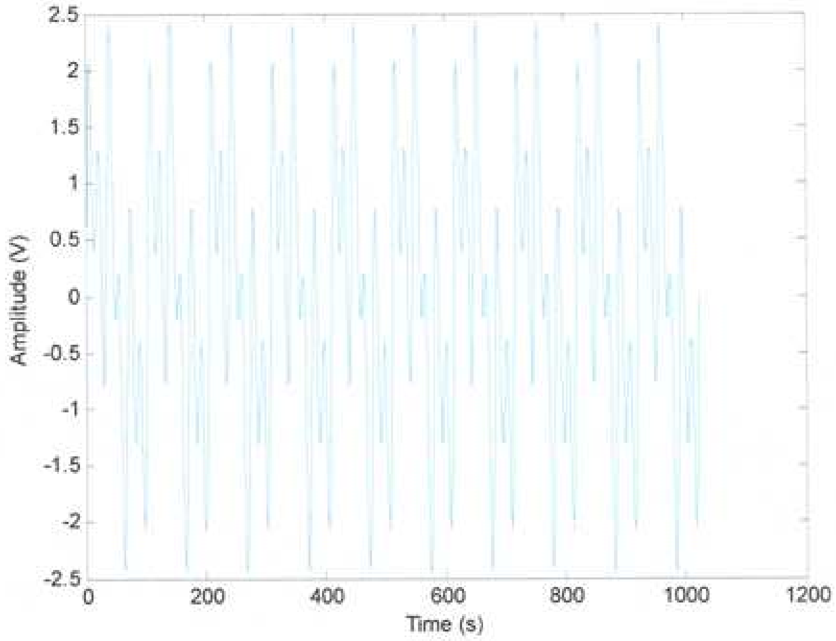


Fig. 3.2: 3 Mixed sinusoids

The frequency content of both signals is entirely identifiable as the 3 peaks at different frequencies. Whilst frequency (and phase) information can be deduced by Fourier methods and their derivatives there is no indication of resolution in time, and from inspection of Figs. 3.3 and 3.4 we would conclude that they were the spectral signatures of the same, or very similar looking signals. From the FFT in this form it is impossible to deduce temporal information such as overall signal length, frequency transition boundaries and signal component length.

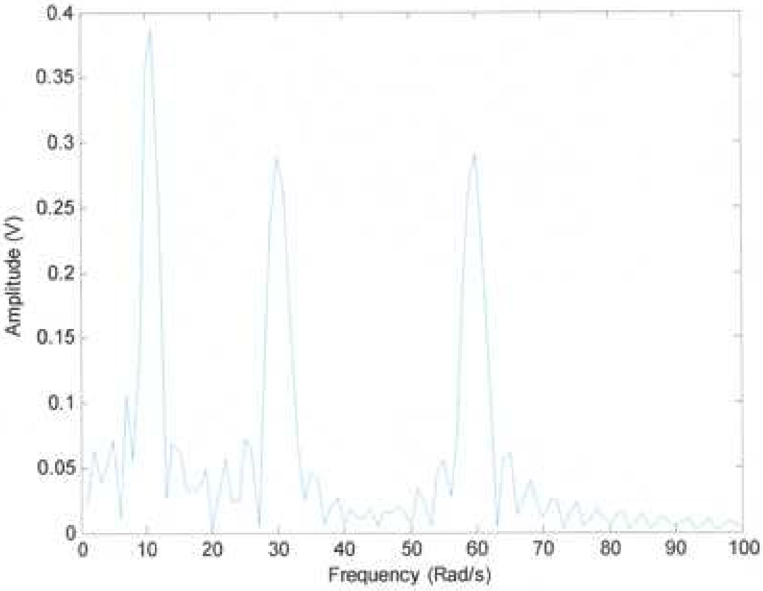


Fig. 3.3: FFT of signal in Fig. 3.1

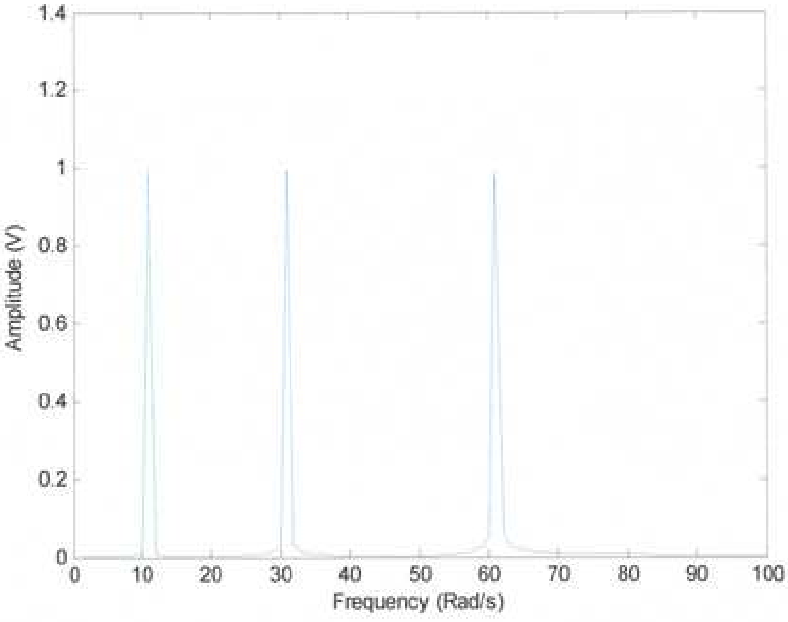


Fig. 3.4: FFT of signal in Fig. 3.2

This situation can be improved by employing the short term Fourier transform (STFT), this derivation of the Fourier transform essentially segments the signal of interest into constant duration discrete time blocks or windows. The STFT is given by [109]:

$$STFT_X^{(\omega)}(t,f) = \int_t [x(t) \bullet \omega^*(t-t')] \bullet e^{-j2\pi ft} dt \tag{3.4}$$

Where $x(t)$ is the signal of interest and ω is the window function.

The STFT does isolate signal artifacts in time; however, the constant window width ' ω ' proves troublesome in terms of resolution. This relates back to Heisenberg's uncertainty principle, originally applied to the movement and location of particles. In essence it is only possible to determine a band of frequencies during a particular time interval and not an exact frequency at a particular time. A wide window ' ω ' that spanned the time period of the entire signal tends toward the standard Fourier transform case that provides exact frequency resolution with no time information. At the other extreme a small window would provide good time resolution but poor frequency resolution. These issues are insurmountable with the fixed window duration employed by the STFT. Given these limitations we must consider other methods that can resolve frequency in a more scaleable manner.

3.3 Wavelet Analysis

The topic of wavelets is multi-faceted and highly mathematical, and a subject that is arguably dominated by researchers with a pure or applied mathematical background.

Difficulties due to the mathematical complexities of wavelet analysis are alluded to by some tutorials [109] and by the authors of some of the many mathematically dominated texts published on the subject. In addition, further comments are made as to the theoretical details of the subject not being entirely necessary for engineering applications. This is undoubtedly a debatable point of view, however; it is undeniable that due to its complexity and length that a comprehensive review of wavelet theory is beyond the scope and space available for this thesis. Discussion of wavelet theory is by necessity limited to its application in this work and as an overview of concepts to aid understanding and possible future research paths for the advancement of work presented here. Ultimately, wavelet analysis is applied in this work as a tool rather than a research objective in its own right and the topics treatment will reflect this.

Evidently the first recorded mention of what has become to be known as a "wavelet" seems to have been in 1909 in a thesis by Alfred Haar [110]. The concept of wavelets in its present theoretical form was first proposed by Jean Morlet and the team at the Marseille Theoretical Physics Center working under Alex Grossmann in France [110].

Since then a number of notable names have worked on the subject such as Yves Meyer with the main algorithm being credited to work undertaken by Stephane Mallat in 1988 [110]. Since that time, work on the subject has become widespread and covers disciplines too numerous to mention. In later years research became particularly active in the United States with fundamental work being undertaken by Ingrid Daubechies, Ronald Coifman, and Victor Wickerhauser [110]. A comprehensive history of the subject can be found in [112].

The feature rich mapping of signals to wavelet coefficients has resulted in an explosion of papers and potential applications in many fields of science and engineering from stock market applications and human motor behavior to optimization of the JPEG2000 standard [113 – 119]. However, it is possible to generalise these applications into three main areas, these being: data compression, denoising and feature extraction. In some cases it is a subtle combination of these elements that form a particular application. In this work it is arguably the combined abilities of denoising and feature identification that are employed. The identification and denoising properties of wavelets are also raising medical research interests in diagnostic applications such as [120].

One of the most famous applications of wavelet analysis is the compression of fingerprints by the Federal Bureau of Investigation (FBI) Los Alamos National Laboratory in the US [121]. Here, an algorithm referred to as 'Wavelet/Scalar Quantisation' is used to compress 2000 TB (and growing) of images at an average ratio of 15:1, something that JPEG standards of the time could not achieve with a satisfactory level of detail. Ironically, another general-purpose use of wavelet technology for compression is the joint picture expert group's (JPEG) JPEG2000 standard [122]. This standard is aimed at virtually any application that requires image compression from Internet applications, through digital cameras to medical imagery. Research and advances in this field yield new compression algorithms and simplifications such as a near-lossless coder described in [123].

The subject of communications engineering is well subscribed to by researchers with the subject of wavelets being applied in almost every facet of the field. Wavelet packets have recently been used in CDMA systems [124 – 129] and also as the basis of a modulation method [130 – 136] The problem of channel induced distortion, a major difficulty in diffuse IR systems, also receives interest with the publication of wavelet based equalization schemes being presented [137 – 151]. Haykin et al proposed a time division multiple access (TDMA) receiver using a neural network and wavelet analysis section [152].

3.3.1 The Continuous Wavelet Transform

Although the Heisenberg uncertainty principle applies to any transform, multi-resolution analysis (MRA) minimizes its impact as not every spectral component is resolved equally as in the STFT. The CWT is an implementation of MRA and is given by [109]:

$$CWT_x^{\psi}(\tau,s) = \psi_x^{\psi}(\tau,s) = \frac{1}{\sqrt{|s|}} \int x(t)\psi^*\left(\frac{t-\tau}{s}\right)dt \tag{3.5}$$

Where τ and s are the translation and scale parameters respectively, ψ is the wavelet function and x is the signal of interest. The term $|s|^{-0.5}$ normalises the energy content of the transformed signal to the same level for every scale.

Compared to the STFT the most significant difference is that the width of the windowing function ψ is changed for every spectral component. ψ is also known as the 'mother wavelet' as all other 'scaled' wavelets are derived from it. The scaling refers to compression or dilation of the mother wavelet. Large scales relate to low frequency features of the signal $x(t)$ and small scales relate to higher frequency components. However, in the wavelet transform the scale parameter 's' appears in the denominator and the numerical relationship is inversed; lower scale values refer to high frequency features, and high scale values to lower frequency features.

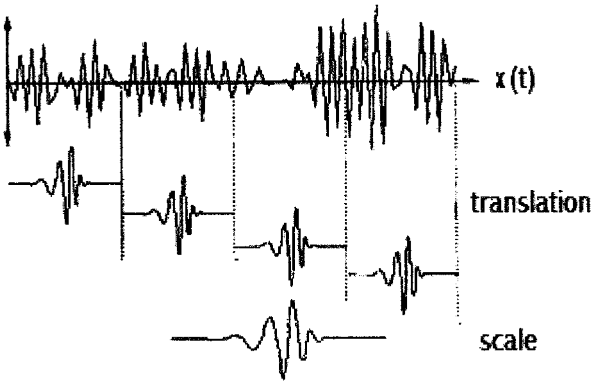


Fig. 3.5: Pictorial representation of the CWT process

The CWT coefficients are generated by multiplying the signal $x(t)$ with ψ at a defined scale and time and then integrating over all time. The wavelet ψ is indexed along the signal by some amount τ and the next coefficient calculated. This is repeated for the entire length in time of the signal $x(t)$ and a set of coefficients for a particular scale *'s'* is produced. The scale is increased by some amount and the process repeated providing a new set of coefficients for scale *'s+1'* as pictorially represented in Fig. 3.5. As a result a coefficient is produced for every scale and every time resulting in a time, scale and amplitude representation of the original signal. Fig. 3.6 shows a 3 dimensional plot of coefficients of the CWT of the signal in Fig. 3.1, and Fig 3.7 for the signal in Fig. 3.2. It is intuitive to see how the time-scale map identifies attributes of the original signal, localising features in time; which is in complete contrast to the FFT's shown in Figs. 3.3 and 3.4. It is this feature rich representation of the signal that makes the CWT and its derivatives to be a popular choice of research and application in signal analysis applications. The time-scale mapping is an essential feature of the majority of the work presented here.

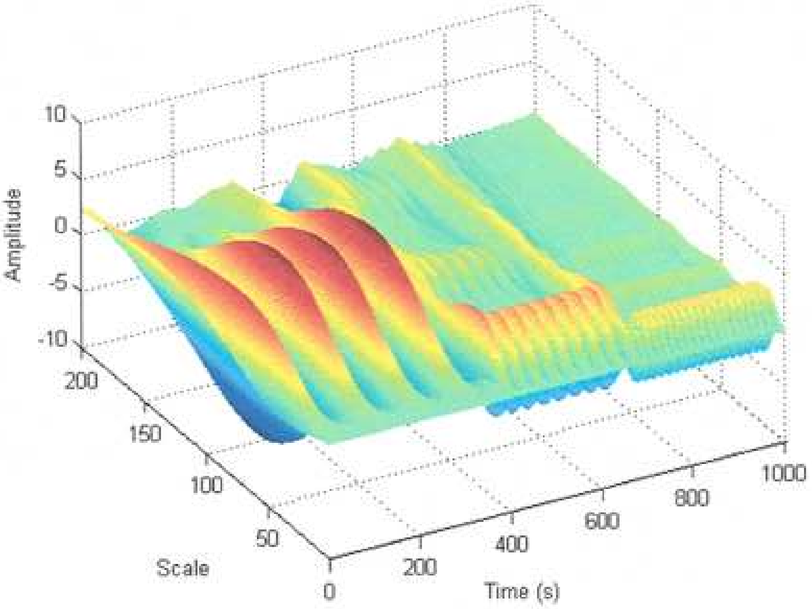


Fig.3.6: CWT of signal in Fig. 3.1

The amplitude of the coefficients at a particular scale relate to how good the correlation or 'fit' is with the wavelet at that particular point in time. The larger the value the better the fit between signal and wavelet; the actual values depend on the type of wavelet used for the decomposition. In later work the coefficient values are only used as a comparative measure by a neural network and our interest in actual minima and maxima is limited. What is more important is the wavelets ability to define attributes of the signal that enable correct detection of a data stream.

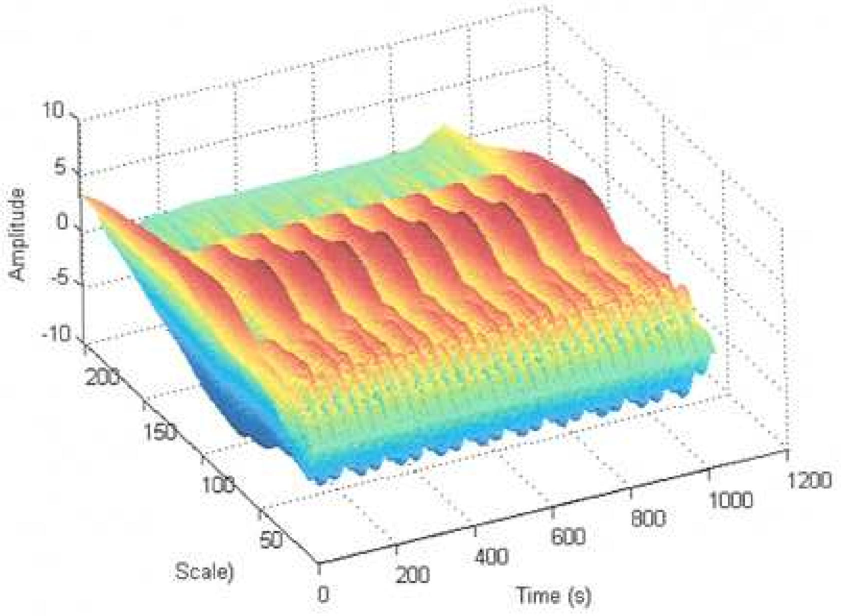


Fig. 3.7: CWT of signal in Fig. 3.2

The mulitresoluton properties of the CWT are pictorially represented in Fig. 3.8a. Whilst the figure looks like an empty grid, it actually represents how the time and frequency plane is divided by the wavelet transform. In the figure the horizontal axis represents time and the vertical axis represents frequency. Each rectangle in the figure represents a value of the wavelet transform. Every box has a non-zero area indicating that an exact point in the time-frequency plane cannot be known. Each box has an identical area occupying an identical amount of the time-frequency plane; however, each area has different dimensions for length (frequency) and width (time). For lower frequencies this equates to better frequency resolution but poor time resolution, whilst higher frequencies have better time but poorer frequency resolution. In contrast the time–frequency map for the STFT, shown in Fig. 3.8b, would show areas of equal heights and widths, however, different STFT windows (equivalent to mother wavelets) would result in different areas up to a point of some lower bound determined by the uncertainty principle. For the CWT, the resolution in time or frequency is also determined by the choice of wavelet.

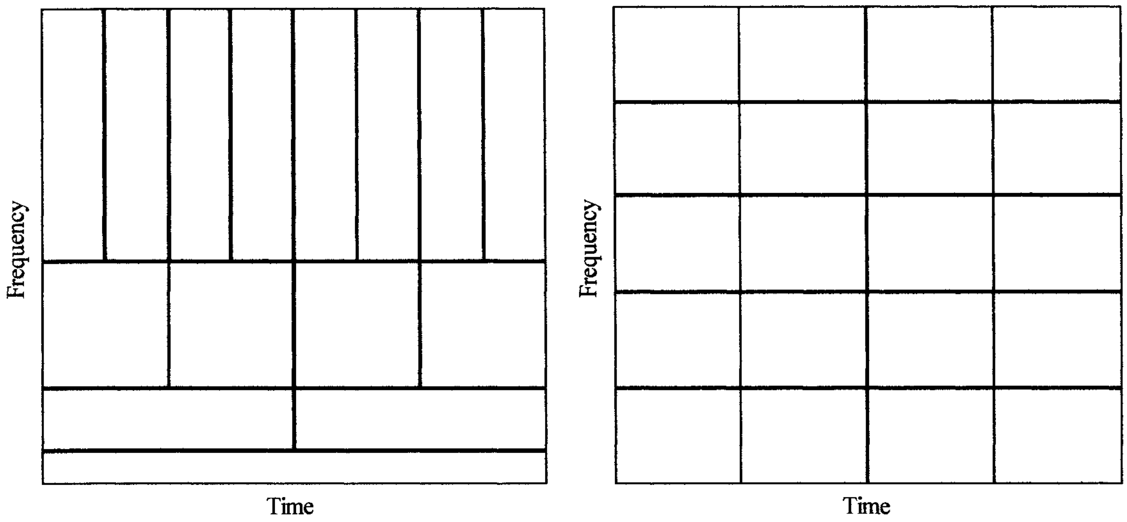


Fig. 3.8a: Time – frequency representation of CWT, and Fig. 3.8b: the STFT

As with the FFT and STFT, CWT analysis of meaningful signals by hand is almost impossible and computers are used for signal processing. In this case a discretised or time sampled version of the CWT is used. Fortunately as the mother wavelet is dilated (higher scales and lower frequencies), the sampling rate can be reduced without affecting the results. If synthesis (reconstruction from wavelet coefficients) is required the Nyquist sampling rate must be observed. In this work the concern is only with decimation of signals for feature extraction purposes and synthesis given by equation (3.6) [109] is not considered.

$$x(t)=\frac{1}{c_{\psi}^{2}}\int_{s}\int_{\tau}\psi_{x}^{\psi}(\tau,s)\frac{1}{s^{2}}\psi\left(\frac{t-\tau}{s}\right)d\tau ds \qquad (3.6)$$

Where C_{ψ} is the admissibility constant that depends on the wavelet used, given by [3]:

$$c_{\psi}=\left\{2\pi\int_{-\infty}^{\infty}\frac{\left|\hat{\psi(\xi)}\right|}{|\xi|}d\xi\right\}^{1/2} <\infty \qquad (3.7)$$

Where $\psi\char`^\xi$ is the Fourier transform $\psi(t)$. Evidently, this condition is satisfied if:

$$\int \psi(t)dt = 0 \tag{3.8}$$

Many wavelets can be found that satisfy this condition.

As intimated, there is a relationship between scale and frequency; however, many texts such as [110] suggest it is better to think in terms of pseudo-frequency. The relationship between scale and frequency is dependent on the centre frequency of the wavelet F_c and the sampling period Δ, and is given by [110]:

$$F_a = \frac{F_c}{s \bullet \Delta} \tag{3.9}$$

Where F_a is the pseudo-frequency corresponding to the scale 's', in Hz. Fig. 3.9 shows the relationship between pseudo-frequency and scale for the 'db2' wavelet shown in Fig. 3.10. The relationship between the 'db2' wavelet and its centre frequency approximation is also shown in Fig. 3.10. From this figure it is intuitive to see how pseudo-frequency relates to the frequency of a sinusoid.

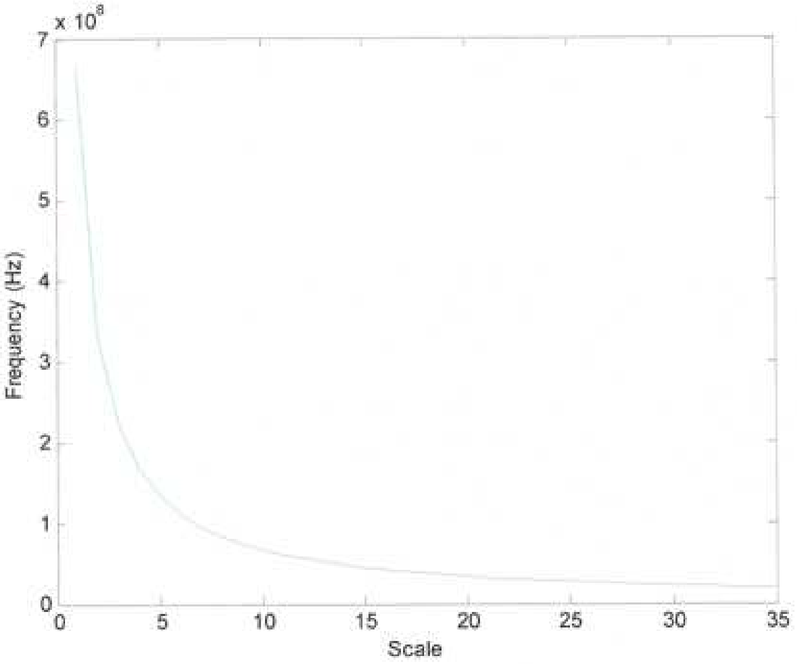


Fig. 3.9: Relationship between frequency and scale for 'db2' wavelet

There are numerous pre-defined wavelets that can be used in signal analysis, or you can design your own. Chapa and Rao have developed an algorithm for designing wavelets to match the signal of interest [153]. Which type of wavelet one uses depends on the application, but it is worthwhile to note that wavelets are more effective at analysing waveforms that are similar to themselves.

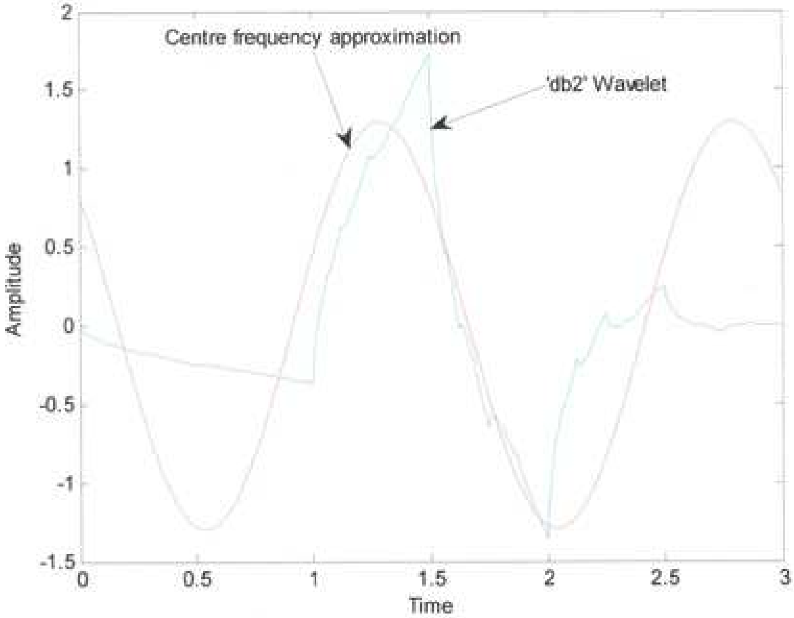


Fig. 3.10: 'db2' wavelet and centre frequency based approximation

Fortunately a number of software packages and tools are available for the manipulation and development of wavelets, and wavelet applications such as The Mathworks Wavelet Analsysis Toolbox, Wavelab and the Rice Wavelets Tools to name a few [154]. Further to this, practical implementations can be realised with tools such as SharcLab [155] that effectively interfaces Matlab™ code into Bittware general purpose DSP hardware [156]. Such tools ease pre-deployment development before committing resources to any special to type hardware.

In this work, extensive use is made of Matlab™ software for developing and simulating model behavior. Matlab™ is a highly capable, well respected product used by industry and academia around the world for research and development activities.

3.3.2 The Discrete Wavelet Transform (DWT)

The CWT produces coefficients for every scale up to a scale where we choose to terminate the analysis. Unfortunately this can lead to a significant amount of computational effort. In this work the time taken to run simulations using the CWT technique under Matlab™ have been restrictive, employing the computationally efficient DWT has the potential for reducing simulation times and increasing the performance of deployed systems.

Evidently if we use scales and positions based on powers of 2 (dyadic scales and positions) the analysis will be much more efficient and just as accurate [110]. This technique forms the basis for the DWT. Stephane Mallet [110, 111] developed a way to carry out the DWT using filters and his algorithm yields the fast wavelet transform.

The DWT process involves using successive, complementary low-pass and high-pass filters to split the signal under analysis into its approximation and detail coefficients. With traditional digital filtering this would leave us with two signals both with the same number of sample points as the original. However, the DWT process down-samples both signals by a factor of 2. The decomposition process can be iterated as many times as required, the approximation coefficients being the starting point for the following set of filters. The wavelet decomposition tree shown in Fig. 3.11 represents this iterative procedure where 'S' is the original signal, 'cAx' represents the approximation coefficients and 'cDx' represent the detail coefficients.

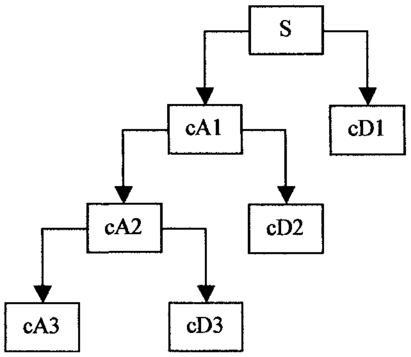


Fig. 3.11: DWT tree

3.3.3 Wavelet Packet Analysis

Wavelet packet analysis is a generalisation of wavelet decomposition where both approximations and details can be decimated into further approximations and details. Such decomposition leads to a binary tree type structure as shown in Fig. 3.12, where the original signal 'S' is decomposed in to approximations 'A' and details 'D', Each of these branches is decomposed in to further approximations and details, and so on. Although wavelet packet analysis gives us great flexibility in how we encode or process a signal, we must measure the contribution that further branches or splits have on the end product. Sometimes an entropy-based criterion is used to decide the level of decomposition for a given problem.

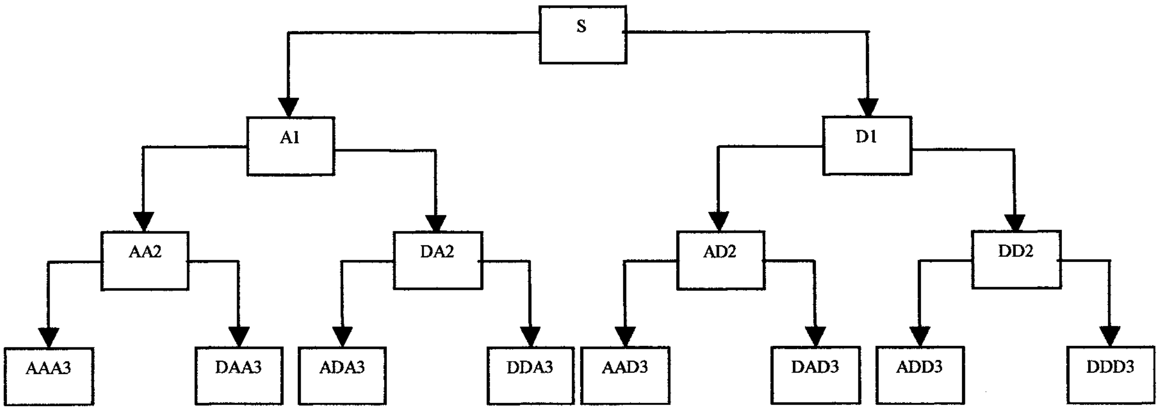


Fig. 3.12: Wavelet packet tree

3.4 Direct Detection Using Wavelets

The temporal location of features afforded by wavelet analysis can actually be employed for the direct detection of signals. The coefficient intensity plot of all scales for a 'db2' wavelet from 1 to 75 is shown in Fig. 3.13, it clearly indicates the time-based correlation of wavelet coefficient features with the source sample signal shown in Fig. 3.14. The wavelet coefficients at scale 22 shown in this figure have been isolated and plotted on a common time axis with the original signal, showing clear temporal relationships between the two. The amplitude of the graph is not important here, however, the amplitude of the signal can be taken as 1 V; the amplitude of the scale coefficient has no units and is related to the 'fit' between the scaled wavelet and the source signal as discussed in section 3.3.1.

Whilst direct detection employing wavelets is unlikely to yield performance improvements over established enhanced detection techniques, it does emphasise the contrast with Fourier methods. Simulations indicate that implementing a direct detection scheme employing wavelets on a 41.666 Mb/s OOK NRZ signal with multipath distortion offers an improvement over a comparable, simple threshold detector under the same conditions.

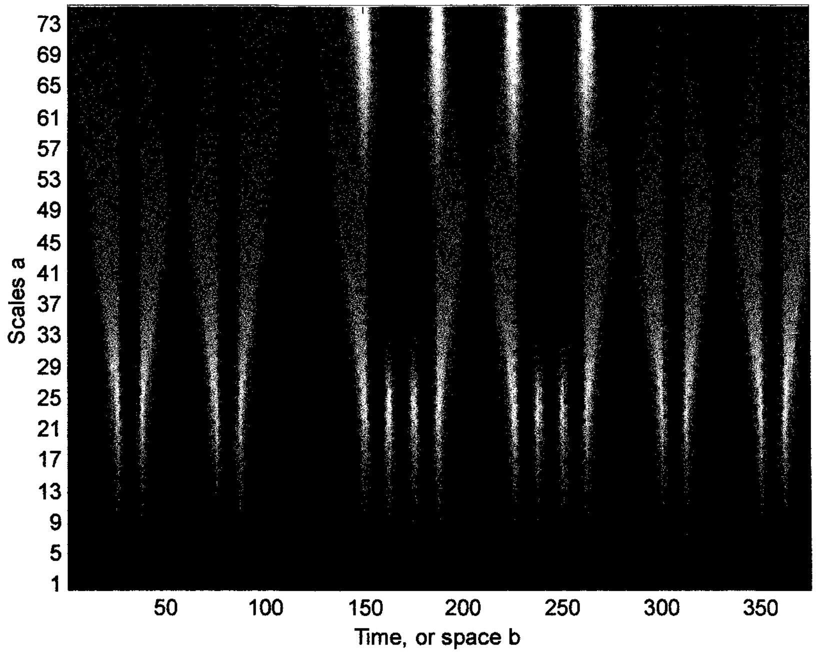


Fig. 3.13: CWT ('db2' wavelet) all scales to 75 for a 40 Mbps OOK RZ signal (Fig. 3.14)

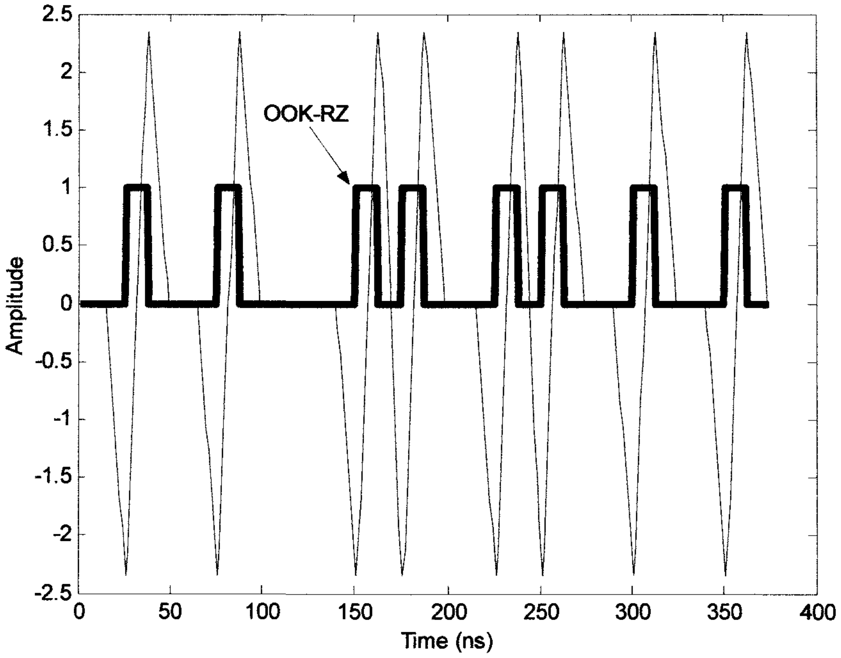


Fig. 3.14: OOK RZ and single scale ('db2' wavelet) CWT coefficient for 40 Mb/s OOK RZ signal

The simulation model for the system is shown in Fig. 3.15. The input bits (1 or 0) are passed to the transmitter filter that has a unit amplitude rectangular impulse response $p(t)$, with a duration $T_b/2$ (duty cycle $\gamma = 0.5$). The pulses are scaled by the transmitted optical signal power, in this case 4*Pavg*, where *Pavg* is the average transmitted signal power. The signal is then passed through the multipath channel with an impulse response $h(t)$, which is discussed more fully in chapter 6. The received optical power is scaled by the photodetector responsivity, taken to be unity and white Gaussian noise is then added before the signal is passed through an analogue anti-alias filter. The signal is sampled by an ADC with a sample interval of 1 ns. Finally, the signal is passed to a threshold detector with a threshold set between the minimum and maximum signal excursion.

For simulation purposes the entire process is discrete with a sampling interval of 1 ns, therefore the anti-alias filter and ADC is notional in the model, but is included as it would be required in practice since the received signal would be analogue. The signal amplitude is always unity at the transmitter; $n(t)$ being varied to produce the correct SNR. R, is also taken as unity.

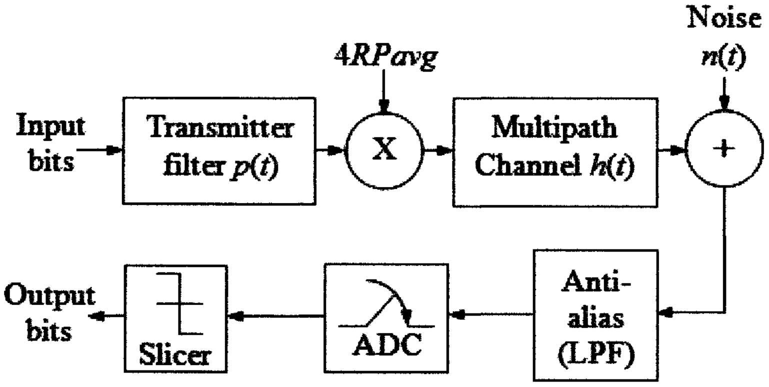


Fig. 3.15: OOK RZ threshold based detection scheme

The program flow chart for the model of Fig. 3.15 is shown in Fig. 3.16, and the program listing on the accompanying CD.

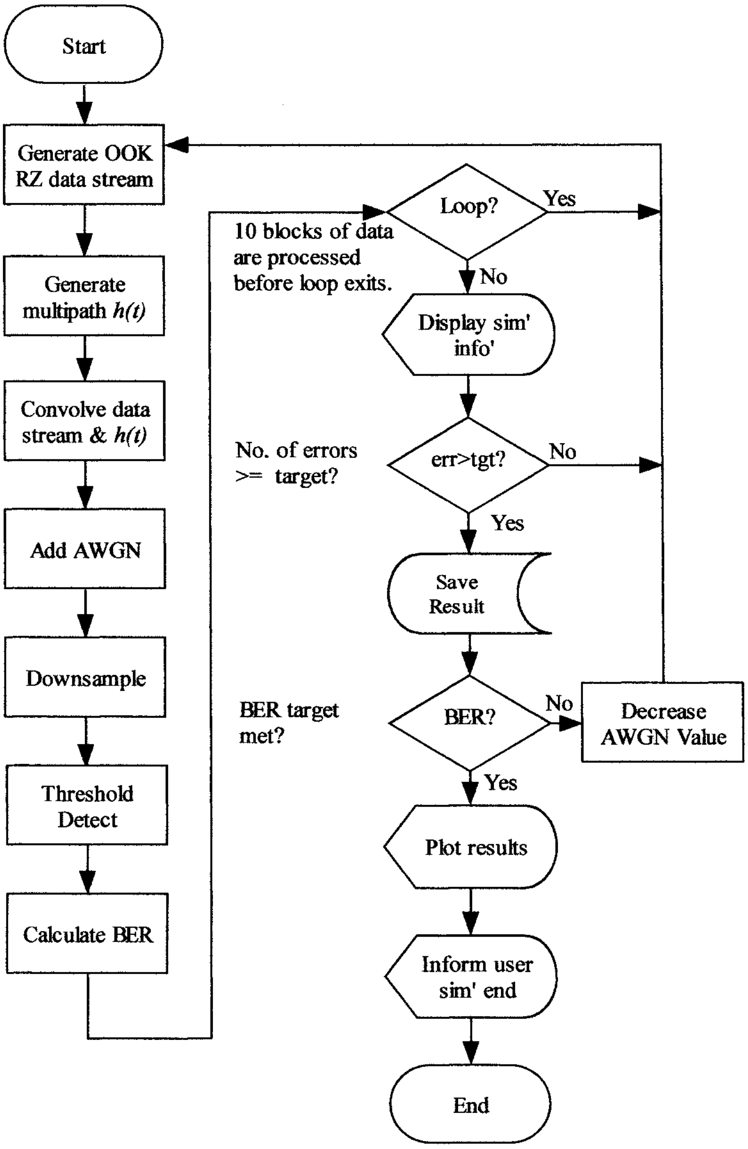


Fig. 3.16: Flowchart for Fig. 3.15.

The simulation model for direct wavelet detection in shown in Fig. 3.17 and is identical to Fig. 3.15 up to the wavelet analysis section. In this model the CWT is employed to produce a single scale coefficient, in this case '17' for the 'sym2' wavelet.

Subsequent to the CWT section two options have been implemented, the first being a simple wavelet threshold detector (WTD) with a threshold value based on observation of the coefficient maximum and minimum. The second being wavelet slope detection (WSD) where the difference between the coefficient maxima and minima are compared with a threshold value. In both cases values above the threshold are detected as a binary '1' and below as binary '0'.

The program flow chart for the model incorporating both options is shown in Fig. 3.18, and the program listing on the accompanying CD.

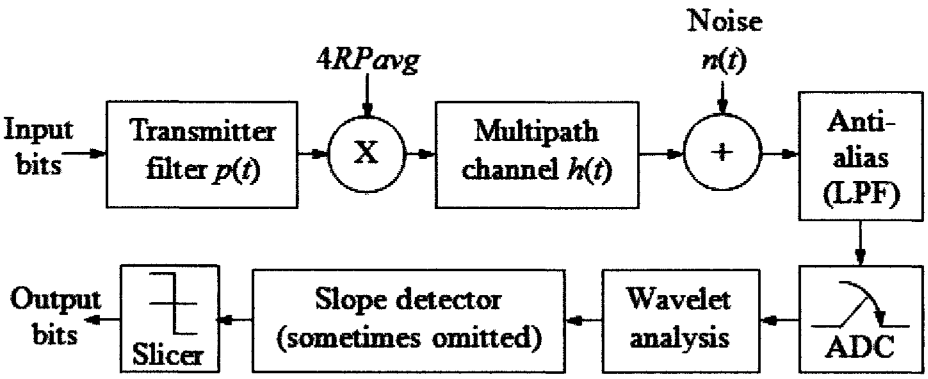


Fig. 3.17: OOK RZ wavelet based detection scheme

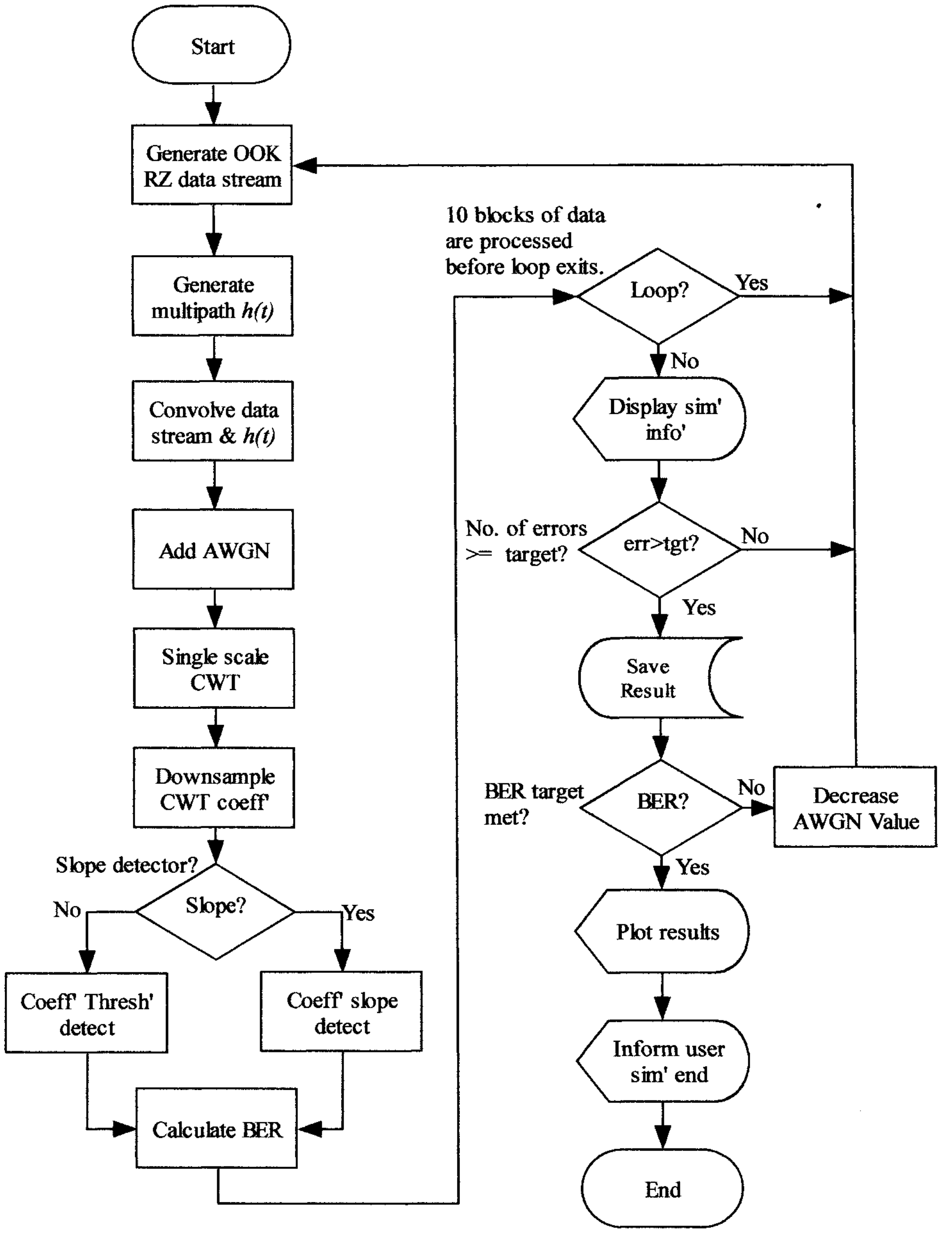


Fig. 3.18: Flowchart for Fig. 3.17

Fig. 3.19 shows the simulation results for the bit error rate (BER) against SNR for 41.66 Mb/s OOK-RZ data format for WTD and WSD for a multipath channel with a normalised delay spread (D_T) of 0.34; D_T is discussed more fully in chapter 6. Also shown for comparison is the result for the basic threshold detection scheme. As can be observed from the figure for BER of 10^{-5}, a system employing WTD and WSD offers improvements in SNR of approximately 2 and 4.5 dB, respectively compared with the traditional scheme.

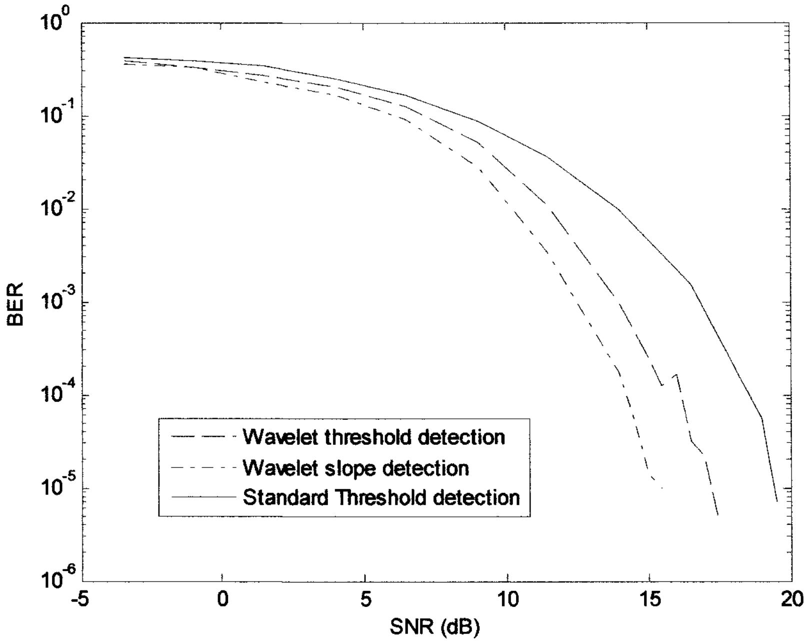


Fig. 3.19: BER against SNR for 41.666 Mb/s OOK-RZ using WTD, WSD and a basic threshold receiver

3.5 Summary

In this chapter Fourier analysis has been introduced and briefly reviewed followed by wavelet analysis and in particular the CWT. The limitations of Fourier methods in terms of identifying and temporally isolating signal features was investigated and compared with time-scale-amplitude representations afforded by the CWT confirming the techniques ability to identify signal features.

The computationally efficient DWT was briefly introduced followed by the feature rich technique of Wavelet Packet Analysis. It was noted that the DWT may yield computational performance enhancements in work of this type.

Finally, simulation models and results for the direct detection of an OOK RZ signal, subjected to noise and multipath dispersion were introduced. The models included a basic threshold detector, a wavelet threshold detector and a wavelet slope detector. For both wavelet capable models the simulations yielded results superior to that of the simple threshold detector, confirming that significant signal features can be isolated in time potentially enabling the method to form the foundation of a receiver system.

Chapter 4

Artificial Intelligence and the Wavelet-AI Receiver

4.1 Introduction

The aim of this chapter is to provide an introduction to the subject of artificial intelligence and its subsequent partnering with the previously introduced CWT. Whilst the subject of artificial intelligence is more established than that of wavelet analysis with a history of some 5 decades it is only relatively recently that is has found employment in solid implementations and the field is developing rapidly [157] with thousands of publications on the subject available covering countless applications. Many papers cover aspects of communications [162 – 170], particularly with respect to channel equalisation. It is a broad and complex subject, therefore an in depth review is beyond the scope of this work. The intention here is to provide sufficient material to enable understanding of the concepts used in this thesis and to briefly introduce facets of the topic that may progress this work further in the future. Section 4.2 briefly reviews the subject of artificial intelligence and popular implementations. Section 4.3 introduces a neural network based receiver for OOK RZ $\gamma = 0.5$ and the results of subsequent simulations. Section 4.4 introduces Wavelet-AI receiver architecture and the results of initial simulations. The section continues with parameter simplification including sample window reduction, wavelet scale reduction and neural network size reduction. Finally the chapter concludes with a summary in section 4.5.

4.2 Artificial Intelligence

AI systems employ simple processing elements working in parallel; these elements are loosely based on the biological neurons found in the human brain. Research into AI has been active for many years and has resulted in its use in numerous fields from the adaptive techniques used in communication electronics to the classification of medical conditions [157].

There are many varieties of artificial neural network (ANN), some of them quite complex. However, the basic structure of a simple network of artificial neurons is relatively straightforward to understand. The construction of a single artificial neuron is shown in Fig. 4.1, it has X_1 to X_n inputs that are multiplied by weights W_1 to W_n that increase or attenuate the input. The summing junction 'Σ' combines the inputs to produce a result that is subsequently modified by an activation function 'F' to give an output 'Out'. A neuron also has a bias input (not shown) that is used to shift the input by some fixed amount.

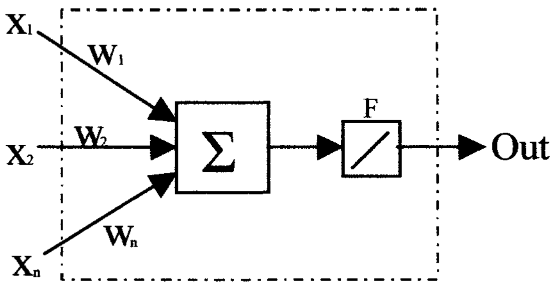


Fig. 4.1: Schematic of an artificial neuron.

There are few types of well known activation functions, including linear, log-sigmoid and tan-sigmoid. These functions impose some characteristic on the output from the summing junction, for example, holding the minimum and maximum values between '0' and '1' respectively. The basic neuron can become part of a network consisting of one to many layers as in Fig. 4.2.

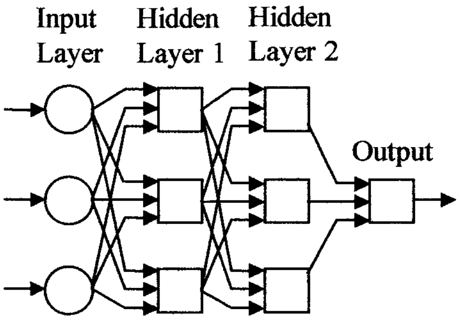


Fig. 4.2: Structure of a neural network.

Only the first and last layers have connections to the outside world, the inner links being connections to other neurons only. A network of neurons can solve complex problems for which no analytical solution exists [157], the detail of how this is accomplished is beyond the scope of this text; however, many excellent texts on the subject, including [158] are available.

For ANNs to produce meaningful outputs they must be trained. There are two types of training: supervised and unsupervised. This work focuses on supervised training where the ANN is presented with a particular input vector of data to be classified and a target vector corresponding to the correct output. Together these two vectors are referred to as the training pair and are used in conjunction with a training algorithm to modify the behaviour of the neural network in such a manner that it is able to classify an input it has never seen before.

4.2.1 Feed-Forward Back-Propagation Networks

Back-propagation is applied to multi-layer networks with non-linear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by the user. A two layer network, where the first layer employs a sigmoid activation function and the second layer a linear function can be trained to approximate any function (with a finite number of discontinuities) arbitrarily well. [157]

Although various training algorithms are available the underlying principles for their implementation is similar. Each input to the network produces an output that is compared with the corresponding value of the target set. The training algorithm then modifies the weights associated with the neuron inputs in an effort to minimise the difference between the input and target vectors. This process may continue many times until a desired error value is attained. An interesting but undesirable property of neural networks is that they can be over-trained, here the network effectively ‘tunes’ itself to the training inputs and loses its ability to generalise. In such a case the network would yield minimal errors between its training and target vectors, however, a new set of input and target vectors would produce significant errors. It is therefore important to adopt a training scheme that minimises the effect of over training. In this work interest in ANNs is primarily confined to its pattern recognition capability that is used to detect a pattern from imperfect data, corrupted by noise and dispersion.

A further problem associated with network design is that of dimensionality. Unfortunately there is no rigorous way of determining the optimum number of layers or neurons per layer [157]. Such design is usually done by experience, empirically or by an iterative network pruning process.

4.2.2 Radial Basis Function Networks

As the name implies radial basis function networks use the radial basis function as their transfer function, which has a maximum of ‘1’ when its input is zero. The radial basis neuron architecture is also different from the neurons described earlier, here the net input to the radial basis transfer function is the vector distance between its weight vector ‘w’ and the input vector ‘p’, multiplied by the bias ‘b’ [157]. Radial basis function networks consist of two layers, a hidden radial basis layer and an output linear layer. Such networks may require more neurons than the feed-forward back-propagation networks but they can be trained in a fraction of the time. The implementation of a radial basis function as a maximum likelihood sequence estimator (MLSE) has been demonstrated by Sung-Hyun Heo *et al* [159].

A special case of the radial basis function network is the probabilistic neural network, which can be used for classification problems and may therefore be useful in this type of work. This variety of network calculates the distance between the input and training vectors to produce a vector that indicates how close the input is to the training input. The second layer totals the elements in this vector for each class of input to produce an output vector of probabilities. Finally a 'compete' transfer function in the second layer picks the maximum of these probabilities and produces a '1' for that class and '0' for other classes.

4.2.3 Recurrent Networks

Elman and Hopfield networks are both types of recurrent network. Unfortunately, Hopfield designs may be unstable and are rarely used in practice, and therefore they will not be considered further in this work. Elman networks on the other hand are two-layer back-propagation networks with the addition of a feed-back connection from the output of the hidden layer to its input [157]. This feedback path allows Elman networks to learn to recognize and generate temporal patterns, as well as spatial patterns [157].

4.3 Direct Detection Using Neural Networks

Use of AI in this work has focussed on feed-forward back-propagation networks as these are arguably the most stable, best understood and widely deployed of the artificial neural networks. Initial investigation into other artificial intelligence architectures yielded inferior results or uncovered technical difficulties not encountered with the feed-forward back-propagation types. However, it must be stressed that the wide scope of the work contained in this thesis limited the time available to investigate alternative techniques to significant depth; the suitability of other techniques should not therefore be discounted. Even so, there are still substantial well-known and largely unresolved difficulties with feed-forward back-propagation networks, not least being dimensionality in terms of number of layers, types of transfer functions and number of neurons employed.

This section focuses on the use of feed-forward back-propagation networks for direct detection of noisy multipath OOK signals, in a similar way to the treatment of direct detection with wavelets in chapter 3. Ultimately, the goal of this chapter is to couple the wavelet and AI sections together to form a receiver unit based on the two techniques.

A variety of network layer architectures and transfer functions were modelled and it was found that dimensionality did not play a major role in classification performance. However, due to time constraints evaluation of other attributes, such as optimised learning times, were not considered. A network consisting of two layers with 100 neurons in the first and 1 in the output layer was found to provide classification results that were consistently as good as more complex structures. This layer architecture was therefore employed in all simulations discussed in this section. The first layer uses a log-sigmoid transfer function given by:

$$a = 1/(1 + \exp(-n)) \qquad (4.1)$$

Where n would relate to the output of the summation section of any neuron.

The second layer uses a linear transfer function, both functions being show in Fig. 4.3 and Fig.4 .4, respectfully.

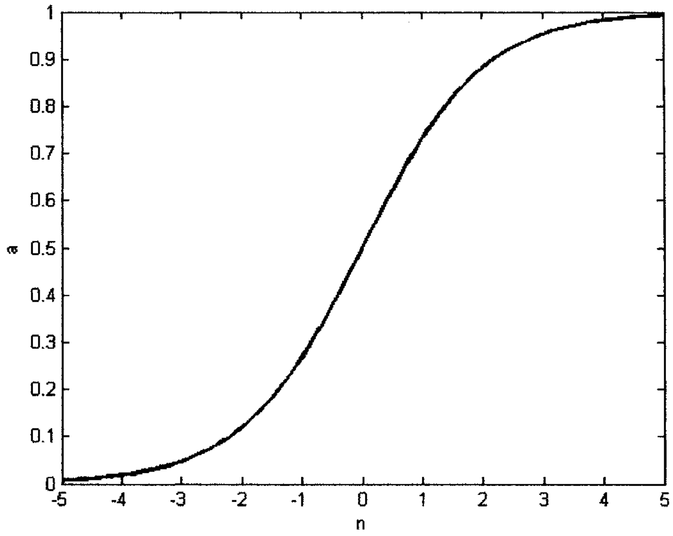


Fig. 4.3: Log-Sigmoid transfer function

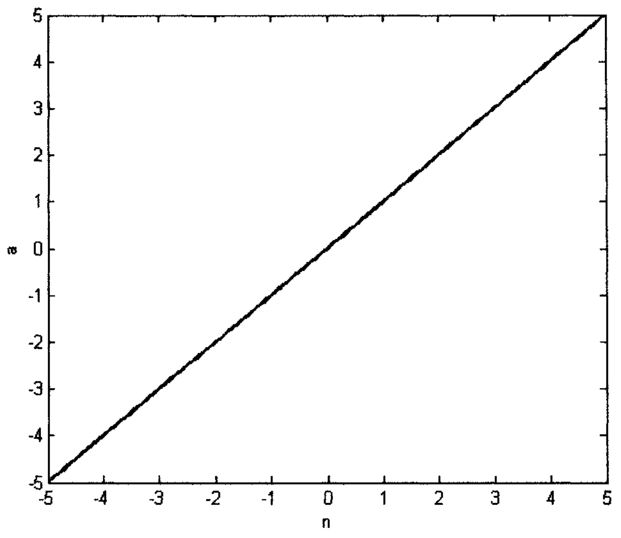


Fig. 4.4: Linear transfer function

Another major consideration when using a feed-forward back-propagation network is the selection of training algorithm. A number of well-established algorithms exist and are available in the Matlab™ simulation package as shown in Table 4.1. A number of these algorithms were tried, however, the most successful was the conjugate gradient with Powell/Beale Restarts [160] and this was used throughout the work contained in this section.

Table 4.1: Training algorithms.

	Algorithm
1	Levenberg-Marquardt
2	BFGS Quasi-Newton
3	Resilient Backpropagation
4	Scaled Conjugate Gradient
5	Conjugate Gradient with Powell/Beale Restarts
6	Fletcher-Powell Conjugate Gradient
7	Polak-Ribiére Conjugate Gradient
8	One-Step Secant
9	Variable Learning Rate Backpropagation

The neural network detector employed in the simulation model shown in Fig. 4.7 analyses data on a consecutive 5 bit sliding window, where the network is trained to detect the value of the centre bit from the 5 bit sample. For bits 1 to 5 the neural network would detect bit 3; next the window would be indexed by 1 and the network would detect bit 4 from bits 2 to 6, and so on as shown in Fig. 4.5. The rationale behind this being that the neural network is able to extract further information from the 5 bit window to help it make the correct decision. This may be particularly important under the influence of multipath dispersion where previous bits interfere with the present bit. In all simulations the incoming data is sampled at 1 ns intervals and assumes perfect timing alignment, all the samples being passed on to the neural network.

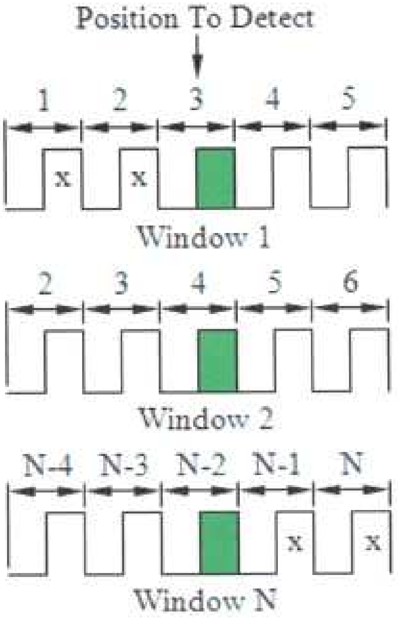


Fig. 4.5: The 5 bit sliding window

Fig. 4.6 shows how the samples from the 5 bit window are arranged by the AI section and processed by the neural network. The use of this 5 bit window would mean that bits 1 and 2 at the start of a transmission and bits N and N - 1 at the end of a transmission would never be detected unless some remedial action was taken. In these simulations two zeros are added to the start and end of all transmissions to take account of this.

In this architecture the number of neurons in the first or hidden layer can be scaled to any number that is computationally possible. The output layer however, will always consist of one neuron due to the single digit classification the network is making.

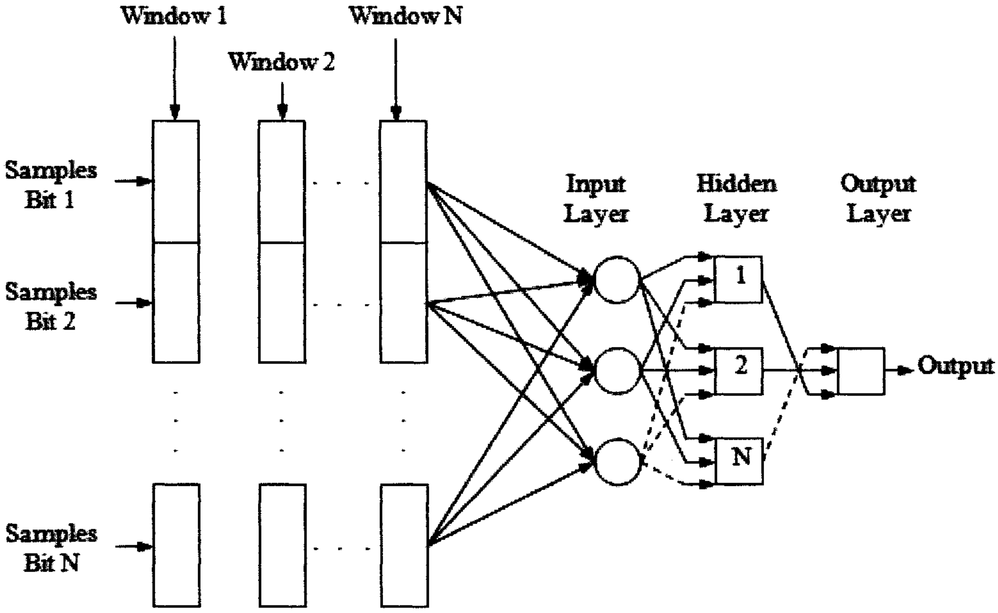


Fig 4.6: Matrix storage of sampled signal

The slicer or threshold detector of Fig. 4.7 is required due to the variation in output of the neural network. The network is trained to provide a numerical output of '1' when it detects a binary mark as the centre bit of the 5 bit window, and numerical '0' otherwise. However, small variations in output occur, even for near ideal signals, which would be meaningless in a binary sense. To overcome this, a threshold detector is used to confine the output to binary levels, with the threshold set midway between the minimum and maximum signal excursions.

Before being allowed to classify an unknown signal the neural network is trained on a known series of 1500 binary symbols (bits). For this model training is confined to signals with particular SNR figures, rather than a training signal for every SNR subsequently simulated. The network was therefore trained with SNRs of 11 dB, 12 dB and 13 dB. These values were empirically selected to be near the noise value corresponding to a BER of approximately 10^{-6}.

The transmitter filter of Fig. 4.7 has a unit-amplitude rectangular impulse response $p(t)$, with a duration of one bit $T_b/2$. The output of the transmitter filter is scaled by the peak detected signal photocurrent $4P_{avg}$, where P_{avg} is the average received optical signal power. The signal is then subject to multipath distortion, discussed further in chapter 6, with a D_T of 0.819. R is the photodetector responsivity, taken to be unity in this chapter, and the signal independent shot noise, $n(t)$, is then added to the signal. In a practical system the signal would be sampled by an ADC, however, in this case the model is already in discrete sampled form; therefore the ADC shown in the model is notional and no anti-alias filtering is employed. The signal is finally passed to the AI section and threshold detector as described previously.

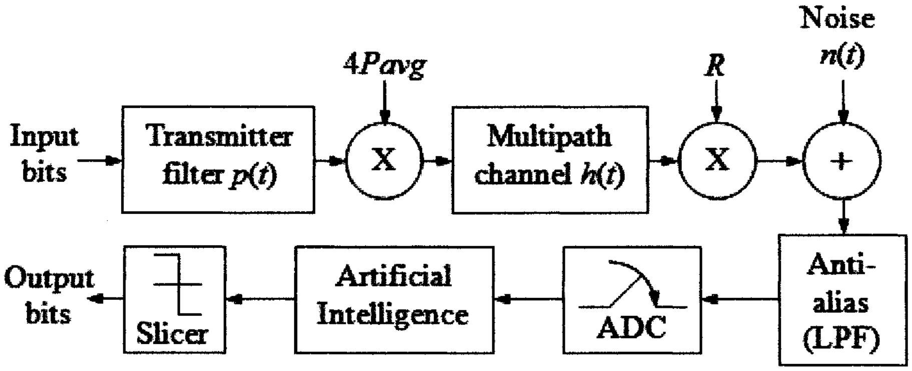


Fig. 4.7: OOK RZ simulation model employing a direct neural network detection scheme

The simulation model of Fig. 4.7 was used to detect 100 Mb/s OOK RZ with a duty cycle 'γ' of 0.5. The program flow chart for the model is shown in Fig. 4.8 and the program listing on the accompanying CD.

The results of these simulations, plotted as BER against SNR for three different trained networks are shown in Fig. 4.9, Three independent simulation runs for each noise figure were completed. As can be seen there is an approximately 4 dB variation in performance, with a 3 dB variation for the curves relating to 13 dB network training. The 11 dB trained network has considerably less variation, being around 1 dB. The results point to significant sensitivity to noise levels, such variation could lead to severe burst errors in practical implementations. Also, in practical circumstances the opportunity to train at other than noise figures imposed by the immediate environment would be limited without additional complication to the system; however, the simulation strategy used dramatically reduces simulation time.

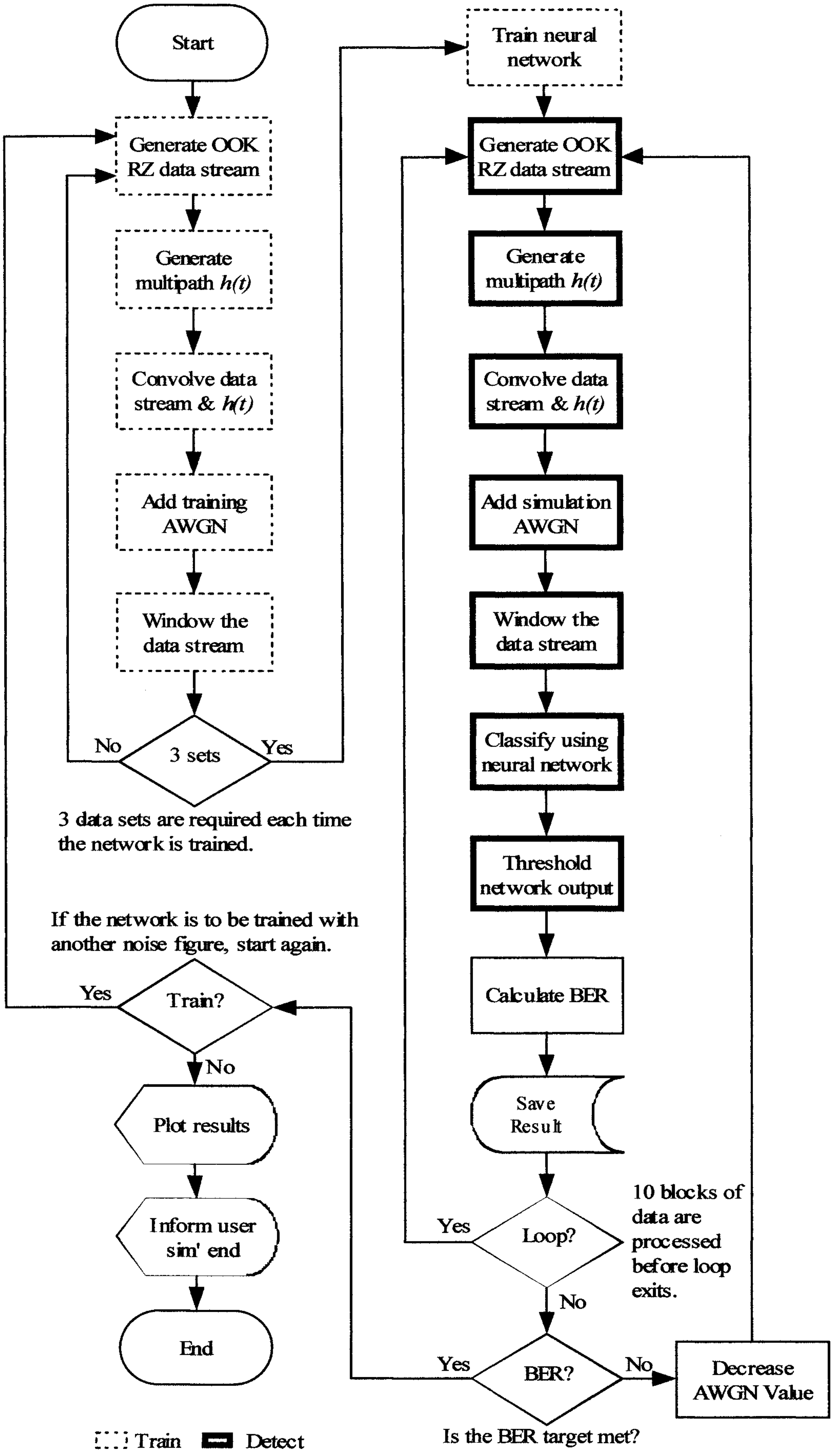

Start
Generate OOK RZ data stream
Generate multipath $h(t)$
Convolve data stream & $h(t)$
Add training AWGN
Window the data stream
No
3 sets
Yes
3 data sets are required each time the network is trained.
If the network is to be trained with another noise figure, start again.
Yes
Train?
No
Plot results
Inform user sim' end
End
Train neural network
Generate OOK RZ data stream
Generate multipath $h(t)$
Convolve data stream & $h(t)$
Add simulation AWGN
Window the data stream
Classify using neural network
Threshold network output
Calculate BER
Save Result
Yes
Loop?
No
10 blocks of data are processed before loop exits.
Yes
BER?
No
Decrease AWGN Value
Is the BER target met?
Train
Detect

Fig. 4.8: Program flow chart for simulation model shown in Fig. 4.6

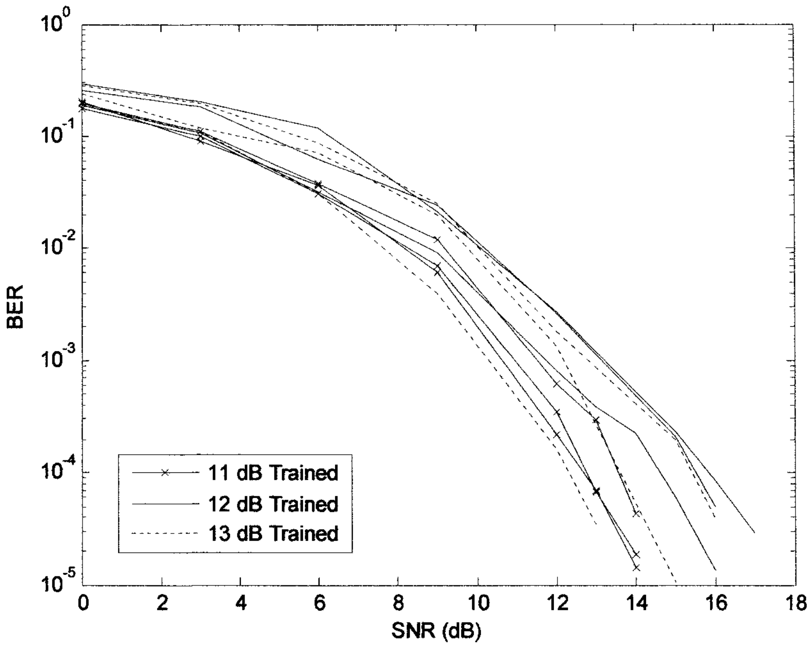


Fig.4.9: BER against SNR for neural network detection of OOK RZ, trained at 11 dB, 12 dB and 13 dB SNRs

4.3.1 Direct Detection Using Filtering and Neural Networks

In this section consideration is given to simple filtering techniques to overcome the apparent variation in results caused by noise as indicated in Fig. 4.9. As explained more fully in chapter 6, work by Carruther and Kahn [25] show that for OOK NRZ a low pass filter with a normalised cut-off frequency of $0.6/T_b$ proved to be the most effective in combating noise in a simple threshold receiver system subjected to multipath propagation. Here, a LPF with a normalised cut-off of frequency of $0.35/T_b$ is used which reflects OOK RZ, with a duty cycle 'γ' of 0.5.

The filter is a 5th order digital low pass Butterworth type with a transfer function $H(z)$ given by [161]:

$$H(z)=\frac{B(z)}{A(z)}=\frac{b(1)+b(2)z^{-1}+\cdots+b(n+1)z^{-n}}{1+a(2)z^{-1}+\cdots+a(n+1)z^{-n}} \tag{4.1}$$

The Matlab™ function 'butter' can be used to determine the filter coefficients, where the cut-off is normalised to the Nyquist frequency. The magnitude of the filter response at the cut-off frequency is given as $0.5^{0.5}$. The filter coefficients for a 100 Mb/s OOK RZ with γ = 0.5 for a 1 ns sampling interval are given in Table 4.2. The simulation model is shown in Fig. 4.10 and the program listing on the accompanying CD.

Table 4.2: Butterworth LPF filter coefficients

Order	Coefficients	
	a	*b*
0	1.000	0.0052
1	-2.1750	0.0262
2	2.3095	0.0523
3	-1.3184	0.0523
4	0.4028	0.0262
5	-0.0514	0.0052

The operation of the model is as described in the previous section save for the inclusion of the low-pass filter elements. The program flow chart for the model is shown in Fig. 4.11, where the filtering process can be clearly seen. In all cases the delay caused by the filter is removed and perfect bit synchronisation is achieved. The ADC section is notional as the model is discrete from end to end based on a 1 ns sample interval as previously described; however, the block is included in the schematic since the transmission part of the model could in theory use a much higher sampling rate should we wish. The results of the simulation for direct neural network detection of an OOK RZ incorporating a LPF are plotted as BER against SNR depicted in Fig. 4.12. The signal was subject to the same channel imperfections as the previous section. The results show a significant reduction in variation with all but one curve closely grouped.

Ignoring the one outlying result at 13 dB the variation is 1 dB compared to the 4 dB variation of the unfiltered case.

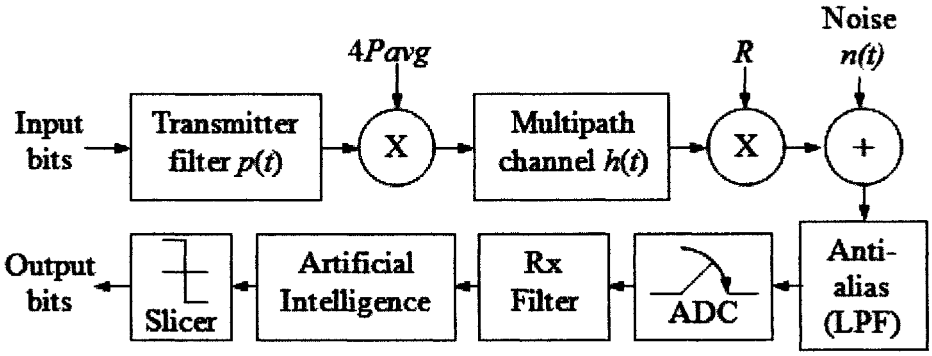


Fig. 4.10: OOK RZ simulation schematic employing a direct neural network detection scheme with incorporated LPF

Such results cannot be described as conclusive in relation to the ultimate type of filter or its cut-off frequency in relation to this type of network. However, it would indicate that a neural network of this type would benefit from some form of pre-processing if results are to be obtained that are not subject to impractically wide variations.

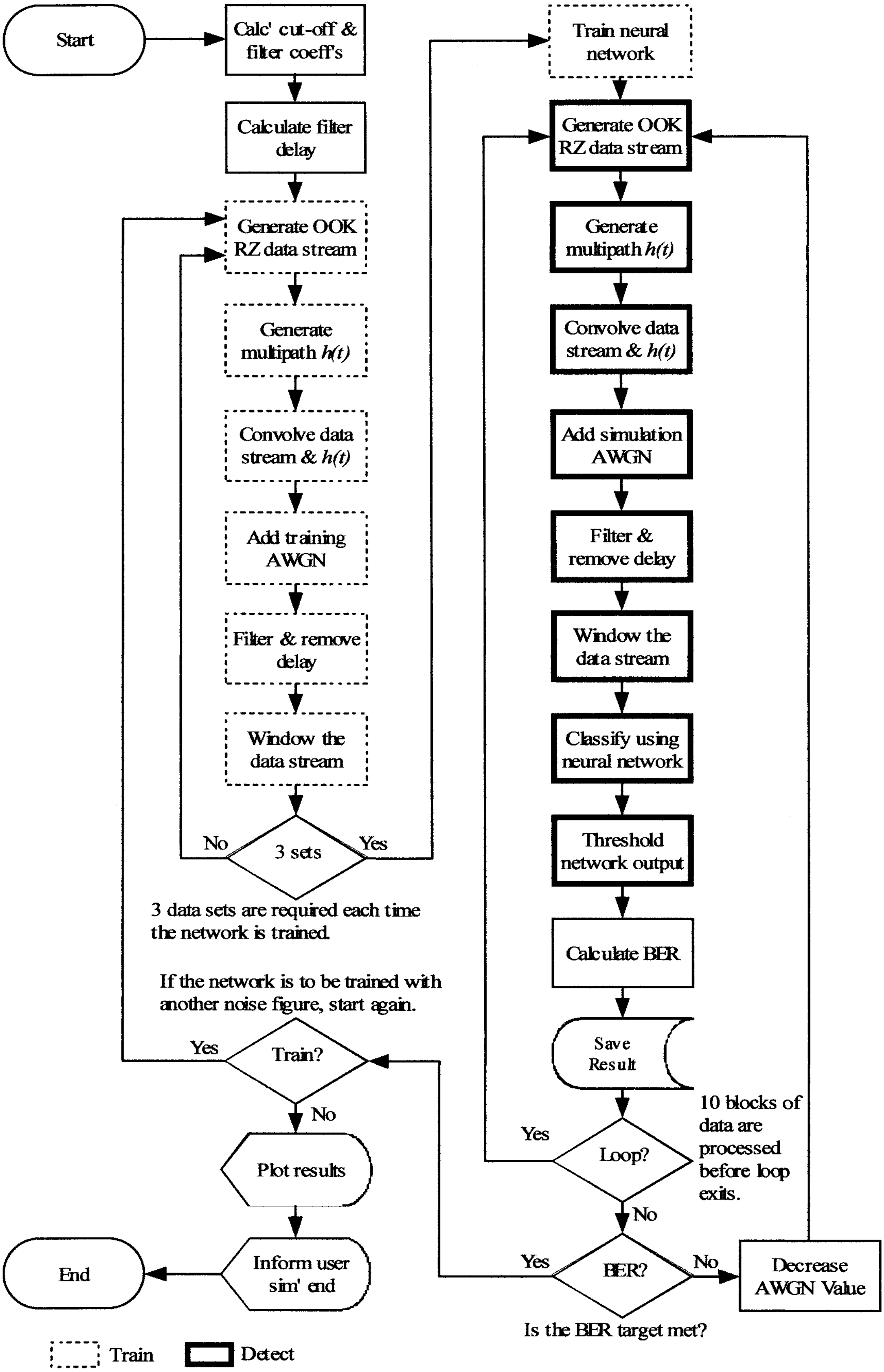


Fig. 4.11: Program flow chart for simulation model shown in Fig. 4.10

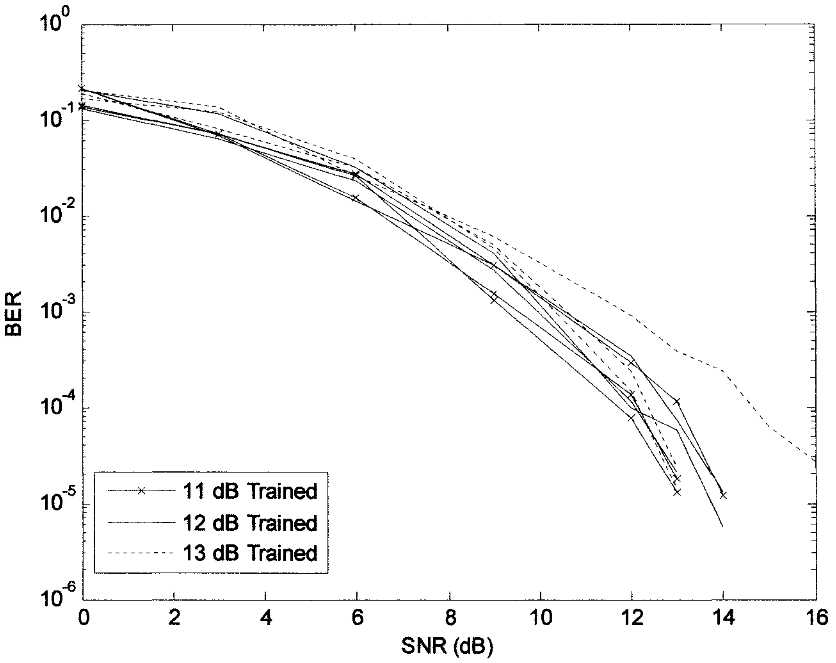


Fig.4.12: BER against SNR for neural network detection of OOK RZ incorporating a LPF, trained at 11 dB, 12 dB and 13 dB SNRs

4.3.2 Direct Detection Using Filtering and Neural Networks with Down-Sampled Inputs

Section 4.3 introduced the concept of direct detection using neural networks, which whilst showing that detection is clearly possible, indicated that variations in results for a particular set of parameters could occur. Section 4.3.1 showed that introducing a pre-detection filter reduced this variation producing tighter grouping of performance curves. In this section we briefly look into the effects of down-sampling in an effort to reduce computational burden.

The model architecture of Fig. 4.13 is basically the same as that described for Fig. 4.10; however, here a down-sampling section is introduced between the filter and neural network. The function of the down-sampler is simply to reduce the number of sample points passed to the neural network. The filter section passes 10 samples per bit period based on a 1 ns sample interval and a 100 Mb/s OOK RZ, $\gamma = 0.5$ signal to the down-sampler, which then discards the first 9 samples of the symbol. This provides the neural network with one sample taken at the end of the bit period. The result is a '5 bit window' consisting of 5 sample points rather than 50 as in the previous 2 cases of section 4.3 and 4.3.1. Again, filter delay is removed and perfect symbol (bit) synchronisation achieved with the sample points coinciding at the received, filtered signal maxima.

In a practical implementation, the down-sampling section would be an ADC running at the desired sample rate. However, this would require the inclusion of an analogue anti-alias filter before prior to the down-sampler, which could perform the combined functions of noise, and anti-alias filtering.

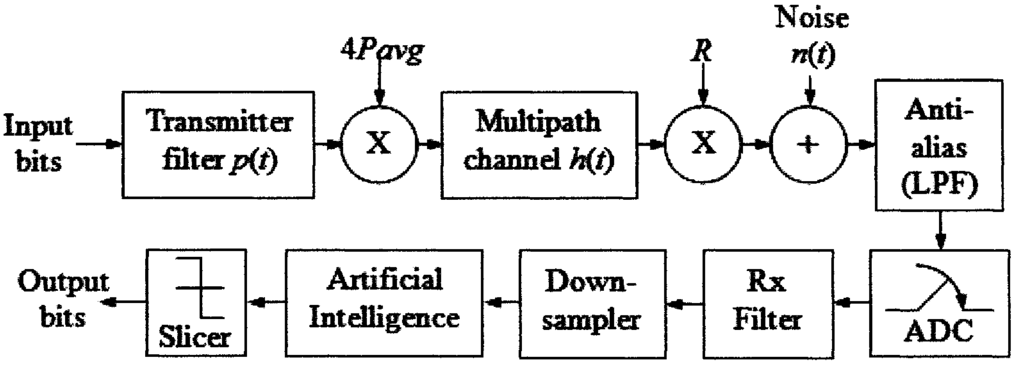


Fig.4.13: OOK RZ simulation schematic employing a direct neural network detection scheme incorporating a LPF and a down-sampler

The flowchart for the simulation program is shown in Fig. 4.14, and indicates the new down-sampling activities in addition to the filtering introduced in the previous section. The program listing is on the accompanying CD. The results of the unfiltered model, shown in Fig. 4.9, indicated that the lower noise value used for training provided the best performance. In this section we reduce the training SNR figures of the previous two sections by 1 dB to 10 dB, 11 dB and 12 dB in anticipation of the results improving further. However, the results of the simulations shown in Fig. 5.15 reveal a marked deterioration in performance with the best SNR being 15 dB for a BER of approximately 10^{-5}, compared with 13 dB of the previous two sections. The results also show a variation of 5 dB from the best to worst performance curves indicating that the down-sampling process has negated the advantage of using a filter.

The evidence from this and the previous two sections suggests that pre-processing the signal and employing multiple sample points per symbol enhance the neural networks detection performance for the particular architecture chosen. The characterisation of optimum low-pass filter types and values is beyond the scope of this work. In reality, the number of samples available for the neural network to process will in practical terms be governed by the speed and cost of the technology available. Later simulations ensure numerical sample parity of traditional benchmark models so that a true like for like performance comparison is obtained.

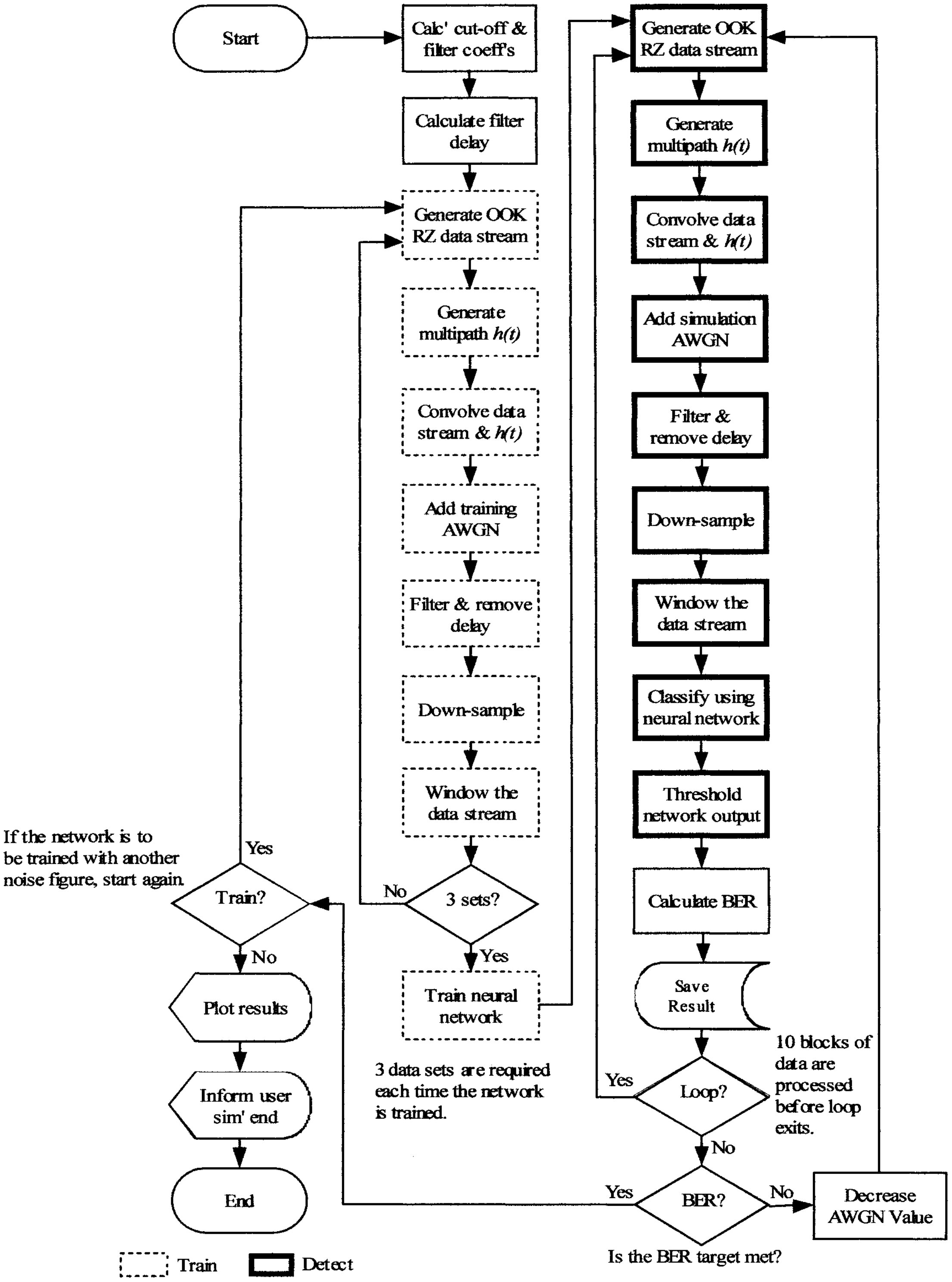


Fig. 4.14: Program flow chart for simulation model shown in Fig. 4.13

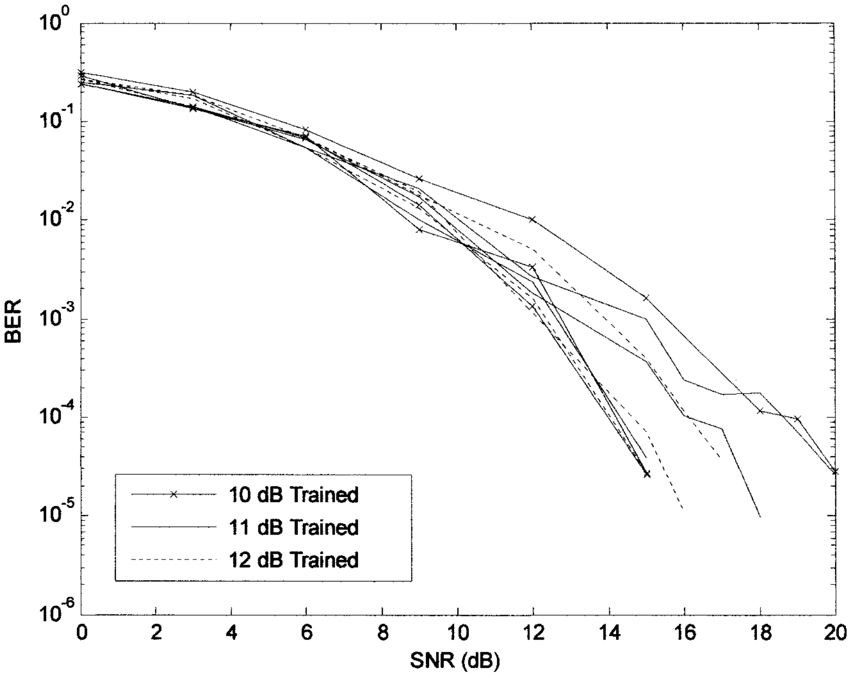


Fig.4.15: BER against SNR for neural network detection of OOK RZ incorporating a LPF and down-sampler, trained at 10 dB, 11 dB and 12 dB SNRs

4.4 The Wavelet-AI Receiver

Wavelet analysis was introduced in chapter 3 and Artificial Intelligence in sections 4.1 through to 4.3 of this chapter. In the following sections these principles are combined into a single model architecture where the wavelet analysis is used for signal pre-processing (feature extraction) and the neural network for signal detection (classification). The focus remains on 100Mb/s OOK RZ, $\gamma = 0.5$ signal detection.

4.4.1 Model Architecture

The simulation model is shown in Fig. 4.16, where the Wavelet analysis section employs the CWT algorithm using a 'Sym2' wavelet at intervals of 2 from scales 2 to

74 in all cases. The model uses a 5-bit window as discussed in section 4.3 and shown in Fig. 4.5. The organisation of the CWT coefficients is shown in Table 4.3 and a schematic of information flow is shown in Fig. 4.17, where the incoming data stream is decimated into 5 bit windows as previously described. Each window is processed by the CWT to produce wavelet coefficients, which are then sorted into a matrix where each column contains the coefficients that relate to a single 'window'. Each of the coefficient columns is passed in turn to the neural network for subsequent binary classification. The neural network contains 100 neurons in the first layer and 1 in the output layer, again as described earlier in this chapter.

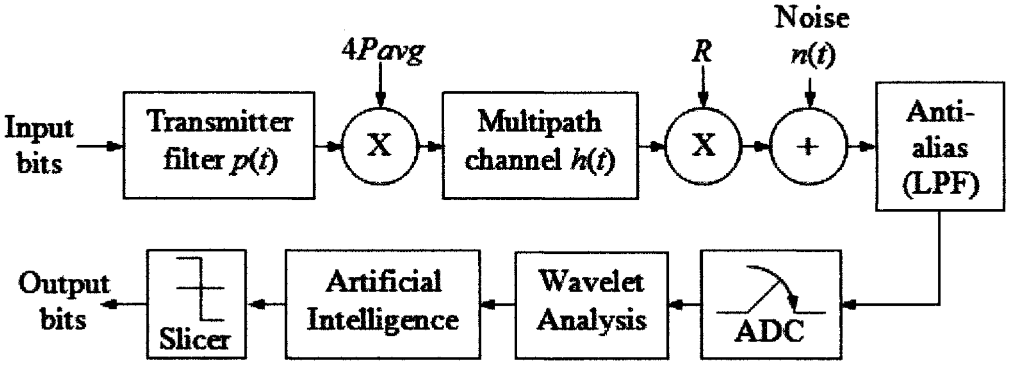


Fig. 4.16: OOK RZ simulation model schematic employing a Wavelet-AI detection scheme

The neural network is trained with 1500 known samples prior to classification of unknown data. The network classifies bit No. 3 of the 5 bit window as binary '1' or '0'. Again the ADC shown is notional since the entire system is discrete with a sample interval of 1 ns, providing 10 sample points per bit.

The difficulties with network dimensionality, choice of transfer function and number of training samples remain. The number of system parameters is further complicated by the inclusion of Wavelet type and number of scales to include. It is understood from literature that rigorous methods for parameter selection for either the neural network or the wavelet analysis do not exist and present a further research opportunity. The neural network is therefore as described earlier and the wavelet scales are chosen visually from a noise free coefficient, time-scale plot to include scales relating to all features on a best-guess basis. This is somewhat of a scatter-gun approach with a great likelihood redundancy in scale selection.

Table 4.3: Coefficient structure

Window 1	**Bit 1**	**Bit 2**	**Bit 3**	**Bit l 4**	**Bit 5**
	Scale 2				
	Scale 4				
	Scale 6				
	⋮				
	Scale 74				
Window 2	**Bit 2**	**Bit 3**	**Bit l 4**	**Bit 5**	**Bit 6**
	Scale 2				
	Scale 4				
	Scale 6				
	⋮				
	Scale 74				
Window 3	**Bit 3**	**Bit 4**	**Bit 5**	**Bit 6**	**Bit 7**
	Scales 2:74				

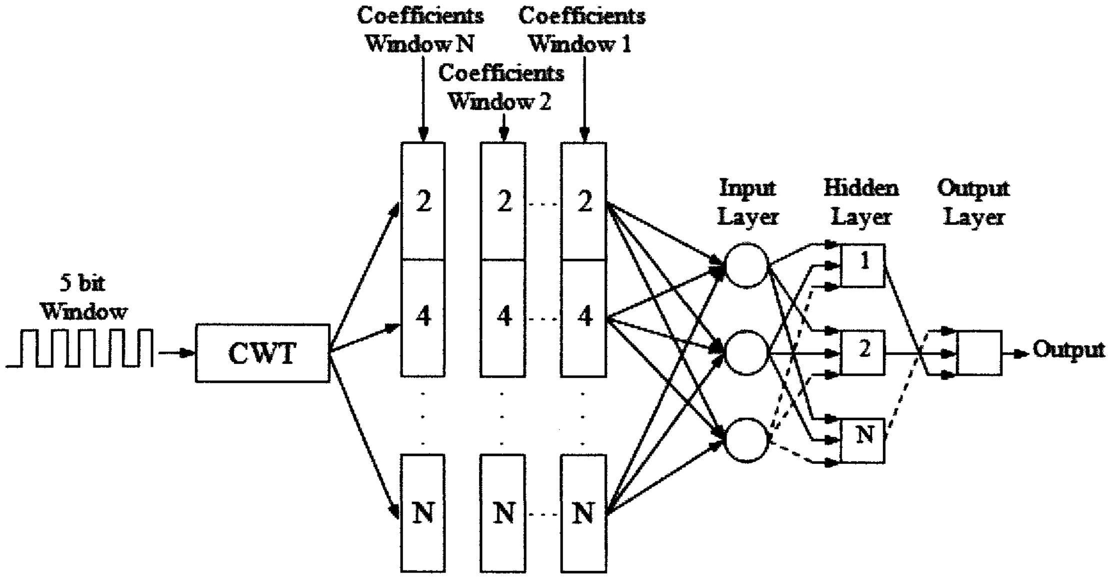


Fig. 4.17: Schematic of the wavelet and AI section information flow

The program flow chart for the simulation model is shown in Fig. 4.18 and the program listing on the accompanying CD. The CWT section has replaced the filtering and down-sampling elements relating to the previous sections. The results of the simulations are illustrated in Fig.4.19. It can be seen that the performance of the Wavelet-AI system is broadly similar to that of direct neural network detection incorporating a pre-filter as discussed in section 4.3.1. It could be argued that there is less variation between the lowest and highest noise training figure, however, it is less than conclusive from these results.

Simulations with the model parameters indicated and available computing equipment proved extremely expensive in terms of execution time. A single simulation designed to provide a result at a BER figure of 10^{-5} would take several days to complete. In addition, as well as increasing simulation time, increasing model parameter dimensionality increased the likelihood of simulation failures due to unachievable memory requirements.

The aim of this research is to investigate the viability of the Wavelet-AI architecture for use in the diffuse indoor environment. Whilst definitive architecture optimisation is beyond the scope of this work it is imperative that some model parameter simplification is pursued in order that the objectives of the work are met within the time limits required. These simplifications are considered in the next section.

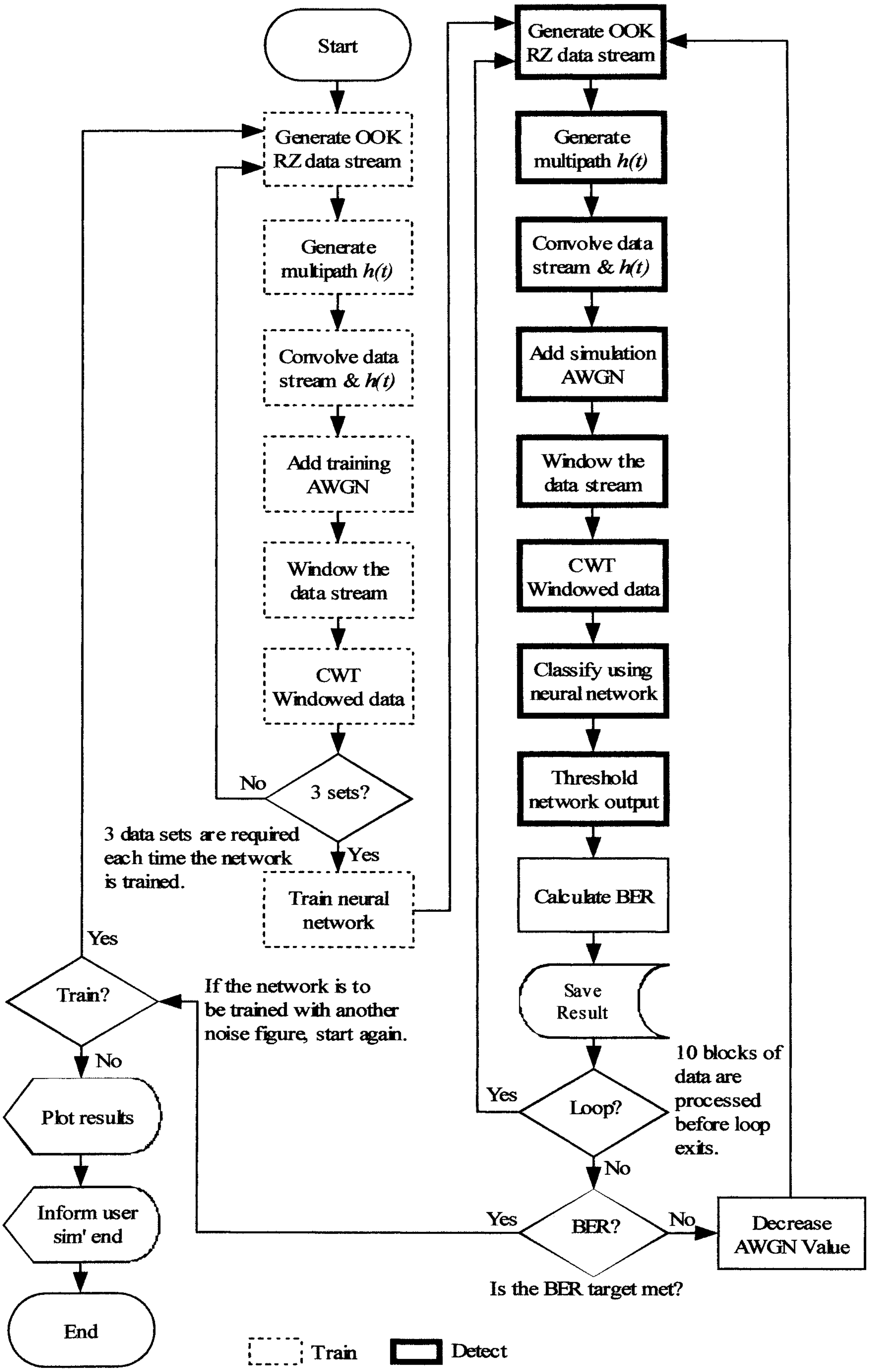

Start
Generate OOK RZ data stream
Generate multipath $h(t)$
Convolve data stream & $h(t)$
Add training AWGN
Window the data stream
CWT Windowed data
3 sets?
No
Yes
3 data sets are required each time the network is trained.
Train neural network
Yes
Train?
No
If the network is to be trained with another noise figure, start again.
Plot results
Inform user sim' end
End
Generate OOK RZ data stream
Generate multipath $h(t)$
Convolve data stream & $h(t)$
Add simulation AWGN
Window the data stream
CWT Windowed data
Classify using neural network
Threshold network output
Calculate BER
Save Result
10 blocks of data are processed before loop exits.
Loop?
Yes
No
BER?
Yes
No
Decrease AWGN Value
Is the BER target met?
Train
Detect

Fig. 4.18: Program flow chart for simulation model shown in Fig. 4.16

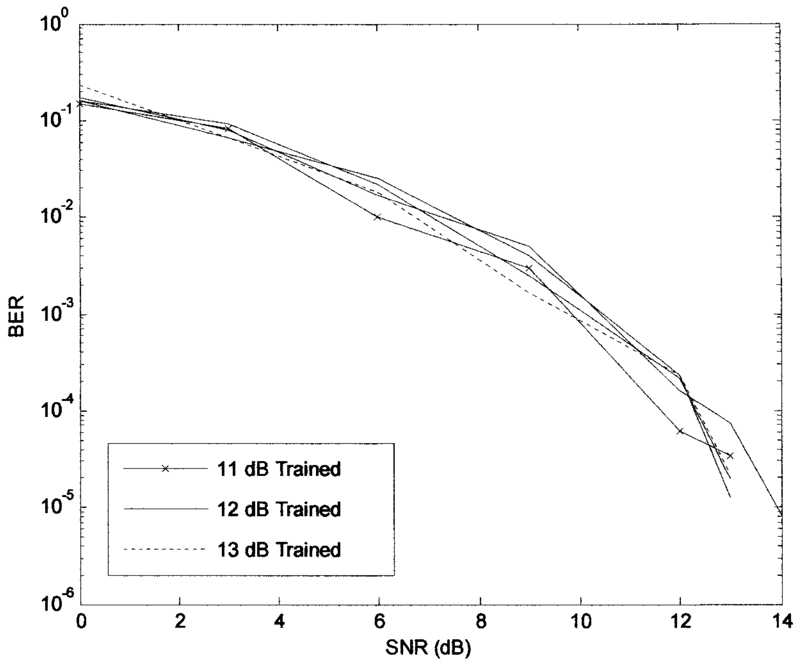


Fig 4.19: BER against SNR for Wavelet-AI detection of 100Mb/s OOK RZ, trained at 11 dB, 12 dB and 13 dB SNRs

4.4.2 Wavelet-AI Receiver Simplification

The next few sections will investigate model simplifications in order to reduce computational burden. The areas to be investigated are window size, multiple noise training, scale reduction and neural network simplification.

4.4.2.1 Wavelet AI Receiver Window Sizes

Window size and the number of scales employed directly affects the computational burden imposed by the CWT process. In this section the effects of window size on Wavelet AI receiver performance is explored. The simulation model is very similar to that of section 4.4.1 and the schematic is not repeated. However, the model employs a network trained with 1500 symbols with varying noise values added, provinding SNRs as follows: 12 dB in the first 3000 symbols, 6 dB in the second 3000 symbols and 15 dB in the third 3000 symbols, 9 dB in the forth 3000 symbols and 18 dB in the last 3000 symbols. This strategy is aimed at reducing the training burden of the simulations by finding a near optimum BER for a given SNR using only one training set. However, as all the simulations in this section adopt the same strategy, a like for like comparisson of results is still valid.

Briefly to recap, the model emplys the CWT using the 'Sym2' wavelet with scales from 2 to 74 in steps of 2. The neural network is 100 input / 1 output neuron with log-sigmoid, and linear transfer functions repectfully; the training algorithm being conjugate gradient with Powell/Beale restarts. The network is trained with 1500 samples before classification of unknown data begins. The multipath distortion induced normalised delay spread remains unchanged from previous sections. The program flow-chart for the simulation model is shown in Fig. 4.20. The loop around the training section of this flow-chart is absent and the AWGN process depicts a (X5) to indicate multiple SNR figures in the training signal data stream. No program listings are given for simulation models in the remainder of this chapter as the majority of the core code remains as for Fig. 4.16.

The results of the simulations are shown in Fig. 4.21. It is immediately apparent from observation that a single bit window is significantly inferior to multiple windows. This may indicate that the neural network is indeed extracting information from previous and following bits to make a decision. However, employing multiple bit windows over the range simulated show little significant variation in performance. Given these results, all later simulations for OOK RZ will be based on a 3-bit window with centre bit detection to reduce computational effort.

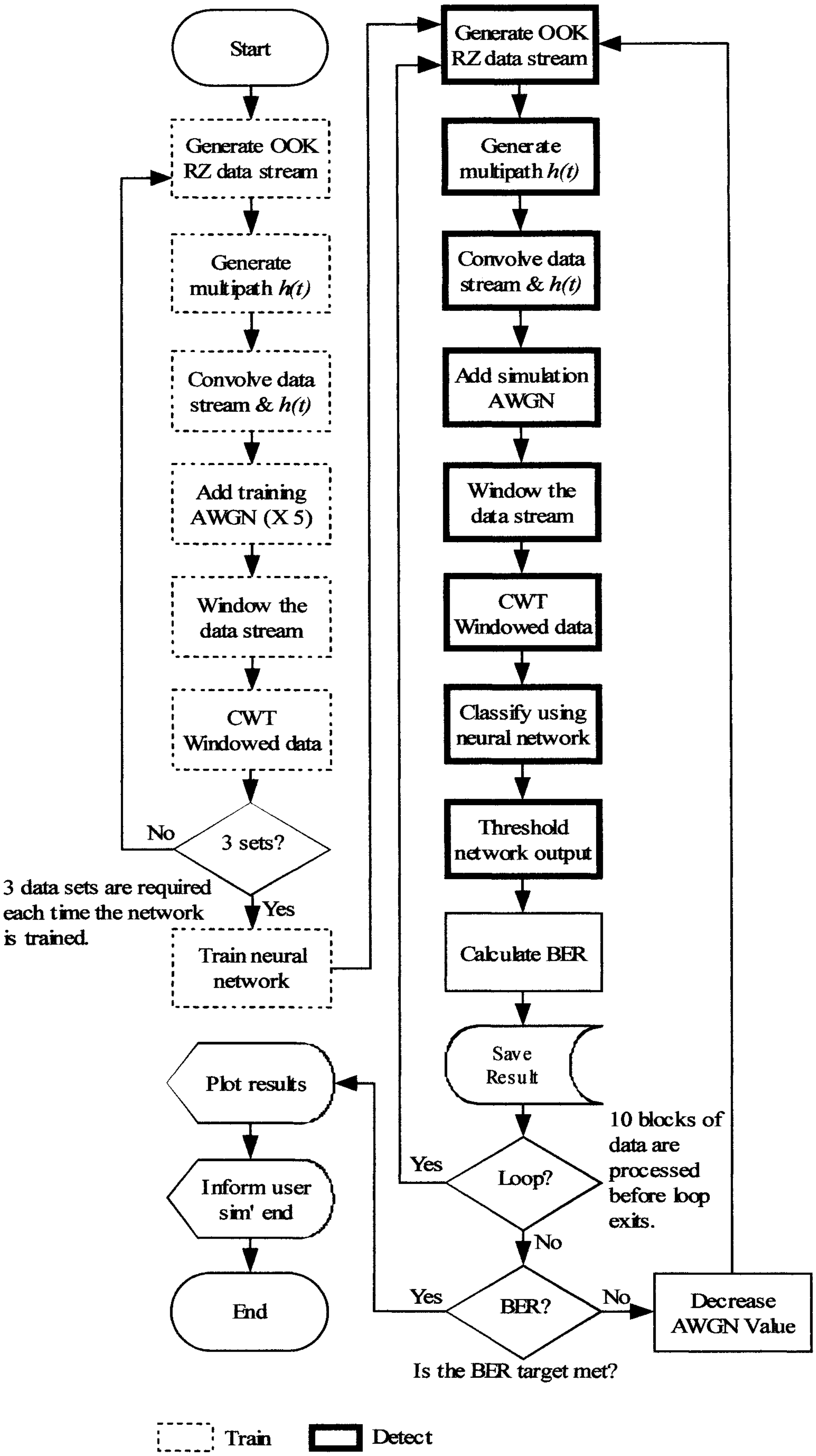


Fig. 4.20: Program flow chart for simulation model with multiple 'window' sizes

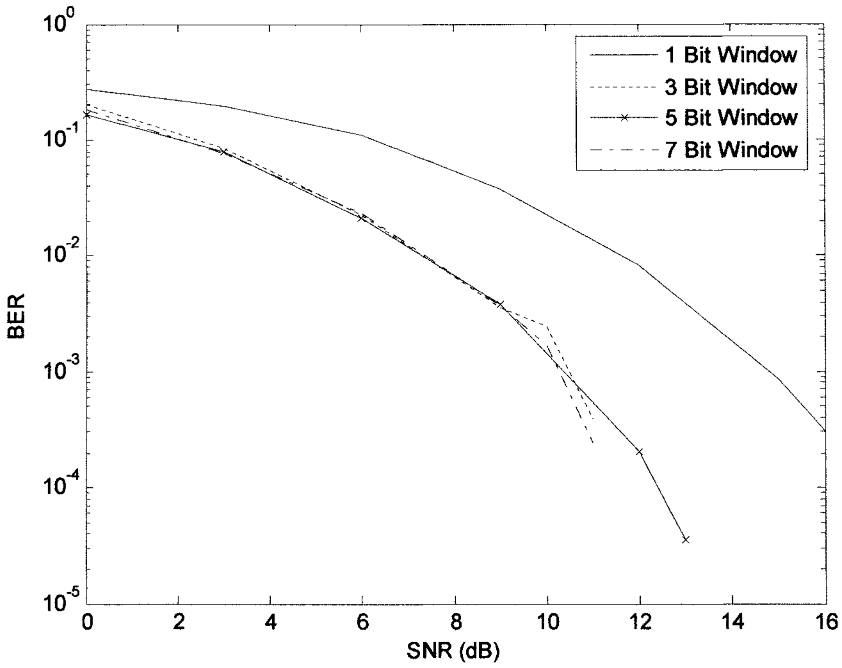


Fig. 4.21: BER against SNR for Wavelet-AI detection of 100Mb/s OOK RZ with different window sizes and single training session

4.4.2.2 Wavelet AI Receiver Scale Reduction

A further impact on the computational burden of the CWT section is that of scale selection. It is a relatively simple process to visually select the scales that cover as many signal features as possible; under the correct conditions this would allow perfect synthesis of the original signal. Such a brute force method would give confidence that the neural network section was passed near optimum time-scale information for classification purposes. However, such methods impose two costs, that of CWT complexity and the subsequent level of numerical calculations required to be undertaken by the neural network.

This section briefly investigtes the effect of reducing scales to those which visually encompass the major features of the signal as seen on a time-scale plot. These are scales

2 and 7, scales 2, 7 and 14 and scales 2, 7 and 40. Again the simulation model is as shown in Fig. 4.16, and as discussed in the previous section; however, here it will employ a 3 bit window only. The program flow-chart for the simulation model remains unchanged from Fig. 4.20, as all the processing elements and training strategies remain unchanged save for the scales used when employing the CWT.

When compared with Fig. 4.19 and Fig. 4.21, the simulation results shown in Fig. 4.22 indicate that considerable reduction in the number of scales can be made without significant reduction in system performance. In comparison to the previous section the scales have been reduced in number from 38 to 3, representing a major saving in computational time.

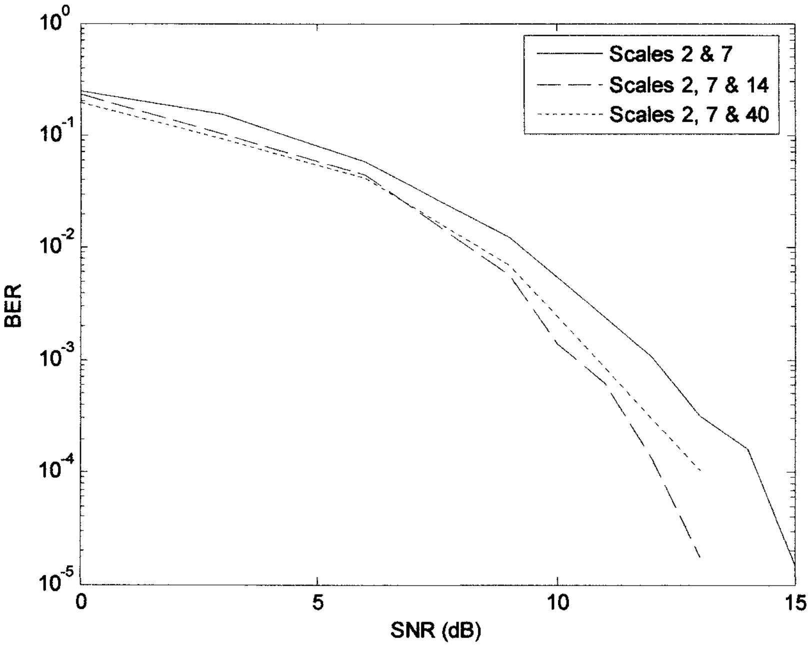


Fig. 4.22: BER against SNR for Wavelet-AI detection of 100Mb/s OOK RZ with 3 bit window, reduced scale inputs and single training session

4.4.2.3 Reduced Neurons

As alluded to in earlier sections a further contributor to computational complexity is the size of the neural network. Given that all scales from the CWT section are passed to all neurons in the input layer further redcution in computational burden can be made by reducing the size of the network. As previously indicated, no definitive method exists to determine the optimum size of a feed-forward back-propagation neural network and 'network pruning' is sometimes employed.

In this section a simulation model derived from the previous two simplifications is used. The model utilises a 3-bit window where the CWT section processes scales of 2, 7 and 14. However, the schematic and program flow-charts remain as per section 4.4.2.1 and so will not be repeated. Four simulations were executed with the only variation being the size of the first processing layer, or 'hidden layer' in the neural network. Simulation runs for a first processing layer of 30, 15, 5 and 1 neurons were made.

The results, shown in Fig. 4.23, indicate little variation between 30 first layer neurons and 5 first layer neurons. However, the results from 1 first layer neuron model show a consistently worse BER performance than other models, whilst the 15 neuron model shows some behaviour anomalies from a BER of approximately 10^{-3}. Whilst not exhaustive the results indicate that a substantial reduction in network complexity can be made without excessively compromising receiver performance providing a further minimisation of computational effort.

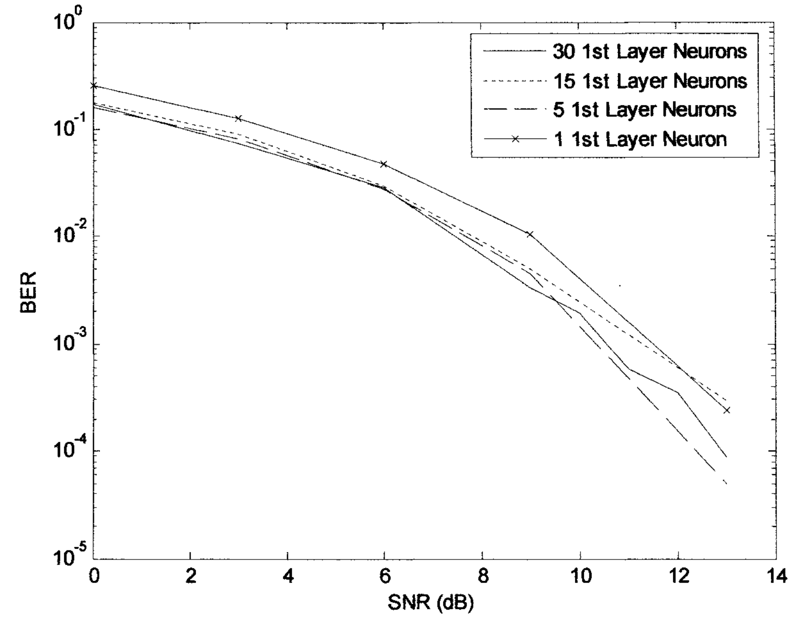


Fig. 4.23: BER against SNR for Wavelet-AI detection of 100Mb/s OOK RZ with 3 bit window, scales of 2, 7 & 14, reduced neurons and single training session

4.5 Summary

In this chapter Artificial Intelligence and some of the more popular AI schemes that could arguably be employed in a Wavelet-AI receiver have been introduced. The chapter then focussed on feed-forward back-propagation networks and discussed simulation models for direct detection including pre-filtering and down-sampling indicating that a neural network used as a receiver benefits from some form of pre-processing but that down-sampling yields a less stable performance.

The chapter then considered the CWT as a pre-processing element of the neural network and considered the performance of such architecture. Performance issues were discussed in relation to computational burden and subsequently prohibitively long simulation times. The remainder of the chapter then focused on simplifying the architecture to improve these difficulties without sacrificing performance to achieve realistic simulation periods with available hardware. Whilst not exhaustive and fully conclusive, the chapter indicates that substantial reduction of architecture complexity in terms of parameters such as window size, scales and network size does not seem to adversely affect the receiver's performance for the modulation type and bit rates considered. A summary of the simulation models discussed in this chapter are given in Table 4.4. Due to the prohibitive time impact comprehensive simulations defining optimum model simplifications cannot be afforded in this work, but provide significant scope for further investigations.

The reduced complexity architecture described will be carried forward to later work when multipath distortion and artificial light interference will be considered in more depth. It should be noted that the selection of schema considered and simulated represents a fraction of those that could be implemented with an architecture as flexible as Wavelet-AI. Other variations including decision feed-back, word detection, coding and majority vote to name a few may all be valid additions to those already implemented and could represent a rich vein of research not only for the IR domain, but also for the RF environment.

Table 4.4: Simulation model summary.

	Receiver type	**Comments**
1	Direct Neural Network Detection.	Susceptible to relatively wide variations in performance for correspondingly small vairations in noise.
2	Direct Neural Network Detection and LPF.	Less susceptible to performance variations in comparrison to receiver 1.
3	Direct Neural Network Detection with LPF.and down-sampling	Reduced complexity and lower sampling rate
4	Wavelet AI receiver	First Wavelt-AI receiver, lower performance against noise variation compared to 1, 2 & 3. Much more computationally expensive.
5	Wavelet AI Receiver with reduced window	Similar to 4, but with reduced computational burden.
6	Wavelet AI Receiver with reduced window and reduced scales	Similar to 4, but with further reduced computational burden.
7	Wavelet AI receiver with reduced window, reduced scales and reduced neural network	Similar to 6, but with further reduced computational burden.

Chapter 5

The Effect of Ambient Light Sources on Link Performance

5.1 Introduction

Indoor infrared communications systems suffer performance degradation due to the effects of intense interfering ambient light from both natural and artificial sources. These sources include office and domestic fluorescent and incandescent lamps. The average power of background radiation is modelled as white, Gaussian and independent of the received signal [27]. In addition, artificial light sources induce periodic interference that contains harmonics of the mains or switching frequencies of their integral electronic circuits. These periodic interfering signals have the potential to seriously impair link performance. The most degrading of these is the interference generated from a fluorescent light driven by an electronic ballast as the interfering signal contains harmonics of the switching frequency that can extend into the MHz range [76, 79].

A common mitigation technique is to use high pass filtering (HPF) to remove the interfering signal, this may be achieved by tuning the AC coupling between successive amplifier stages [76]. Unfortunately, although the filtering may be effective at attenuating the interference, it introduces a form of ISI known as baseline wander which is particularly severe for base-band modulation techniques which contain a significant amount of power at low frequencies. The higher the cut-on frequency of the HPF the greater is the attenuation of the interfering signal, but the baseline wander increases correspondingly. Therefore, there is a trade off between the extent of interference rejection and the attenuation of baseline wander [80, 171].

There are a number of significant factors that affect the performance of an indoor infrared system with a virtually infinite permutation of window size, direction, types and quantities of artificial light sources. Not to mention the orientation and spacing of the transmitter receiver pair. To provide some consistency with existing data, this chapter considers two specific cases published by Moreira *et al* [76]. The cases are as follows:

Case 1: No Interference:

Natural (Sun) ambient light, generating an average photocurrent I_B of 200 μA

Case 2: Fluorescent light interference:

Natural light as in case 1, plus a fluorescent light driven by electronic ballast, generating a photocurrent of 2 μA, giving a total background photocurrent of 202 μA.

The major differences between electronic ballast driven florescent lamps is the switching frequency used, which are usually in the range of 20 – 40 kHz, and the relative strengths of the high and low frequency components [74, 75]. The emissions from lamps driven by electronic ballast are not synchronised with the mains frequency and therefore the worst case is represented by a one or more lamps driven by a single ballast as this will have the greatest amplitude and slope [79]. The interfering fluorescent light signal used in the simulations presented here is generated using the model developed by Moreira *et al.* [74, 75].

In this chapter the performance of OOK and PPM modulation schemes is evaluated when detected by:

a. A matched filter based receiver

b. A match filter based receiver with high pass filtering

c. A Wavelet–AI based receiver.

Throughout this chapter, all power requirements are normalised to the average optical power required by a particular modulation scheme to achieve a BER as defined in the relevant section. The remainder of this chapter is organised as follows: Section 5.2 describes the model used to generate the fluorescent light interference signal; Section

5.3 investigates the effect of fluorescent light interference without electrical high pass filtering; Section 5.4 investigates the effect of baseline wander without fluorescent light interference; Section 5.5 investigates the effect of fluorescent light interference with electrical high pass filtering. Finally, the chapter is summarised in section 5.6.

5.2 Fluorescent Lamp Model

Based on extensive measurements of a variety of fluorescent lamps driven by electronic ballasts, Moreira *et al* produced a model to describe the interference signal [74, 75]. All measurements were taken using an optical long-pass filter, and consequently, the values used in the model reflect this particular case. The interference signal is comprised of a low frequency component, similar to that of a fluorescent lamp driven by a conventional ballast, and a high frequency component, which is generated by the switching circuit of the electronic ballast. Thus, the zero mean periodic component of the photocurrent $m_{fl}(t)$ is given as [74, 75]:

$$m_{fl}(t) = m_{low}(t) + m_{high}(t) . \tag{5.1}$$

The low frequency component may be expressed as [74, 75]:

$$m_{low}(t) = \frac{I_B}{A_1} \sum_{i=1}^{20} \left[\Phi_i \cos\left(2\pi(100i - 50)t + \varphi_i\right) + \Psi_i \cos\left(2\pi \cdot 100it + \phi_i\right) \right], \tag{5.2}$$

where Φ_i and Ψ_i are the amplitudes of the odd and even harmonics of 50 Hz, respectively, given by [74, 75]:

$$\Phi_i = 10^{(-13.1 \cdot \ln(100i-50)+27.1)/20} \qquad 1 \le i \le 20, \qquad (5.3)$$
$$\Psi_i = 10^{(-20.8 \cdot \ln(100i)+92.4)/20} \qquad 1 \le i \le 20, \qquad (5.4)$$

and φ_i and ϕ_i are the phase of the odd and even harmonics of 50 Hz, respectively, as given in Table 5.1. A_1 is the constant that relates the interference amplitude with I_B, taken to be 5.9 in this analysis, and I_B is the average photocurrent generated by the fluorescent lamp, taken to be 2 μA.

Table 5.1: Low frequency component phase values.

i	φ_i (rad)	ϕ_i (rad)	i	φ_i (rad)	ϕ_i (rad)
1	4.65	0	11	1.26	6.00
2	2.86	0.08	12	1.29	6.17
3	5.43	6.00	13	1.28	5.69
4	3.90	5.31	14	0.63	5.37
5	2.00	2.27	15	6.06	4.00
6	5.98	5.70	16	5.49	3.69
7	2.38	2.07	17	4.45	1.86
8	4.35	3.44	18	3.24	1.38
9	5.87	5.01	19	2.07	5.91
10	0.70	6.01	20	0.87	4.88

The high frequency component may be expressed as [74, 75]:

$$m_{high}(t) = \frac{I_B}{A_2} \sum_{j=1}^{22} \Gamma_j \cos\left(2\pi f_{high} jt + \theta_j\right), \qquad (5.5)$$

where Γ_j and θ_j are the amplitude and phase of the harmonics, f_{high} is the electronic ballast switching frequency, and A_2 is the constant that relates the interference amplitude to I_B, taken to be 1 in this analysis. The parameters f_{high}, Γ_j and θ_j depend on the type of electronic ballast, and vary from one manufacturer to the next. In this work, f_{high} is taken to be 37.5 kHz and the parameter values given in Table 5.2 represent a particular case with this switching frequency.

Table 5.2: High frequency component amplitude and phase values.

j	Γ_j (dB)	θ_j (rad)	j	Γ_j (dB)	θ_j (rad)
1	-22.2	5.09	12	-39.3	3.55
2	0	0	14	-42.7	4.15
4	-11.5	2.37	16	-46.4	1.64
6	-30.0	5.86	18	-48.1	4.51
8	-33.9	2.04	20	-53.1	3.55
10	-35.3	2.75	22	-54.9	1.78

For the chosen switching frequency of 37.5 kHz, there are 750 cycles of the high frequency component per cycle of the low frequency component. With I_B set at 2 μA, one complete cycle of the low frequency component and three complete high frequency component cycles of the interference photocurrent are shown in Fig. 5.1 and Fig. 5.2 respectfully [72].

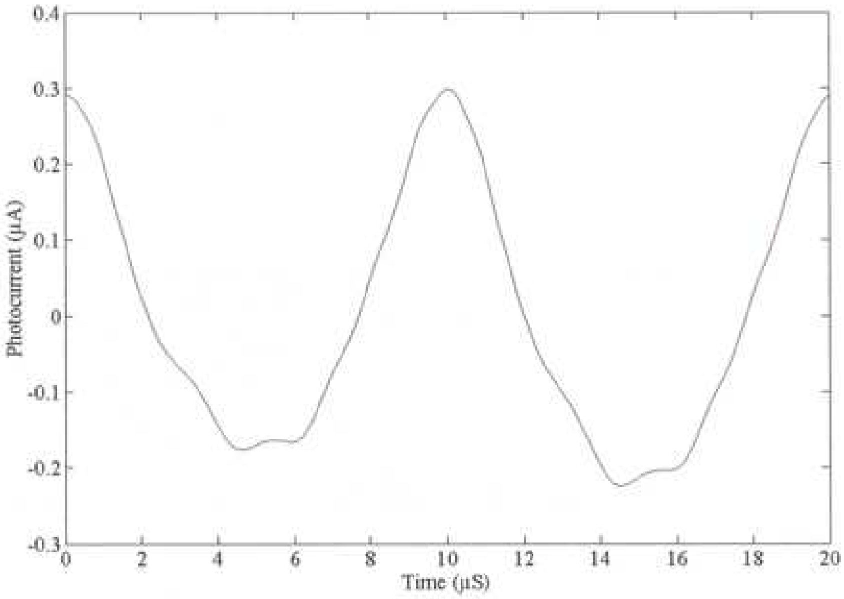


Fig. 5.1: Low frequency interference component

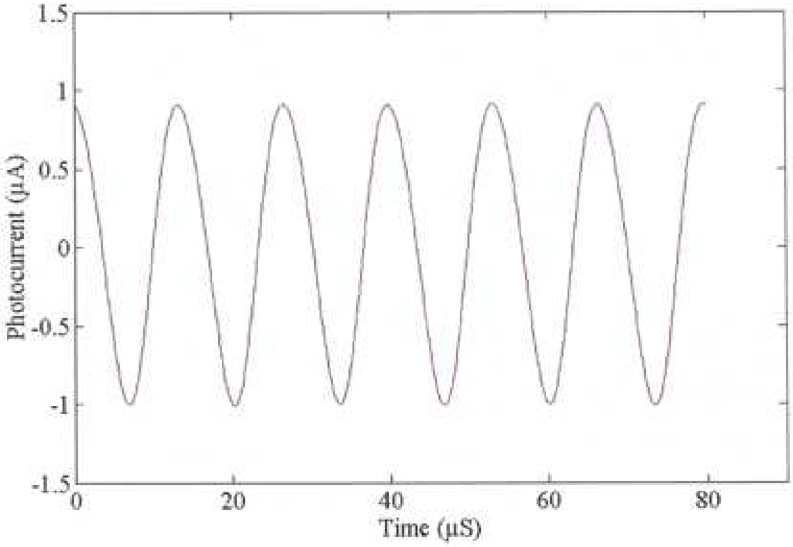


Fig. 5.2: High frequency interference component

5.3 The Effect of Fluorescent Light Interference without Electrical High-Pass Filtering

In this section we consider the type of signal as shown in Fig. 5.3, OOK RZ ($\gamma = 0.5$) at various data rates, distorted by the inclusion of fluorescent light source interference as described in section 5.2. No form of filtering or other compensating techniques will be employed to remove the interference.

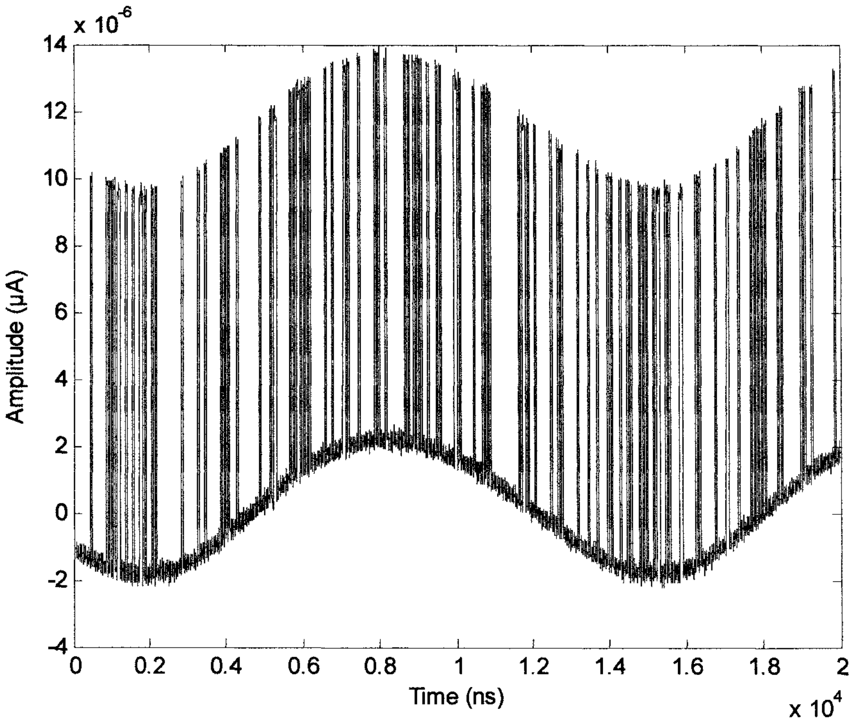


Fig. 5.3: OOK RZ signal with fluorescent light interference

5.3.1 OOK RZ (Matched Filter Receiver)

A block diagram of the OOK matched filter system under consideration is shown in Fig. 5.4. The flowchart for the simulation model is shown in Fig.5.5, and the program listing on the accompanying CD. A program listing for the fluorescent light model is also given on the CD.

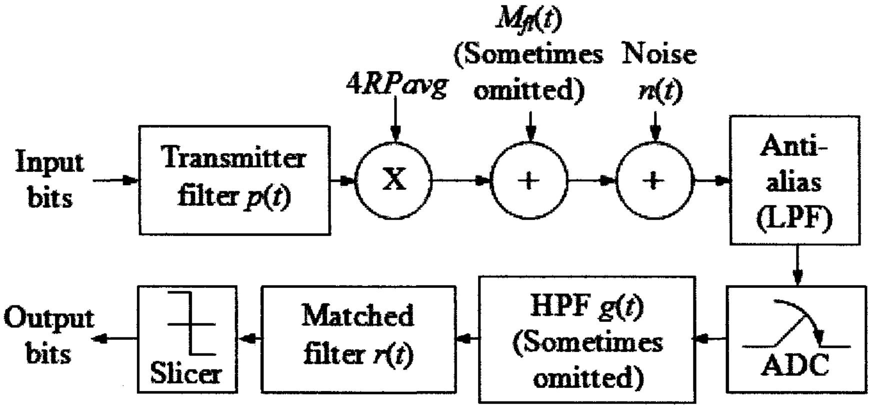


Fig. 5.4: OOK RZ simulation scheme employing HPF and matched filter detection

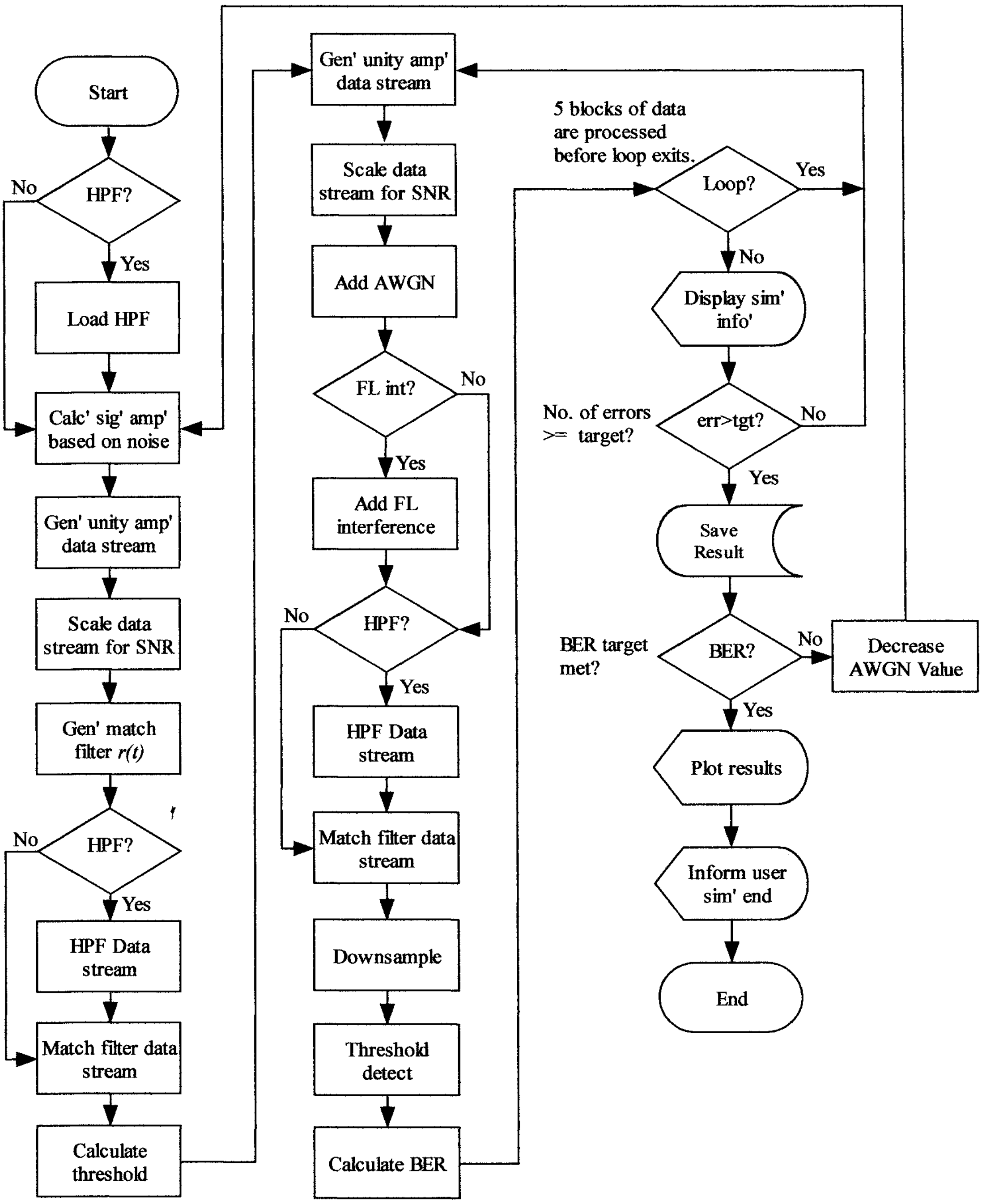


Fig. 5.5: Flowchart for simulation model shown in Fig.5.4

The transmitter filter has a unit-amplitude rectangular impulse response *p*(*t*), with a duration of $T_b/2$. The output of the transmitter filter is scaled by the peak detected signal photocurrent $4RP_{avg}$, where *R* is the photodetector responsivity, taken to be unity in this chapter, and P_{avg} is the average received optical signal power. The fluorescent light induced photocurrent $m_{fl}(t)$, is then added to the signal, along with the signal independent shot noise *n*(*t*), which is modelled as white and Gaussian, with a double-sided power spectral density $N_o/2$, given as [172]:

$$N_0/2 = qI_B\,, \tag{5.6}$$

Where q is the electron charge and I_B is the average photocurrent generated by the background light, which is taken as 202 μA. The signal is analogue anti-alias filtered where the cut off frequency is related to the sampling frequency. In this section, the Finite Impulse Response (FIR) HPF is omitted and the detected signal is passed directly to a FIR unit energy filter with an impulse response *r*(*t*), which is matched to *p*(*t*).

The output signal is given by [108]:

$$y(i) = \sum_{k=0}^{N-1} r(k)x(i-k) \tag{5.7}$$

Where *x*(*i*) are the signal samples and *r(k)* are the matched filter coefficients, with k = kT_s and T_s being the sample interval.

The filter output is sampled at the end of each bit period, and a one or zero is assigned depending on whether the signal is above or below the threshold level at the sampling instant. The threshold level is set midway between expected one and zero levels.

The output of the matched filter due to the fluorescent light interference signal, sampled at the end of each bit period, is given as [79]:

$$m_k = m_{fl}(t) \otimes r(t)\Big|_{t=kT_b}\,, \tag{5.8}$$

Where the symbol ⊗ denotes convolution. Evidently, by considering every bit over a 20 ms interval (i.e. one complete cycle of $m_{fl}(t)$) and averaging, the probability of bit error is given as [79]:

$$P_{e,bit,OOK} = \frac{1}{2M}\sum_{k=1}^{M}\left[Q\left(\frac{RP_{avg}\sqrt{T_b}+m_k}{\sqrt{N_o/2}}\right)+Q\left(\frac{RP_{avg}\sqrt{T_b}-m_k}{\sqrt{N_o/2}}\right)\right], \qquad (5.9)$$

Where M is the total number of bits over a 20 ms interval.

However, the simulation models used here are different, and in order to achieve viable simulation run times, a number of assumptions are required, as described below:

(i) The models used for simulation are based around digital receiver architecture. The number of samples per bit period affects the performance of digital FIR filters as used in this model; however this is also true of the Wavelet-AI receiver. Further to this, increasing the number of samples increases the computational burden resulting in long simulation times. In order to ensure consistency through a range of comparable simulations the models have used a fixed sampling rate for any given set of simulations to ensure like for like comparisons. For this section the sampling interval is 1 ns.

(ii) The inclusion of an anti-alias filter limits the noise power passed to the remainder of the filter section due to the finite bandwidth of the filter. This is known as band limited white noise [173]:

$$P_n = \frac{1}{2\pi}\int_{-2\pi B}^{2\pi B}(N_o/2)d\omega \qquad (5.10)$$

Where P_n is the noise power (W), $N_o/2$ is the power spectral density and B is the bandwidth of the filter. Given the Nyquist theorem and a sample interval of 1 ns the bandwidth of the filter would be 500 MHz. To avoid the computational complication of a further filtering stage the filter has been omitted from the simulation models and the noise power is calculated from the product of the bandwidth and the power spectral density. The amplitude of the data signal is then varied to give specific values of SNR (dB).

(iii) With a chosen switching frequency of 37.5 kHz there are 750 cycles of high frequency component per cycle of the low frequency component. In some simulations [72] the low frequency component is assumed to be an offset that is constant over the duration of one high frequency cycle and a suitable simulations strategy is employed to take account of these effects.

In all simulations in this chapter the actual simulated low and high frequency interference is added to the wanted data signal. However, the duration of the simulated signal can be much shorter than on complete cycle of the low frequency component. Each of the simulations therefore starts at some random point during the low frequency cycle.

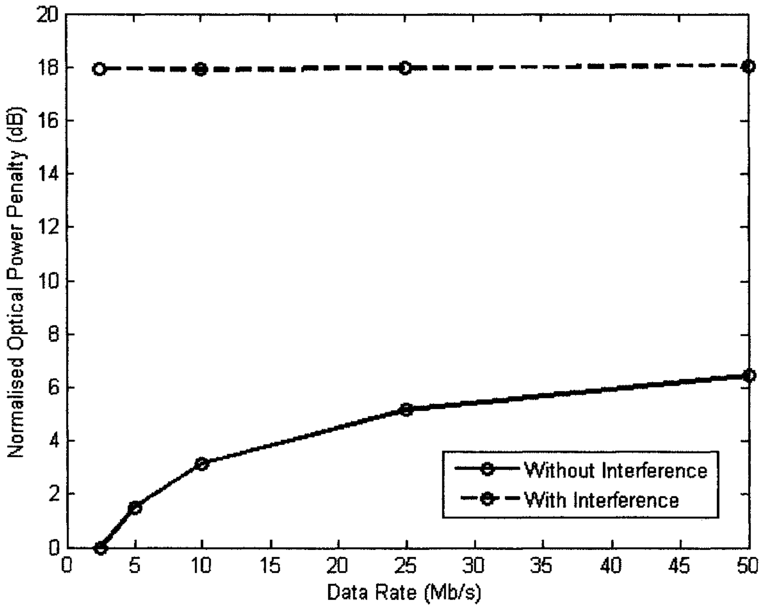


Fig. 5.6: Normalised optical power penalty versus data rate for OOK RZ with and without fluorescent light interference for a matched filter receiver at a BER of 10^{-5}

Fig. 5.6 shows data rate versus normalised optical power penalty for OOK RZ with and without fluorescent light interference. The power penalty is normalised to the power requirement for transmitting a 2.5 Mb/s signal over a perfect channel for a BER of 10^{-5}. It can be seen from the figure that the power requirement for a signal subjected to fluorescent light interference is very similar for all data rates considered. For 2.5 Mb/s the penalty is approximately 18 dB, whilst at 50 Mb/s the penalty is reduced to approximately 12 dB over the case with no interference.

5.3.2 OOK RZ (Wavelet AI receiver)

A block diagram of the OOK Wavelet-AI receiver is shown in Fig. 5.7. The flowchart for the simulation model is shown in Fig. 5.8, and the program listing in appendix A

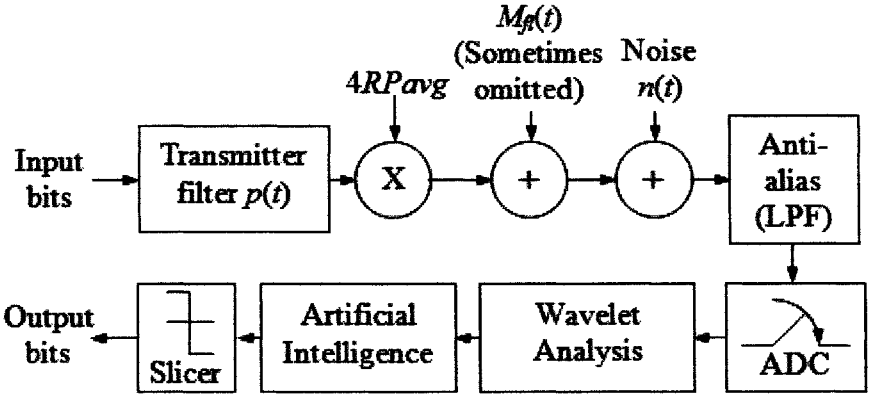


Fig. 5.7: Block diagram of the OOK RZ (Wavelet-AI) system

It can be seen from the figure that from a schematic point of view the model is similar to that of the matched filter system; the wavelet section being in the same position as the HPF of Fig. 5.4, and the artificial intelligence section being in the same position as the matched filter.

However, the model operates in an entirely different manner. The wavelet section collects the incoming signal samples and sorts them in to overlapping 3 bit samples as outlined in Table 5.3. and in a similar manner to that fully described in chapter 4.

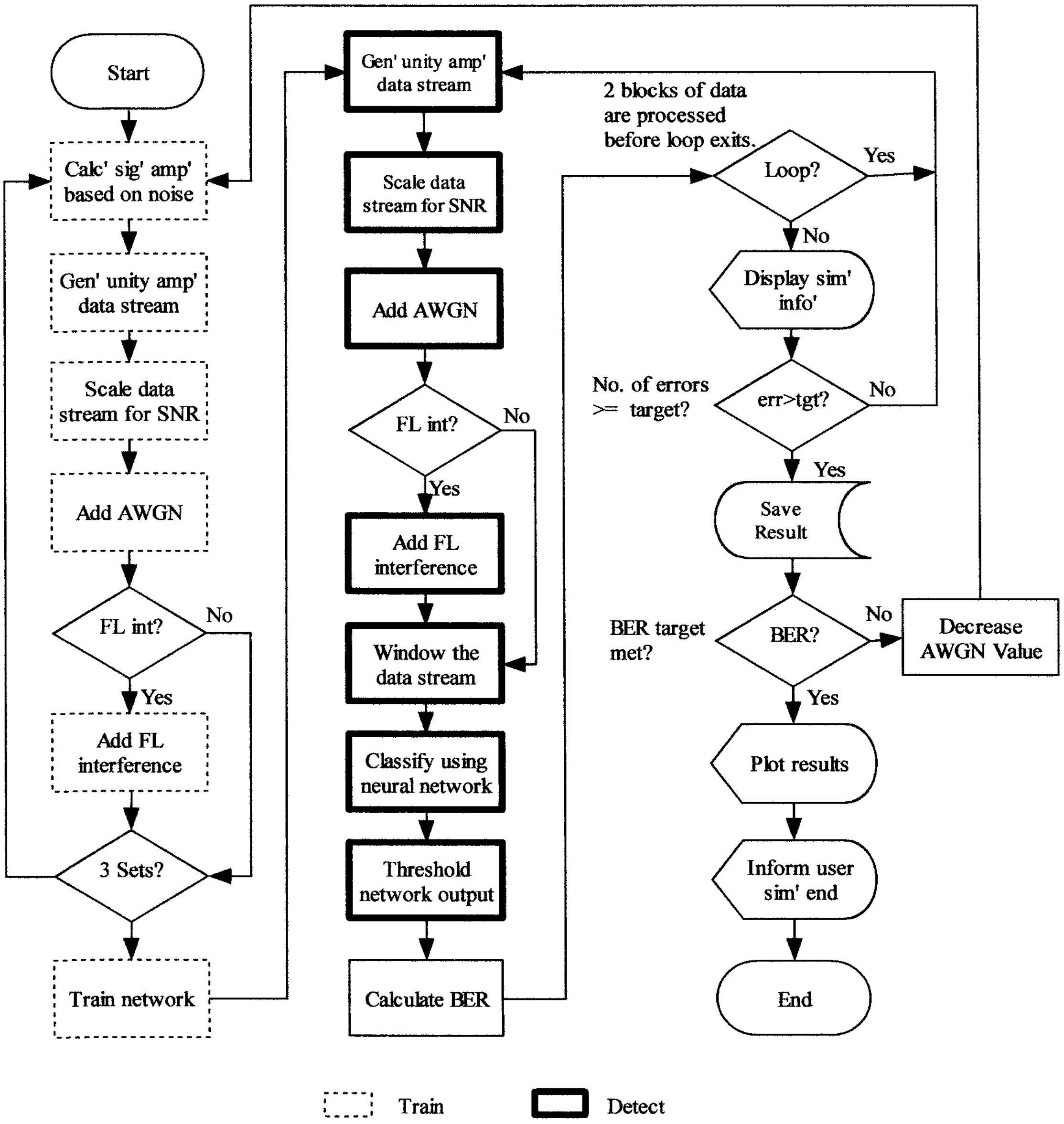


Fig. 5.8: Flowchart for simulation model shown in Fig. 5.7

Table 5.3: Sample 'Window' Structure.

	Position 1	**Position 2**	**Position 3**
Sample 1 (Window 1)	Bit 1	Bit 2	Bit 3
Sample 2 (Window 2)	Bit 2	Bit 3	Bit 4
Sample 3 (Window 3)	Bit 3	Bit 4	Bit 5
Sample N (Window *N*)	Bit *N*-2	Bit *N*-1	Bit *N*

It is the centre position of the window (position 2) that will ultimately be detected by the receiver. Therefore in this case bits 1 and *N* are redundant and not detected. It is therefore essential that any data stream be preceded and followed by an arbitrary mark or space providing a total of *N*-2 data bits. From now on this will be referred to as windowing the data, this being a 3 bit window as shown in Fig. 5.9.

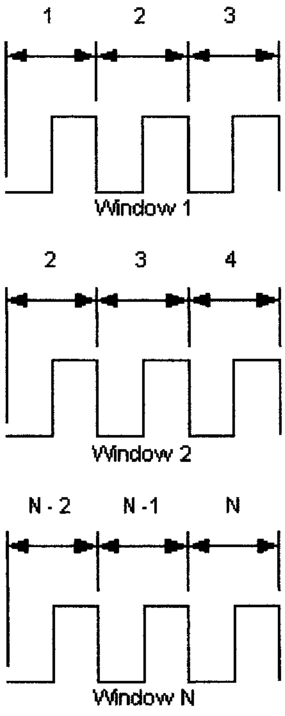


Fig. 5.9: 3 bit window

The CWT detailed in chapter 3 is used to obtain the wavelet coefficients from each sample set where the mother wavelet used is the 'Symlet 2'. The CWT would potentially provide all wavelet coefficients up to some arbitrary maximum, producing large numbers of coefficients. This would greatly increase computational burden and ultimately slow the detection process to a point of being unfeasible even for simulation purposes. The changes in the value of the coefficients between one scale and the next are potentially very small and in the majority of cases can be considered redundant, where the redundancy could span many scales. The wavelet section calculates the coefficients of some pre-determined scales that are considered to contain significant information. Unfortunately, a technique for detecting redundancy, or more importantly significance of a particular scale needs to be developed. In this work scales are selected by visual inspection of 2 and 3 dimensional time-scale-amplitude plots as shown in previous chapters. The scales chosen for simulations in this section are detailed in Table 5.4. It is sometimes useful to note the difference between plots of the noise free and noisy signal, this visual technique can afford the opportunity to select scales that are less affected by interfering sources.

Table 5.4: Coefficients.

Data Rate (Mb/s)	Scales
2.5	250, 750, 1000, 1250, 1500
5	45, 160, 320, 550
10	100, 230, 380, 780
25	70, 150, 250, 300, 550
50	16, 32, 48, 64

For simulation purposes the wavelet coefficients are arranged into a 2 dimensional array or matrix based on the window sample and number of coefficients, where the rows contain the coefficients of the scales and the columns refer to the window as shown in Fig. 5.10. For example a data rate of 2.5 Mb/s and sample interval of 1 ns would produce 400 coefficients for each scale used. Each column is passed to the Artificial Intelligence section for subsequent classification.

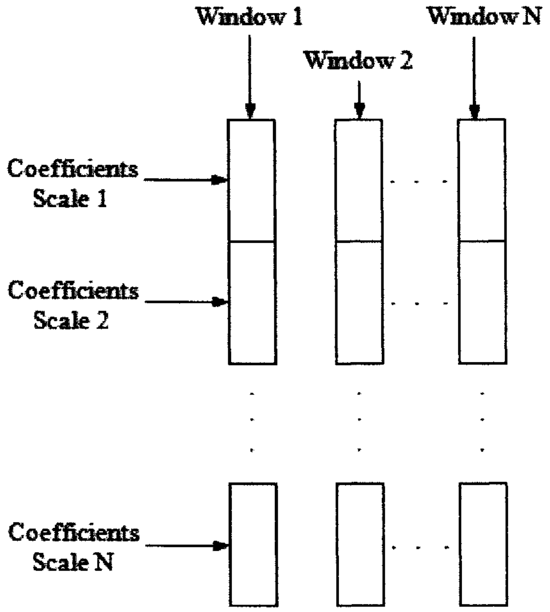


Fig. 5.10: Matrix storage of CWT coefficients

The Artificial Intelligence section consists of a small 2 layer feed forward, back-propagation neural network with 5 neurons in the first layer and 1 neuron in the second layer as described in chapter 4. Each neuron has an associated activation function that is constant for a particular layer, in this case the transfer function for the first layer is the Log-Sigmoid and for the second layer it is a linear.

There are as many inputs to each neuron in the first layer as there are scales in each column. The neural network processes its inputs in a manner that enables it to classify them into particular types. In this case the neural network output is close to zero to signify OOK RZ binary '0' or 'space', and close to one, '1', to signify binary '1' or 'mark'. The actual output from the final layer varies between something less than zero to something more than one, and in a pure digital sense the information is of little use. A threshold device or 'slicer' is therefore employed to force the output to a binary state.

Before the neural network can be used for classification (detection) purposes, it must be trained. Training consists of the transmission of a pre-determined signal stream sampled by the ADC to give:

$$x = \sum_{k=1}^{k=N} x_k \tag{5.11}$$

Where k = kT$_S$.

The signal is subjected to the noise and interference of the channel at the time. The process employs a training algorithm that may depend on the type of classification being attempted. In this case the 'Conjugate Gradient with Powell/Beale restarts' [160] is used. The topic of neural networks and training is discussed more fully in the previous chapter.

In this architecture, the implication of such a learning strategy is that if the channel characteristic changes the neural network may well start to classify incorrectly. Whilst adaptive techniques may resolve these issues, for simplicity the use of a training set has been adopted here. In these simulations this consists of 1000 or 1500 OOK RZ symbols (bits). In a practical system this would not be a huge burden since the transmission of 1000 bits at 2.5 Mb/s would theoretically only take 400 μs. Obviously the training process may take longer depending on the architecture and performance of the receiver processor. However, the indoor diffuse IR channel is slow moving and does not suffer from fade, therefore the number of training (redundant) bits to sustain a given BER may well turn out to be very low compared to the number of potential information bits.

The normalised optical power penalty of this system is shown in Fig. 5.11, for both the perfect channel and that subjected to fluorescent light interference. It can be seen that the Wavelet-AI receiver is far less susceptible to performance degradation due to the influence of fluorescent light interference than the matched filter case. The optical power penalty being less than 1 dB over the non-interfering channel; and being approximately 1 dB worse than the performance of a matched filter system receiving data over a perfect channel as per Fig. 5.6.

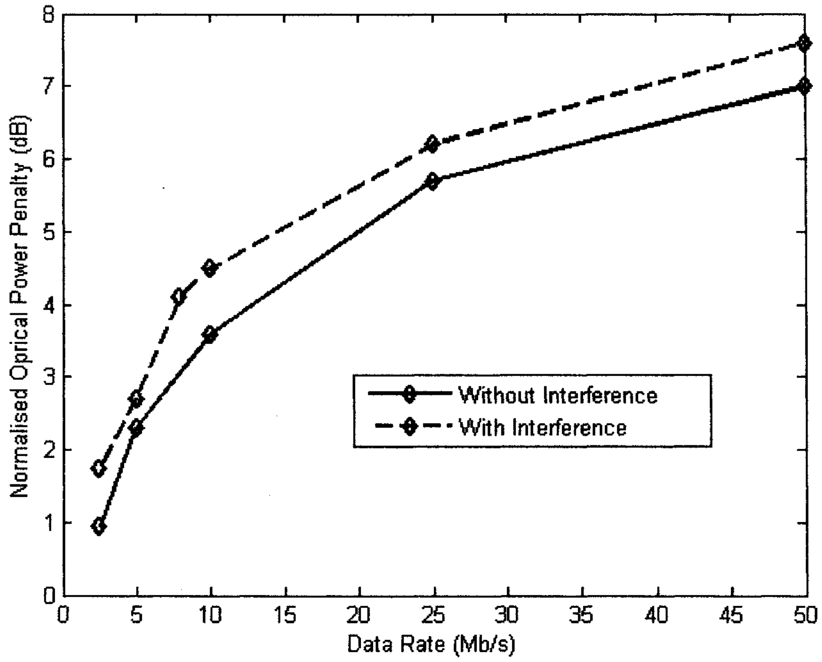


Fig. 5.11: Normalised optical power penalty versus data rate for OOK RZ with and without fluorescent light interference for a Wavelet -AI Receiver at a BER of 10^{-5}

By inspecting Fig. 5.3 we can see the variation in amplitude of the received signal with interference. The matched filter output of a system would be affected by this cyclical variation, severely affecting achievable BER. For reliable detection the signal amplitude has to increase to a point where the slicer can make a decision on variations in the received OOK signal rather than variations in interference. This situation can be contrasted with Fig. 5.12 and Fig. 5.13, which show 3D mesh (surface) plots of wavelet coefficients obtained by processing an OOK RZ signal with the CWT. In frequency (scale) terms the fluorescent light interference shown in Fig. 5.12 lies far away from the actual signal. It is difficult to produce a plot to the same detail incorporating the coefficients produced by the OOK RZ signal, as shown in Fig. 5.13. The visible scales associated with the interference are not passed to the neural network section of the Wavelet-AI receiver, effectively rejecting its contribution.

It is postulated that it is this scale distance between the two signal components that allow the Wavelet-AI receiver to reject much of the influence of the interfering signal without significantly compromising the quality of the information signal at relatively low bit rates. Interestingly it could be argued that passing coefficients associated with the interference to the neural network would lead to similar results due to the neural network compensating for the imposed variation. Although not simulated in this work, such a strategy would lead to increased computational complexity due to the increased processing required for the additional coefficients.

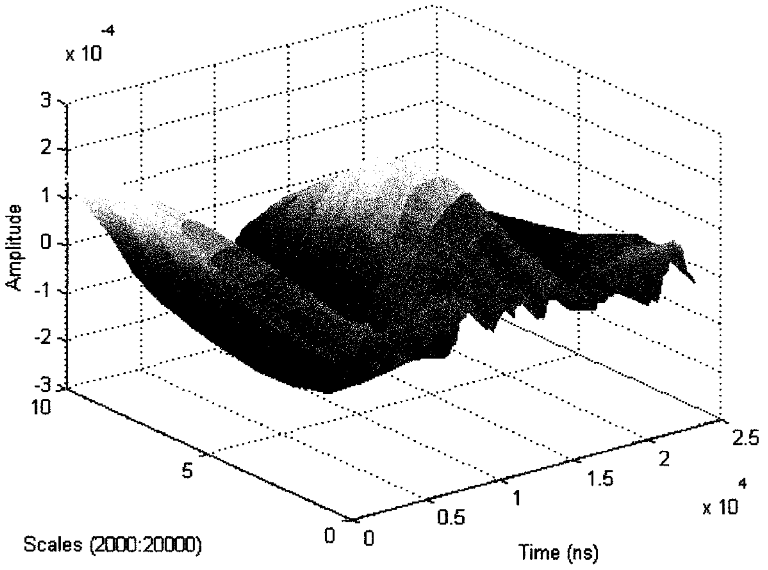


Fig. 5.12: Wavelet coefficients of signal with interference (low scales)

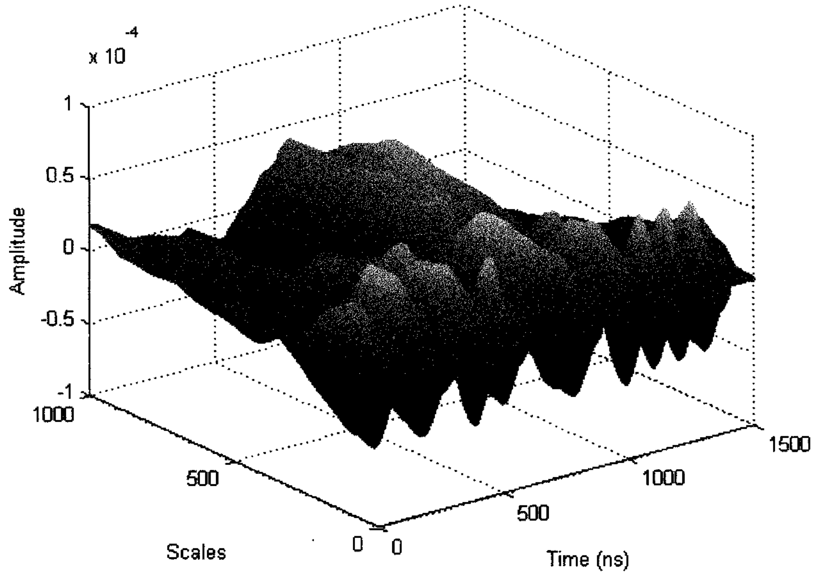


Fig. 5.13: Wavelet coefficients of signal with interference (high scales)

5.3.3 PPM (Matched Filter Receiver)

In this section we consider two different types of detection schemes employing the unit energy matched filter for PPM, these being the threshold detector PPM(TH), and the maximum a posteriori detector PPM(MAP), or soft decision decoder, as shown in Fig. 5.14 and Fig. 5.15, respectfully. Two levels of PPM encoding are simulated, these being L=4 and L=8. The program flowchart for both of these models is shown in Fig.5.16 as the same program is used for both. The selection between the two model types is managed via a variable assignment in the code. As the program for L=4 and L=8 is very similar only the L=4 code is provided on the accompanying CD.

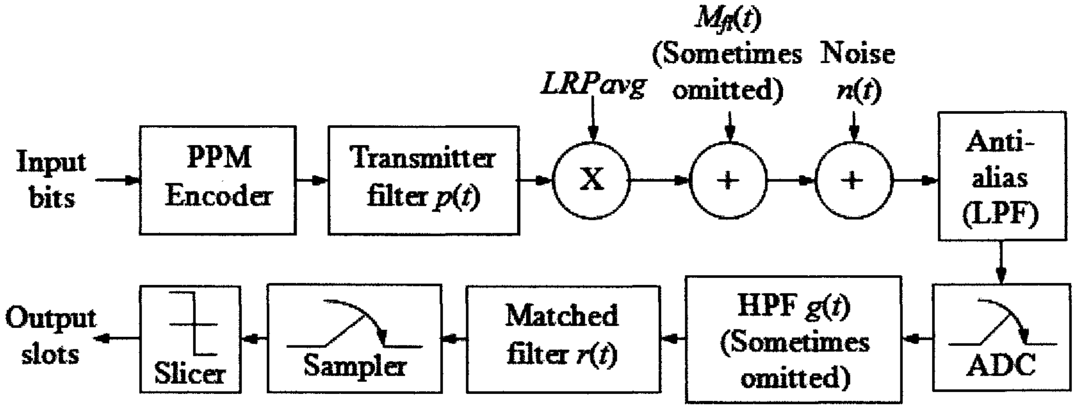


Fig. 5.14: PPM(TH) detector scheme

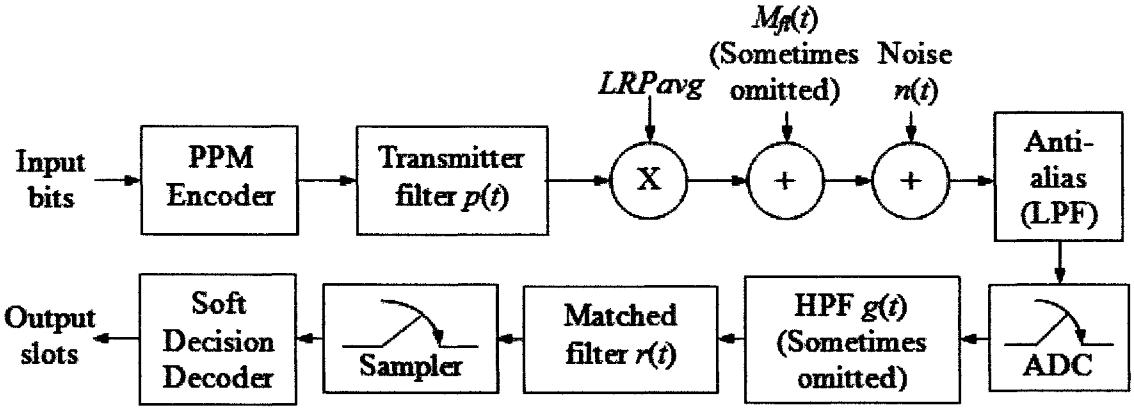


Fig. 5.15: PPM(MAP) detector scheme

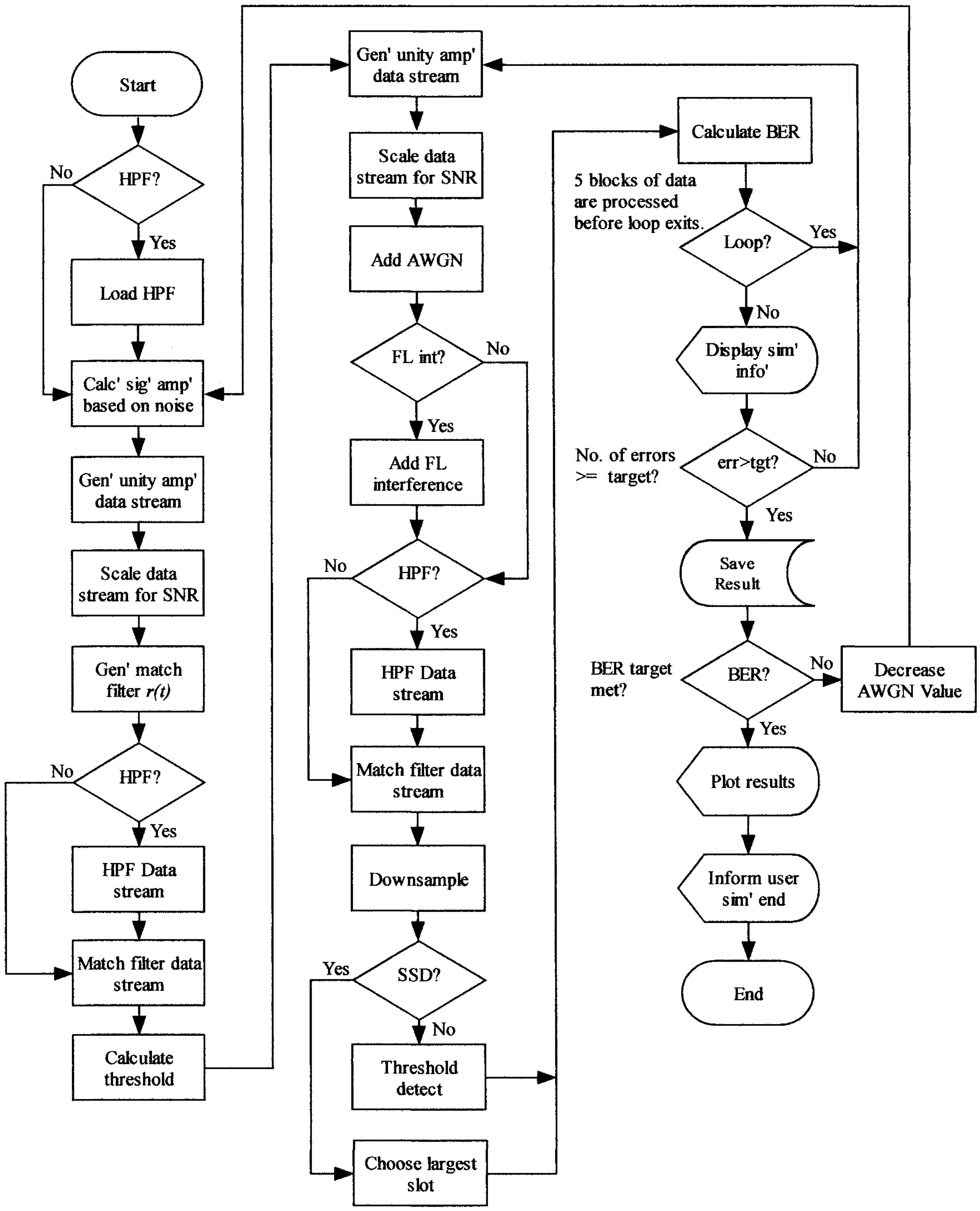


Fig. 5.16: Program flow chart for models shown in Fig. 5.14 and 5.15

In each system the PPM encoder converts each segment of *log₂L* input bits into one of the possible *L* symbols. The transmitter and channel are similar to that described for OOK; the symbols are passed to a transmitter filter which has a unit amplitude rectangular impulse response *p*(*t*), with the duration of one slot T_s, where $T_s = T_b \log_2 L/L$. The output of the transmitter filter is scaled by the peak detected signal photocurrent LRP_{avg}, where R is taken to be unity in these simulations. The fluorescent light interference $m_{fl}(t)$ is added, along with the shot noise $n(t)$. In this section the HPF is omitted and the detected signal is passed directly to the unit energy matched filter with an impulse response $r(t)$ matched to $p(t)$.

For the threshold detecting receiver the signal is sampled at the end of each slot, and a binary decision is made by the slicer depending on whether the signal is above or below some threshold level. Errors occur when the threshold level is exceeded for more than one slot, or when the threshold level is not exceeded at all during a symbol; in these cases some form of arbitration is required. Although theoretically marginally sub optimal, the threshold level in these simulations is set midway between one and zero levels.

Analytical derivations for the probability of error for PPM, with and without HPF are given in [79]. As with the OOK simulations, those carried out in this section do not rely on any analytical probability calculations and use a fully synthesised PPM signal subjected to shot noise and fluorescent light interference as described in section 5.2. However, for the PPM simulations a sampling interval T_s of 1 ns proved too much of a computational burden to be practical. Therefore a sampling interval T_s of 10 ns was used for all PPM simulations and the optical power penalties normalised to the power requirement of a 2.5 Mb/s signal to achieve a BER of 10^{-6}.

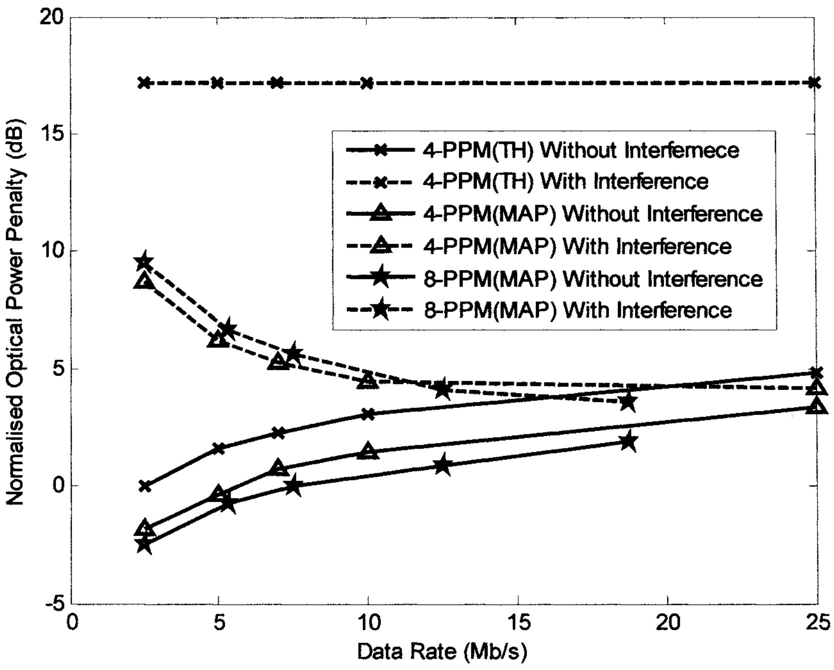


Fig. 5.17: PPM (TH & MAP) normalised optical power penalty a BER of 10^{-6}

Fig. 5.17 shows the results of PPM simulations for a BER of 10^{-6}. For 4-PPM both threshold and MAP detection was simulated, whilst for 8-PPM only MAP was simulated. All simulations were both with and without fluorescent light interference. In all cases the MAP detector outperforms the threshold detector; however, the performance degrades rapidly at lower data rates when fluorescent light interference is present with a marginally worse performance for 8-PPM. It can be seen that without interference the MAP detector yields an approximately 2 dB improvement over the threshold case for 4-PPM over the data rate range simulated, with L=8 improving over the L=4 case. The plot shows a discrepancy between the simulated data rates of the L=4 and L=8 case, this being due to the different number of slots for L=8 and the requirement that each slot have identical number of samples.

5.3.4 PPM (Wavelet-AI receiver)

In a similar way to OOK, the Wavelet-AI OOK RZ receivers shown in Fig. 5.18 and Fig. 5.22, show strong similarity to the traditional matched filter cases. However, again the operation is entirely different and it is conceivable that there are more available Wavelet-AI architectures available for PPM than for OOK; more than implemented here. This work considers two implementations as follows:

5.3.4.1 PPM Wavelet-AI Binary Symbol Detector (WBSD)

Fig. 5.18 shows a block diagram of the PPM(WBSD) detector, the program flowchart for this is essentially the same as for OOK Wavelet-AI receiver (Fig. 5.7) and is not repeated; however, the code is listed in appendix B. Whilst there is much similarity with Fig. 5.14 there are again fundamental differences after the ADC. In this scheme, the wavelet analysis section performs a CWT, using the Symlet 2 wavelet, on a set of previously chosen scales as shown in Table 5.5 and Table 5.6. This time, the window is not 3 bits wide, but 3 symbols or 12 slots as shown in Fig. 5.19, or 24 slots wide for 8-PPM. The coefficients for each window are processed in an identical manner as described for the OOK RZ case. However, in this scheme the neural network section is trained detect an entire symbol and not a single slot as would be analogous to the OOK system. For 4-PPM the artificial intelligence section has 12 neurons in the hidden layer and 2 neurons in its output layer as shown in Fig. 5.20; for 8-PPM a total 24 neurons are used in the hidden layer and 3 neurons in the output layer.

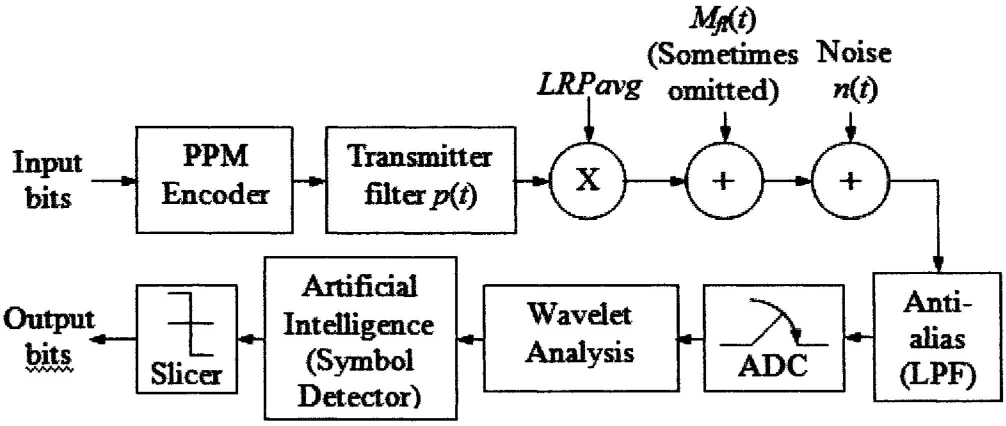


Fig. 5.18: PPM (WBSD)

For the 4-PPM case, given an array of coefficients corresponding to 12 slots, the neural network produces a corresponding binary symbol output. However, as with the OOK system, the actual outputs of the two neurons will range from somewhere less than zero to a little more than one. A slicer is therefore employed to force the output to a binary word as in the OOK case.

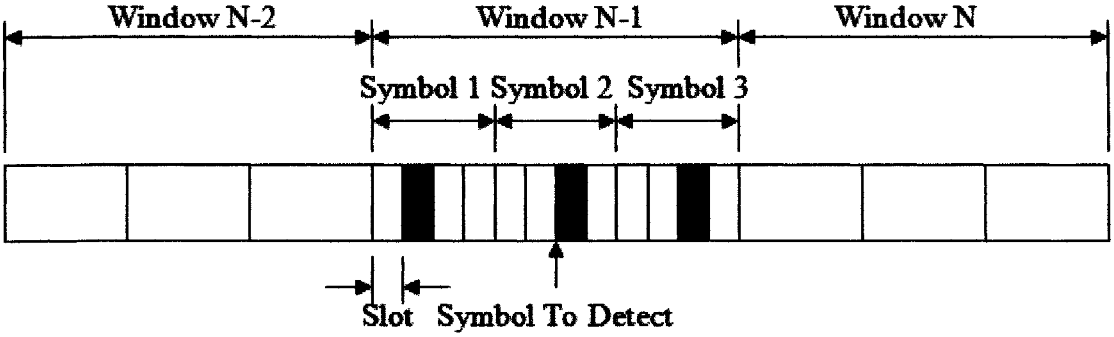


Fig. 5.19: 3 Symbol (12 slot window)

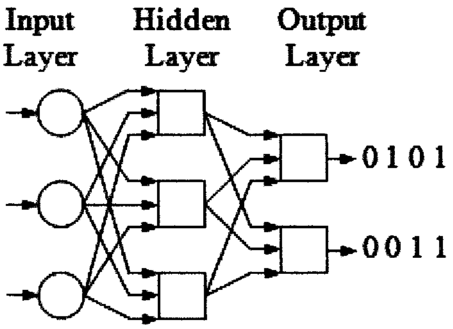


Fig. 5.20: Neural network for 4-PPM(WBSD)

Table 5.5: Scales for 4-PPM (WBSD).

Data Rate (Mb/s)	Scales
2.5	30, 45, 60, 75, 90, 120
5	7, 11, 22, 33, 44
7	7, 11, 22, 33, 44
10	5, 10, 15, 20, 25, 30, 35
25	5, 10, 15, 20, 25, 30, 35

Table 5.6: Scales For 8-PPM (WBSD).

Data Rate (Mb/s)	Scales
2.5	25, 75, 125, 175, 225
5.35	11, 22, 33, 44, 55, 66
7.5	9, 18, 27, 36, 54
12.5	3, 9, 12, 15, 18, 27
18.75	5 to 50 in steps of 5*
*Haar wavelet used for this simulation.	

5.3.4.2 PPM Wavelet-AI Soft Slot Detector (WSSD)

The Wavelet-AI soft slot detector (WSSD) is shown in Fig. 5.21, and the corresponding program flowchart in Fig. 5.22. The code is listed in appendix C. In this scheme slot detection rather than symbol detection is employed to determine whether a particular slot is a mark or a space as shown in Fig. 5.23. The technique used is very similar in most respects to the OOK RZ case where the CWT coefficients from a 3-bit window are passed to a neural network with 1 neuron in the output layer. The network is trained to output close to one for a mark and close to zero for a space.

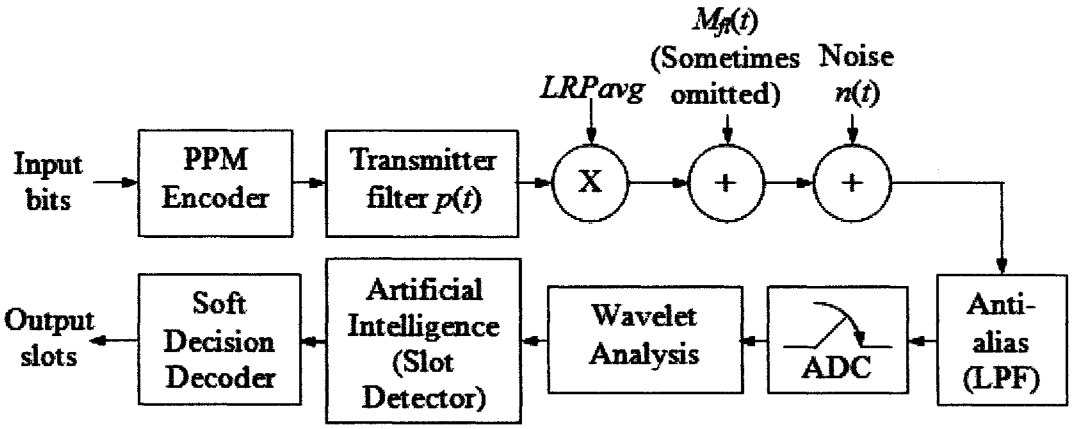


Fig. 5.21: PPM (Wavelet-AI soft slot detector)

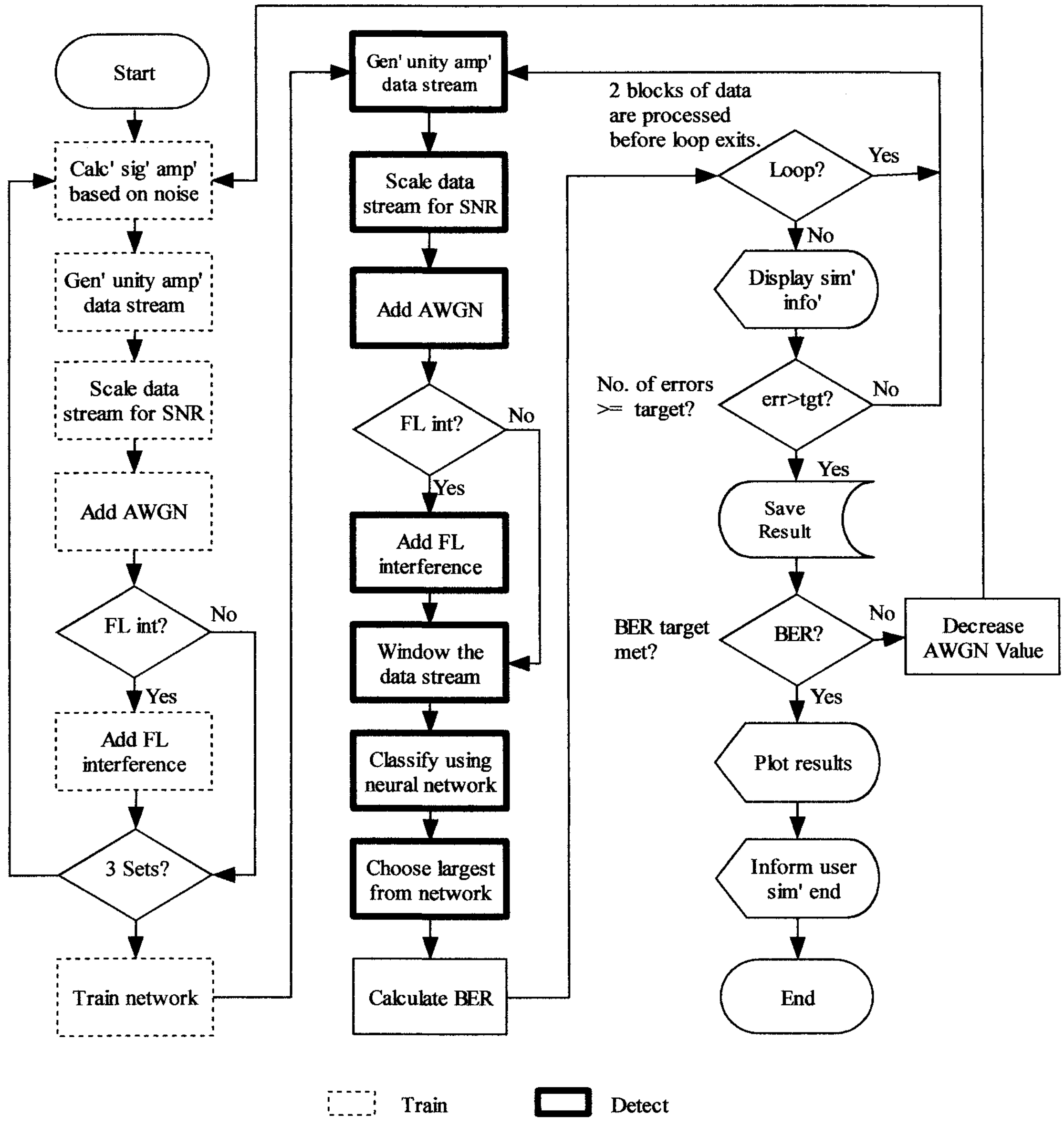


Fig. 5.22: Program flow chart for simulation model shown in Fig. 5.21

In this scheme a slicer is not used to force the output to a distinct binary level, instead a method very similar to PPM(MAP) is used. The actual levels output from the neural network for each slot in a particular symbol are compared as shown in Fig. 5.24. A mark is assigned to the slot with the largest value and a corresponding binary word produced by the receiver. The scales used for these simulations are shown in Table 5.7; in this case a neural network with 6 neurons in the hidden layer and 1 in the output layer was used.

Table 5.7: Scales for 4-PPM soft slot detection

Data Rate (Mb/s)	Scales
2.5	30, 45, 60, 75, 90, 120
5	16, 32 48, 64, 80
7	11, 22, 33, 44
10	5, 10, 15, 20, 25, 30, 35
25	5, 7, 9, 11, 13

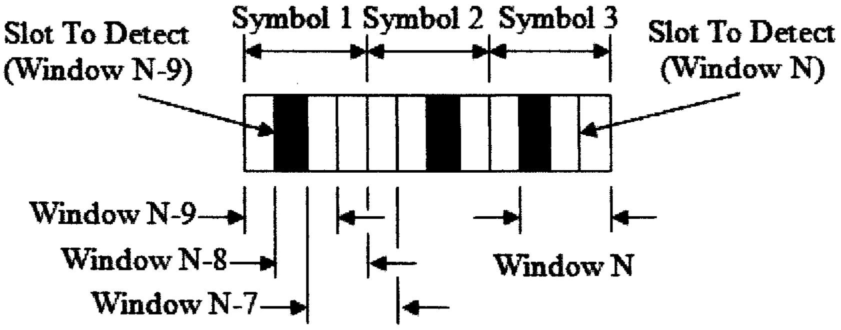


Fig. 5.23: Windowing methodology for soft slot detection

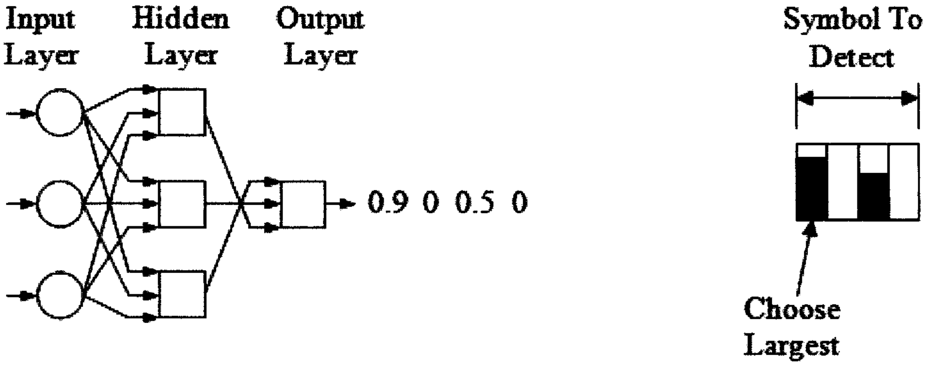


Fig. 5.24: (a) Single neuron slot detector and (b) choose largest method

5.3.4.3 PPM Results (Wavelet-AI Receiver)

The results obtained from simulations of *L*x-PPM for different cases with a BER of 10^{-6} are shown in Fig. 5.25. The results show a definite trend, indicating a reduction in power penalty for a reducing data rate; however, it is observable that the individual curves are more erratic. Whilst impossible to prove or disprove with the current data, this behaviour may connected to a deficiency in neural network training. Looking at the OOK case and the 3 bit window, the neural network is presented with coefficients representing 8 unique patterns (ignoring channel influences). On average, the neural network will see the same pattern 125 times during 1000 training bits. If we now consider the 4-PPM(WBSD), the number of unique patterns increases to 64 with 12 slots for each pattern, requiring 768 slots if every pattern is to be seen at least once. 1000 training symbols provides 4000 training slots, so on average each pattern occurs just over 5 times in 1000 training symbols. For 8-PPM(WBSD) the number of unique slot patterns increases to 512 with 24 slots required for each pattern, meaning that 12288 training slots are required for every pattern to be seen by the network at least once. 1000 training symbols would provide 8000 training slots. This obviously means that the neural network may never see an example of some patterns during its training period.

The neural network, through training, will ignore redundant information. Whilst unproven and subject to further research, it is possible that for the 4-PPM binary symbol detector only the centre symbol is important. If the network disregards the other symbols then there are only 4 unique patterns, 8 for 8-PPM. From this it is apparent that further work on training strategies may yield improvements in both variations and ultimate performance; however, time limitations and sufficiently positive results from the simulations so far dictate that this has to be the subject of further work.

The Wavelet-AI 4-PPM(WBSD) performance without interference below 5 Mb/s is almost identical to the 4-PPM(TH) case; however, above 10 Mb/s the Binary Symbol Detector shows an approximately 0.6 dB improvement. When fluorescent light interference is introduced the binary symbol detector shows similar performance to the non-interference case, however, at higher data rates the performance degrades by approximately 0.7 dB. In contrast, when 4-PPM(TH) is subjected to interference its performance at lower data rates degrades more than 17 dB, whilst at higher rates the degradation is approximately 12 dB.

The 4-PPM(MAP) detector without interference shows an almost 2 dB improvement over the 4-PPM(TH) case and a slightly smaller improvement over the 4-PPM(WBSD), narrowing to 1 dB at higher data rates. With interference 4-PPM(MAP) and the 4-PPM(WBSD) have similar performances at 20 Mb/s, however, the performance curves diverge at lower data rates with the Binary Symbol Detector showing an approximately 9 dB improvement at 2.5 Mb/s.

The 4-PPM Soft Slot Detector without interference shows an improvement over the Binary Symbol Detector a little under 2 dB and approximately the same as 4-PPM(MAP). However, with interference there is an almost 4 dB penalty with the Soft Slot Detector and the 4-PPM(MAP) detector having the same performance at approximately 9 Mb/s. At 25 Mb/s 4-PPM(MAP) with interference outperforms the Soft Slot Detector by 2 dB, whilst at 2.5 Mb/s the Soft Slot Detector outperforms PPM(MAP) by approximately 7 dB.

Due to time limitations only PPM(MAP) and PPM(WBSD) were simulated for 8-PPM. Given the variations in results for the Binary Symbol Case and the possible limitations due to training strategy it would be difficult to draw firm conclusions from the results other than to say the trend under the influence of artificial light interference still shows improvement over the 8-PPM(MAP) case at lower data rates. The performance of the 8-

PPM(WBSD) for the interference and no interference case being broadly similar, but not showing improvements over the L=4 case.

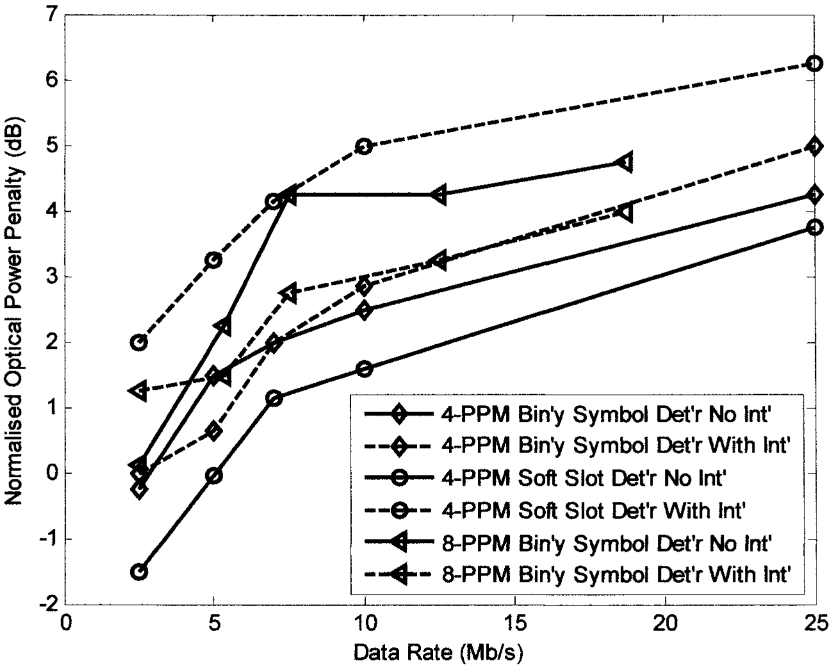


Fig. 5.25: *Lx*-PPM (binary symbol & soft slot) normalised optical power penalty for a BER of 10^{-6} for two detection cases with/without interference

5.4 The Effect of Baseline Wander on OOK without Fluorescent Light Interference

Although a HPF is effective at attenuating frequencies below cut-on, it has the unfortunate effect of introducing a type of ISI known as baseline wander. A sequence of pulses that are passed through such a filter exhibit a variation in the nominal zero level as shown in Fig. 5.26, the value of the offset being determined at any given time by the past history of the pulses [174]. This type of ISI has a degrading effect on the performance of base-band modulation schemes where significant power is located at or close to DC. In this section the optical power penalty to overcome the effect of baseline wander on both the OOK RZ/NRZ matched filter system based on Fig. 5.4 and the Wavelet-AI system of Fig. 5.27 is investigated. No flowchart or code listings are given for these simulations as the components for the standard case are essentially the same as Fig. 5.4.

It should be noted that the configuration shown in Fig. 5.27 is not intended to be a deployable case for the Wavelet-AI system as it is postulated that a HPF is not required for such a receiver. However, the arrangement is required as a vehicle to investigate the effects of baseline wander so that a direct comparison can be made with the traditional structure of Fig. 5.4.

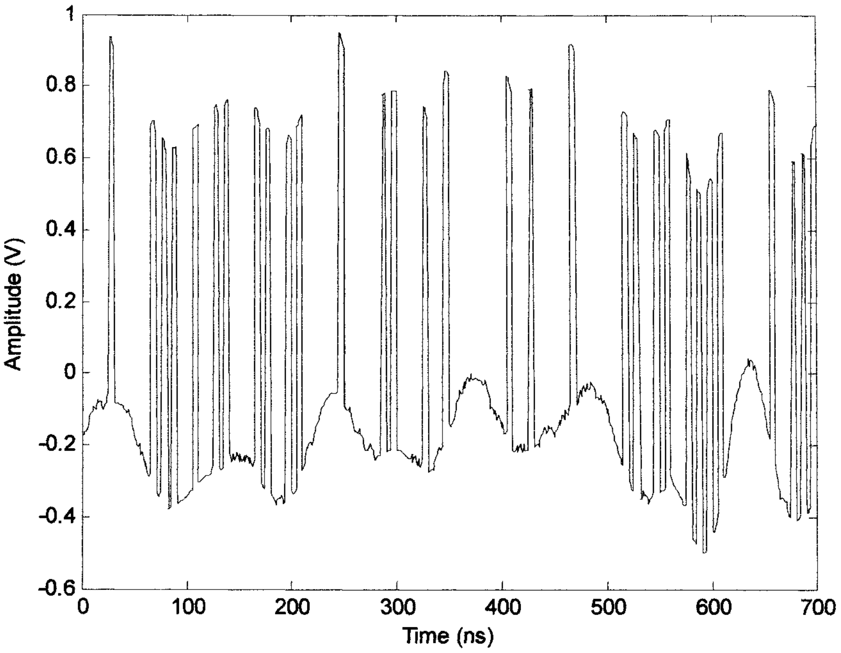


Fig. 5.26: OOK RZ signal suffering baseline wander from HPF

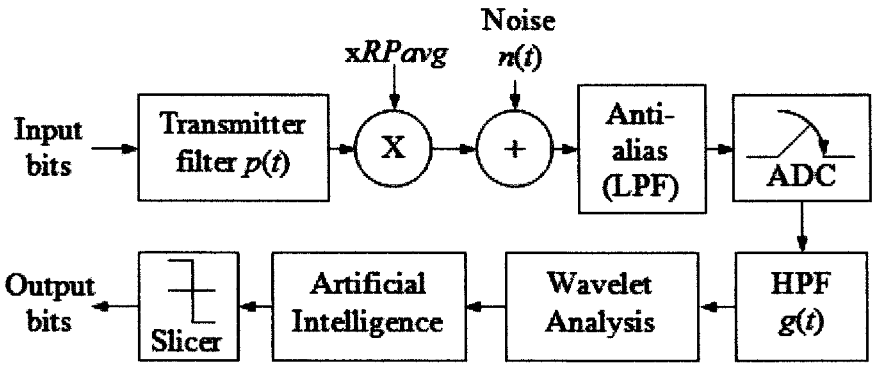


Fig. 5.27: Wavelet-AI system for baseline wander

For this investigation two types of high-pass filtering are used, the first of these being implemented as an analogue first-order RC filter with a 3dB cut on frequency of f_c, and an impulse response denoted as $g(t)$. The second type of filter is a digital FIR type, again with a cut-on frequency f_c and an impulse response given by the filter coefficients b_n.

Taking the analogue filter first, its impulse response to a single rectangular pulse of amplitude A and duration T_b may be expressed as [175]:

$$g_{out}(t) = \begin{cases} Ae^{-t/RC} & 0 \le t \le T_b \\ -A\left(e^{T_b/RC} - 1\right)e^{-t/RC} & t > T_b \end{cases} \tag{5.14}$$

Where RC is the filter time constant and is related to the cut on frequency by $RC = 1/2\pi f_c$.

Due to the superposition of linear systems, if a sequence of pulses is passed through a HPF, the output is equal to the summation of the individual responses of the pulses within the sequence. So for a bit sequence $A_1, A_2 \cdots A_n$ where $A_{1..n} \in \{0,1\}$, the output of such a filter at the end of the n^{th} bit may be expressed as [80]:

$$g_{out}(t)\Big|_{t=nT_b} = \sum_{i=1}^{n} A_i \left(e^{-2\pi f_c T_b} - 1\right)\left(e^{-2\pi f_c T_b}\right)^{i-1}. \qquad (5.15)$$

Previous work [72] implements a technique that considers the discrete time equivalent impulse response of the cascaded transmitter filter, analogue high pass filter and receiver filter, denoted as C_j. The impulse response of such a system is infinite but rapidly decays to zero with time and can therefore be approximated by a truncated sequence of length J.

Evidently, by considering every possible sequence of bits equal to length J, calculating the probability of error for the J^{th} bit and then averaging over all possible sequences would provide a result for the bit error rate (BER) of such a system. Previous work also considers normalised cut-on frequency; this is the cut-on frequency of the HPF divided by the bit rate, f_c/R_b, a metric that is independent of data rate or filter cut-on frequency for a particular model architecture.

In common with other work on the Wavelet-AI system, this section considers a simulation strategy is somewhat different to that adopted in [72] and simply employs a stream of random pulses of finite length. It is ensured that any filter used has become stable and that any DC content has been removed before simulation results are processed. Basically, the parts of the data stream leading into and out of the transmission are stripped away leaving the centre stable part for BER analysis.

The output of the digital FIR filter is given by [108]:

$$y_k = \sum_{n=0}^{L} b_n x_{k-n} \qquad k = 0, 1, \tag{5.16}$$

Where b_n are the filter coefficients.

Furthermore, given the digital nature of the Wavelet-AI receiver, a digital FIR filter would be the one to adopt. For simulation purposes these have the advantage of a response that decays to zero, and within limits, a modifiable length. Such a filter removes the need to consider a truncated length analogue type, and produces an easily quantifiable number of unstable data bits to be removed from the transmission stream prior to BER analysis. All digital filters used in the simulation models were designed using the Matlab™ filter design and analysis tool. With the exception of the lowest cut-on frequency, all filters have 1064 taps with 60 dB of attenuation in the stop band, and a 2 MHz transition band.

Table 5.8: Wavelet-AI simulation parameters.

Wavelet Section:	
Transform	CWT
Mother Wavelet	'Sym 2'
Scales	8, 21, 41 & 61
Window Size	3
Neural Network Section:	
Network Type	Feed Forward, Back Propagation
Layers	2
No Neurons	5 in Layer 1, 1 in Layer 2
Transfer Functions	Log-Sigmoid (Layer 1) Positive Linear (Layer2)
No Training Symbols	1500
Noise Training Values	6, 9, 12, 15 & 18 dB*
Training Algorithm	Powell-Beale
Bit Position Detected	2 (Centre Bit)
HP Filter	
Type	Equi-ripple FIR
Attenuation	60 dB
Transition Band	2MHz**
Taps	1064***
* Added to training symbols. ** 0.4 MHz for f_c/R_b $5\text{x}10^{-3}$. *** 5319 taps for f_c/R_b $5\text{x}10^{-3}$.	

A number of simulations using the traditional model structure of Fig. 5.4, with both the analogue filter and the digital filters were carried out. For the OOK NRZ case the output of the transmitter filter is scaled to $2RP_{avg}$. The simulations included analysing the stable section of each data stream to determine the optimum threshold decision level for every value of f_c/R_b.

The model parameters for the Wavelet-AI receiver are shown in Table 5.8. The output of the HPF is fed through the wavelet analysis section that produces wavelet coefficients by performing a CWT with a Symlet 2 wavelet on a 3 bit window. The coefficients from this are passed to a feed-forward back-propagation 2 layer, 6 neuron neural network for classification into a binary decision based on the centre bit as described in 5.3.2. The output of the neural network is fed into a slicer for regenerating the transmitted data. The three consecutive bits are all indexed by one bit and the whole process is repeated. No changes to the model parameters or structure other than to introduce new filter coefficients were made throughout the range of simulations.

Prior to transmission (BER analysis) the Wavelet-AI receiver must be trained to detect a mark or space from the incoming data stream. For this purpose a predefined sequence of 1500 bits is transmitted with white Gaussian noise added with values between 6 and 18 dB. All simulations were run at a data rate of 100 Mb/s and a sampling rate of 1 ns, hence 10 samples per bit.

Fig. 5.28 shows the normalised optical power penalty against f_c/R_b for 100 Mb/s OOK NRZ and RZ data formats with both the standard and Wavelet-AI models using analogue and digital HPFs. The optical power penalty is normalised to the power required for a BER of 10^{-6} without a HPF in the OOK NRZ case and to 10^{-5} in the OOK RZ ($\gamma = 0.5$) case. It can be deduced from the simulation results (curves 1 and 2) that the digital FIR filters outperforms the analogue filter by a noticeable margin when OOK NRZ is employed in the traditional model. We also see from curve 3 that the shorter pulse duration of OOK RZ yields a further performance improvement.

For the Wavelet-AI case, simulations were run with digital filters having a range of cut-on frequencies from 0.5 MHz to 110 MHz (f_c/R_b = 1.1) which is beyond the bit rate of 100 Mb/s. It can be seen from curves 4 and 5 that the Wavelet-AI receiver is significantly more tolerant to baseline wander than the traditional receiver model, even with the improvement afforded to the traditional model by using a digital FIR filter in place of the analogue version. For example for f_c/R_b. of 10^{-1}, the power penalty for Wavelet-AI is half that of the traditional receiver with digital FIR HPF (RZ).

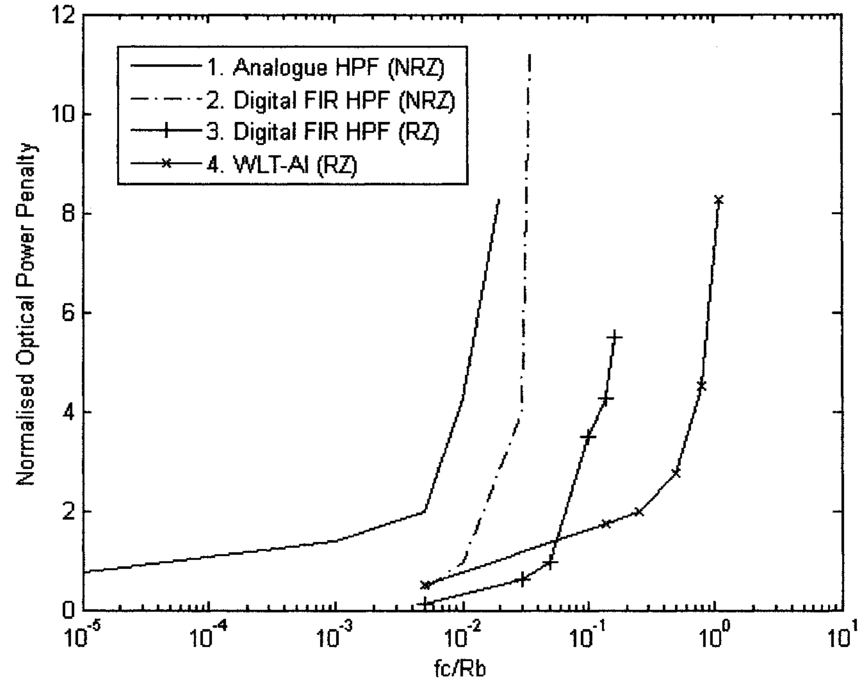


Fig. 5.28: Normalised optical power penalty for OOK with baseline wander for a BER of 10^{-6} without a HPF in the OOK NRZ case and to 10^{-5} in the OOK RZ (γ = 0.5)

5.5 The Effect of Fluorescent Light Interference with Electrical High-Pass Filtering

As discussed earlier an electrical HPF can diminish the effects of fluorescent light interference, whether this is implemented in the analogue or digital domain. Also, the choice of cut-on frequency is a trade off between the ISI introduced by the HPF and the attenuation of the interfering signal; the optimum choice being that which minimises the overall power penalty. Whilst f_c is an important parameter, other filter characteristics such as stop band attenuation and roll off are also likely to have an affect on performance that may produce variations in the optimum f_c. Given that the optimum f_c changes with data rate, one solution would be to simulate a system over all envisaged data rates and f_c to determine the correct cut-on frequency to use. This is obviously less than practical and time consuming; a compromise method being to choose a limited number of data rates and a range of f_c for a given filter to determine as close to the optimum as possible.

Time constraints have not allowed the pursuit of exhaustive simulations required to determine a set of filter characteristics that provide optimum results under the influence of the fluorescent light model for the simulated data rates. Previous work on the subject [72] indicates that filters with f_c in the 1 MHz region dramatically improve traditional system performance for both OOK and PPM. Given the time constraints this section therefore looks at the simulation results of filters with cut-on frequencies of 0.5 MHz and 1 MHz.

5.5.1 OOK RZ

The simulations in this section are based on the models shown in Fig. 5.4 and Fig. 5.7. The HPFs for Fig. 5.4 are those used in the simulations described in section 5.4 Eq. 5.14 and Table 5.8. As discussed, no filters are required for the Wavelet-AI model, the simulations simply being those of section 5.3.2 and Fig. 5.7.

Taking the traditional receiver structure first, the results shown in Fig. 5.29 indicate that the first order analogue HPF provides between 4 dB and 6 dB of improvement over the non-filtered case. However, the FIR filters provide a marked improvement over this; the 1MHz filter showing up to an approximately 13 dB improvement over the non filtered case, whilst the 0.5 MHz filter shows improvement up to approximately 14 dB. The significant difference between the two cut-on frequencies being the data rate at which the performance starts to diminish, the 1 MHz filter showing deterioration below 25 Mb/s with an increasing power penalty for data rates lower than approximately 10 Mb/s. The 0.5 MHz filter being somewhat better than this showing increasing power penalties from approximately 5 Mb/s. The 0.5 MHz filter also suffers from an increase in the number of filter coefficients from 1024 in the 1 MHz case to 5039.

In contrast to the traditional receiver structure the Wavelet-AI receiver's performance is not limited by a particular cut–on frequency with the power penalty continuing to improve as the data rate decreases. At higher data rates the Wavelet-AI system is marginally worse than the traditional receiver employing a 1 MHz filter and shows a less than 1 dB penalty over the 0.5 MHz filter.

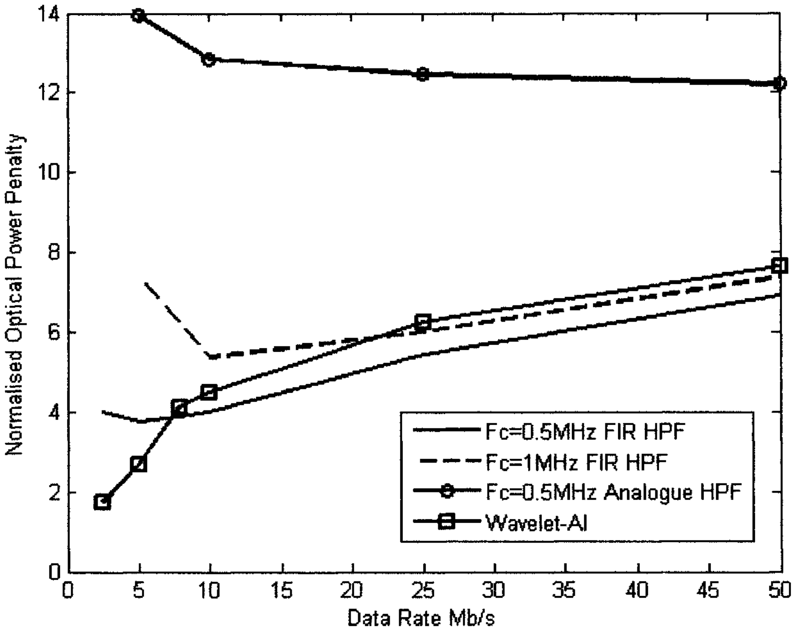


Fig.5.29: Normalised Optical Power Penalty against data rate for OOK RZ for FIR and analogue HPFs and Wavelet-AI

5.5.2 PPM

The traditional receiver models used in this section are shown in Fig. 5.14 and Fig. 5.15; the filters employed being the 0.5 MHz and 1 MHz HPF types considered in the OOK case. 4-PPM(TH) without high pass filtering (Fig. 5.17) shows similar performance to the comparable OOK case (Fig. 5.6). As with OOK, employing a HPF significantly improves performance with both the 0.5 HMz and 1 MHz filter power penalty profiles of Fig. 5.30 being very similar. Filtering shows a power improvement of approximately 15 dB at a data rate of 5 Mb/s, diminishing to approximately 12 dB at 25 MHz. It is interesting to note that the power penalty below 5 Mb/s increases more rapidly than the OOK case, resulting in an optical power penalty of 49.5 dB at 2.5 Mb/s (not shown) being some 32 dB worse than the same system without filtering.

4-PPM(MAP) without high pass filtering also shows significant improvement over the comparable 4-PPM(TH) system, with increasing power penalties at lower data rates. Inserting the filters shows further improvements with both cut-on frequencies providing similar performance below 10 Mb/s with an approximately 2 dB improvement over the comparable 4-PPM(TH) penalty profile. In a similar manner to the (TH) case, the power penalty increases sharply below 5 Mb/s.

4-PPM(WBSD) shows very similar performance to the (TH) system above 10 Mb/s, with marginally worse performance above this figure. At lower data rates the Wavelet-AI system shows diminishing power penalties becoming divergent with the with the 4-PPM(TH) curves at a data rate of 5 Mb/s and showing an almost 5 dB difference in optical power penalty at 2.5 Mb/s. 4-PPM(WDSS) shows diminished performance over 4-PPM(WBSD) and the traditional systems at all but the lowest data rates simulated. However, even here the performance curve becomes divergent with that of the traditional systems indicating further improvements at lower data rates.

8-PPM simulations were very limited in number due to time constraints. However, 8-PPM(MAP) and 8-PPM(WBSD) were simulated with resulting normalised optical power penalties being shown in Fig. 5.31. In the latter case very little empirical research was done to optimise the performance. The MAP receiver employing a 0.5 MHz filter improves on the corresponding 4-PPM case as expected. The Wavelet-AI case shows deterioration in performance over the corresponding 4-PPM case.

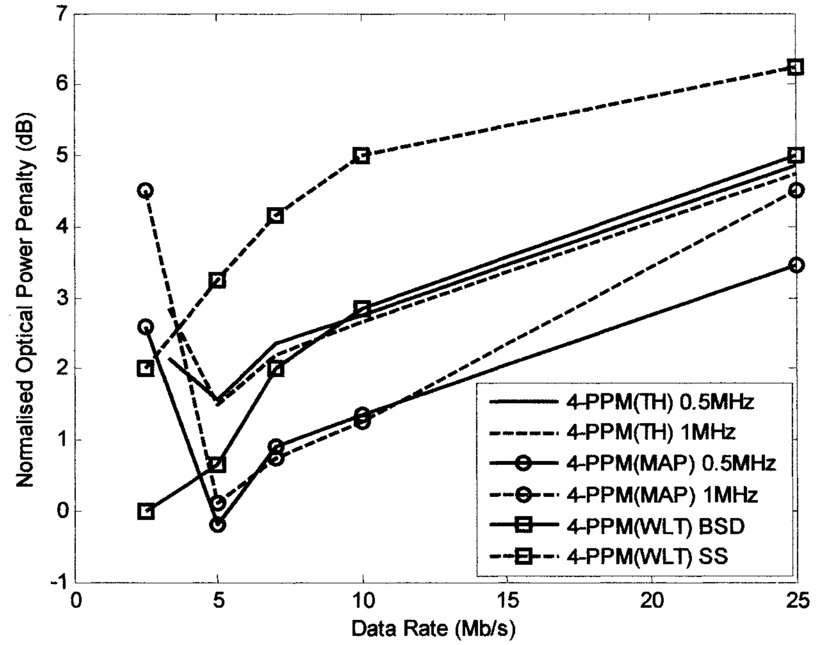


Fig. 5.30: Normalised optical power penalty against data rates for 4-PPM and different detection schemes at BER rate of 10^{-6}

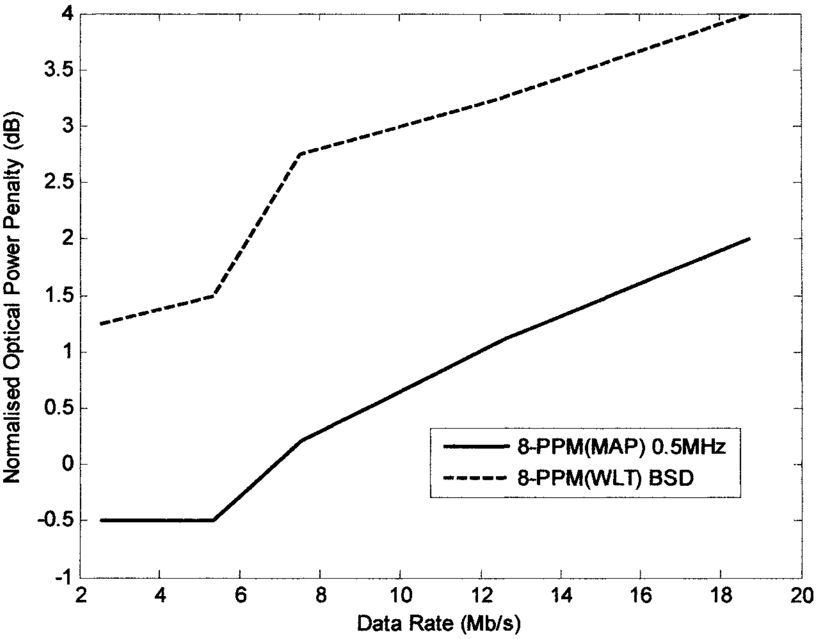


Fig. 5.31: Normalised optical power penalty against data rates For 8-PPM with MAP and Wavelet-AI detection at BER rate of 10^{-6}

5.6 Summary

Given the experimental nature of the Wavelet-AI models described in this chapter it is not unreasonable to suggest that the simulation results could be improved. Selection of optimum scales, wavelet, transform and neural network could all potentially offer performance enhancements. However, even with the limitations of experimental parameter selection the Wavelet-AI receiver performs reasonably well compared to traditional techniques. Indeed the Wavelet-AI receiver outperforms the simulated traditional receivers at low data rates in the presence of fluorescent light interference with the exception of 8-PPM, where only limited simulations were completed.

Further to this, the potential for deploying the Wavelet-AI system with software based digital architecture is attractive. It was shown that with suitable optimisation work on scale selection that a single receiver is capable of implementing numerous modulation schemes, some of which may prove more efficient with this type of architecture. Moreover, being largely software based, the receiver could easily switch between modulation types, choosing the best performance for a particular channel characteristic instance.

As was outlined in this chapter, traditional architectures are reasonably dependent on pre-defined choice, not only in terms of the implemented modulation scheme, but also on particular data rates and corresponding optimum filters. However, the Wavelet-AI receiver is not constrained by these parameters and showed a continuum of performance primarily discretised only by the sampling rate of the ADC. Such potential flexibility perhaps offering enhanced performance in the presence of previously undefined interferers over a much wider range of data rates than were shown here.

Chapter 6

The Effect of Multipath Propagation

6.1 Introduction

One of the major difficulties with the diffuse indoor IR environment is inter-symbol interference (ISI) caused by the many reflections a transmitted signal undergoes before arriving at the receiver. A once reflected signal would have the minimum delay, except in the special line of sight (LOS) case, and maximum power provided that all surfaces attenuate the signal equally. Unfortunately other paths within the room will reflect the signal many more times resulting in a delayed and attenuated additional contribution to the once reflected signal seen at the receiver; these contributions result in a 'smearing' of the pulse shape. Fig. 6.1 shows the effect of multipath propagation on a 40 Mb/s OOK NRZ signal with a normalised delay spread D_T Of 0.328, where variations in signal amplitude due to preceding pulses can easily be seen. From inspection of Fig. 6.1 it is intuitive to see that for a given severity of channel induced distortion higher data rates will cause further variation in signal amplitudes. It can also be noted from the figure that the best sampling point would be at the end of the bit period.

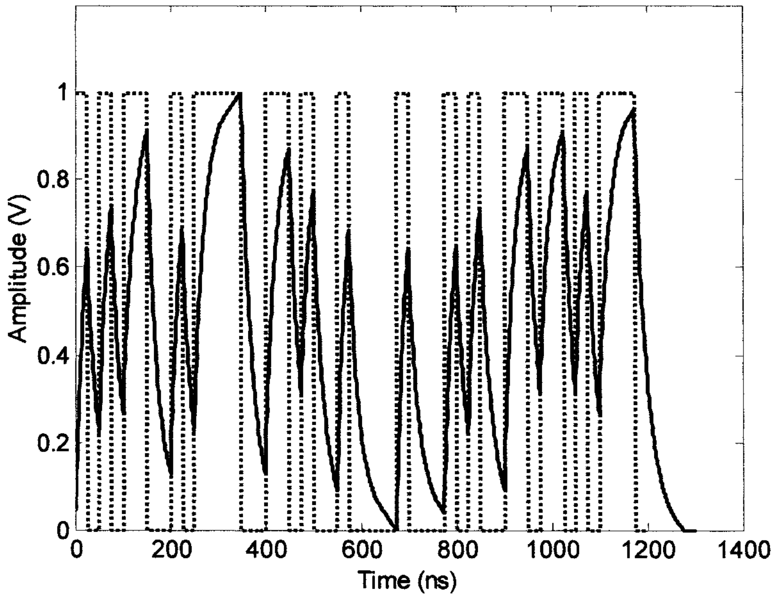


Fig. 6.1: Example of multipath distortion on OOK NRZ 40 Mb/s data stream

The corresponding 'eye diagram', shown in Fig 6.2, is constructed from several overlapping traces and provides us with an insight as to how severe the effects of ISI will be. The height of the diamond shape opening or 'eyes' in the diagram show how much the difference between a '0' (zero) and '1' (one) has been reduced by the multipath distortion, from 1 V down to about 0.3 V in this case. A decrease in 'eye opening' limits the ability of the receiver to discriminate between a '1' or '0' in the presence of noise.

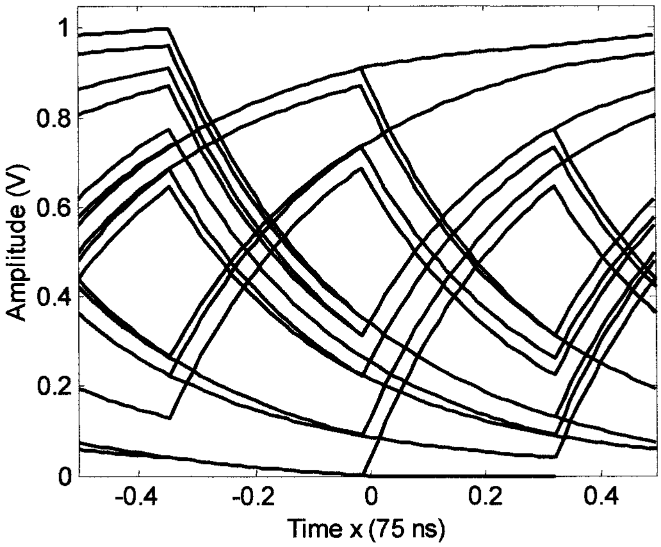


Fig 6.2: Example eye diagram generated from a 40 MHz OOK NRZ multipath signal

The decreased transmission power of lower duty cycle modulation techniques are favoured for implementations of diffuse IR systems due to the limited power available to portable devices and to satisfy ocular safety requirements. On a perfect channel reducing modulation duty cycle results in an improved SNR for a given BER. However, multipath channels affect shorter pulse durations due to the increased rise and fall times of the channel and the subsequent ISI caused by preceding pulses. Both contribute to the unpredictable amplitude of the received signal making threshold-based decisions difficult.

Fig. 6.3 shows the simulation results of a simple threshold detector based receiver for both OOK NRZ and OOK RZ on a mutipath channel with a delay spread of $8.19\text{x}10^{-9}$ s. It can be seen that for this level of delay spread that there is a performance advantage of using OOK RZ for data rates of 10 Mb/s. However, the figure shows that at 25 Mb/s the advantage is lost and the longer duty cycle of OOK NRZ modulation outperforms the RZ case. At 35.7 Mb/s the RZ case requires in excess of 10 dB more power for the same BER. Near 40 Mb/s the BER for OOK RZ becomes irreducible regardless of the transmitted power; whilst for OOK NRZ this situation is not reached until after 70 Mb/s.

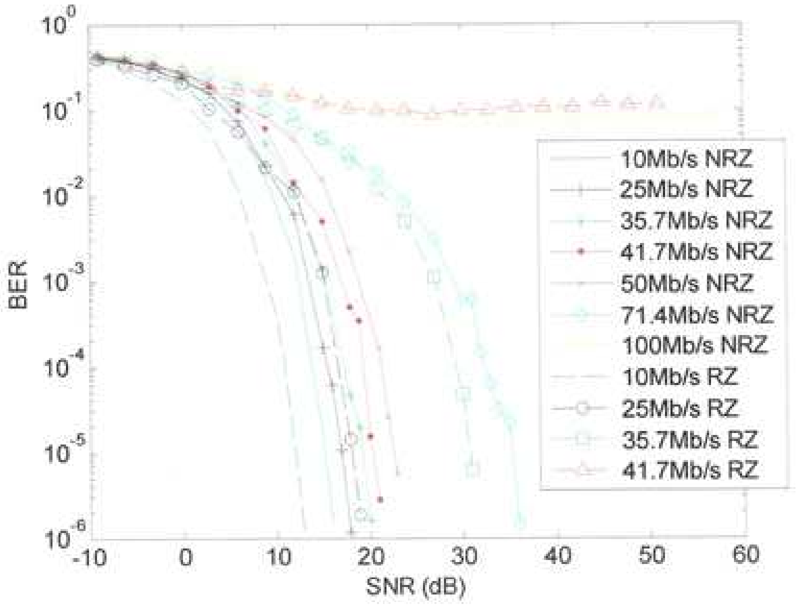


Fig. 6.3: BER against SNR for OOK RZ/NRZ subjected to ISI

The majority of reflective surfaces in the indoor environment behave as near perfect Lambertian reflectors [22], with an incident ray being diffusely reflected in all directions. This has lead a number of researchers to characterise a particular room using ray tracing methods. Other researchers have used mathematical models to define the behaviour of a generic diffuse indoor environment where only the overall dimensions are important, and specific measurements on particular rooms are not required. Characterisation and modelling of the indoor IR channel was introduced in chapter 2 and the subject of numerous publications such as [25, 35, 41, 54, 58-66, 68 - 70].

The multipath channel, which is completely characterised by its impulse response *h(t)*, is fixed for a given position of the transmitter, receiver and reflecting surfaces. The impulse response will only change significantly when any of these move on the scale of the wavelength of the modulating signal, this being about 3 m at 100 MHz [22], thus providing a quasi-static channel characteristic. Due to the high bit rates under consideration and the relatively slow movement of people and objects within a room, the channel will vary significantly only on the time scale of many bit periods, and hence, it is justifiable to model the channel as time invariant.

The remainder of this chapter is organised as follows: Section 6.2 discusses some common mitigation method used to combat the effects of the multipath channel and introduces the Wavelet-AI receiver as an alternative mitigation technique. Section 6.3 introduces the channel model used in subsequent simulations. Section 6.4 reviews the simulated unequalised performance of traditional and Wavelet-AI architectures for both OOK RZ and PPM transmissions over a multipath channel. The simulated equalised performance of traditional receiver architectures is discussed and compared with that of the Wavelet-AI system in section 6.5. Finally, the chapter is summarised in section 6.6. Code listings for the more significant simulation models introduced in this chapter are included in the appendices. In general the more comprehensive models have been listed as the majority of the results discussed can be obtained from them by omission of certain functionality such as equalisers.

6.2 Mitigation Techniques

It is clear that given the performance constraints of a simple threshold based receiver on a noisy multipath channel that ways must be found to reduce the degrading effects of ISI and noise. Most techniques employed in diffuse indoor IR systems are implemented in the electrical domain and are borrowed from the more established RF environment. A selection of these methods is reviewed and described in the following sections.

6.2.1 Filtering

One of the most simple and frequently implemented methods used to improve performance in communication systems is filtering. In IR systems optical filters can be attached to the lens of the photodiode receiver to reject out of band light thus limiting the effect of noise sources such as natural and artificial lighting. Further band limiting filtering can be done in the electrical domain either in analogue, or digital format using DSP techniques. Such filters are used to reject frequencies that are not associated with the received signal therefore reducing the effect of noise.

The Matched filter can also be implemented in the analogue or digital domain and is the optimum detection method for a pulse signal such as OOK transmitted across an ideal channel with additive white Gaussian noise. The perfect way of implementing this matching is by making the normalised gain response of the receiver filter $H_r(f)$ identical with the amplitude spectrum of the pulse $G_r(f)$ as in equation 6.1 [176]. This well known technique is abundantly documented in texts such as [173] and [176] and can easily be approximated in practical and simulated situations by integrate and dump circuitry; as the average power of the noise is zero, such circuitry maximises the signal to noise ratio in the same way.

$$|H_r(f)| = |G(f)| \tag{6.1}$$

The presence of ISI severely affects the performance of a matched filter if it is matched to the transmitter filter and channel distortion is not taken into consideration, since the matching will no longer be accurate. Whilst some other work only considers matching to the transmitter filter, regardless of channel distortion, all simulations here take this affect into account; affording the benchmark traditional systems the best performance under multipath channel conditions.

Often, standard low-pass, high-pass or band-pass filters are used to lessen the effects of noise in communication systems. For a simple indoor diffuse IR system a low-pass filter would be used to reject noise components at a higher frequency than the transmitted signal. Carruthers et al [25], suggest a normalised cut-off frequency of $0.6/T_b$ for unequalised OOK NRZ, where $1/\ T_b$ is the bit rate. Simulations with such filters on a basic threshold based receiver with a signal subjected to ISI from a multipath channel suggest that for this instance the optimum filter cut-off is nearer $0.7/\ T_b$. However, it can be seen from Fig. 6.4 that the difference in performance between the cut-off frequencies of $0.6/\ T_b$ and $0.7/\ T_b$ is marginal.

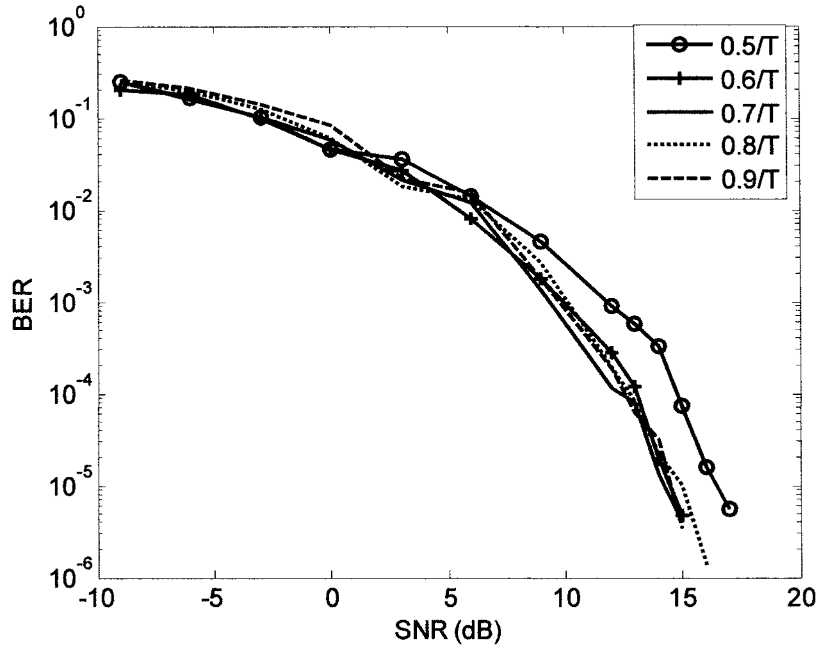


Fig. 6.4: BER against the SNR for 50 Mb/s OOK NRZ with various cut-off frequencies

Fig. 6.5 shows results from similar simulations for the OOK RZ case. Here it can be seen that the best cut-off frequency is nearer 0.35/ T_b; however, the differences are again marginal between adjacent filter cut-off frequencies.

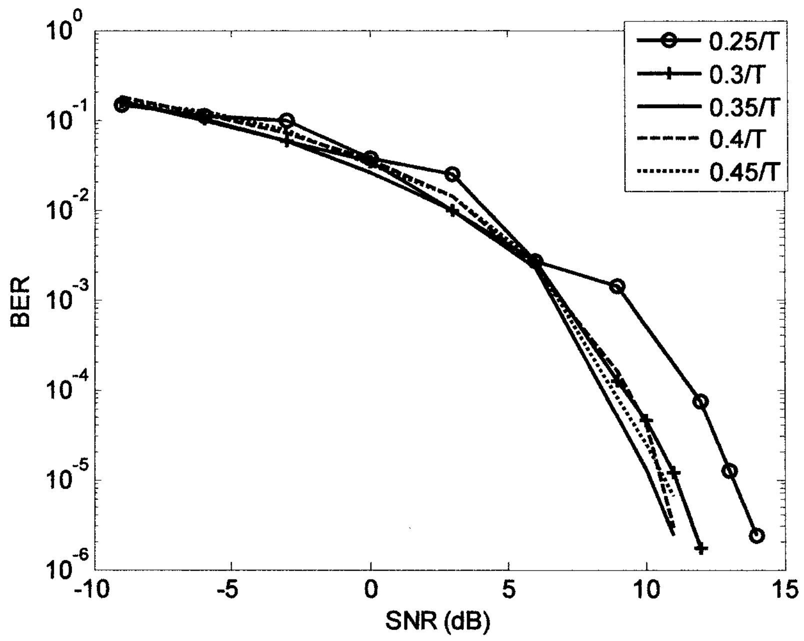


Fig. 6.5: BER against the SNR for 25 Mb/s OOK RZ with various cut-off frequencies

Fig. 6.6 depicts the performance of a threshold based receiver for both OOK-NRZ and OOK-RZ employing $0.6/T_b$ and $0.35/T_b$ respectively. It can be seen that the performance of lower data rates in the NRZ case improves dramatically, however, for higher data rates the performance is diminished with an irreducible BER being reached sooner than the unfiltered case shown in Fig. 6.3. Although the filtered RZ case is an improvement over the unfiltered case, its performance advantage over NRZ has disappeared completely at the lowest data rate simulated. Given these results, it is clear that simple low pass filtering alone cannot compensate for the effects of multipath distortion on shorter duration pulse modulation schemes such as OOK RZ and PPM.

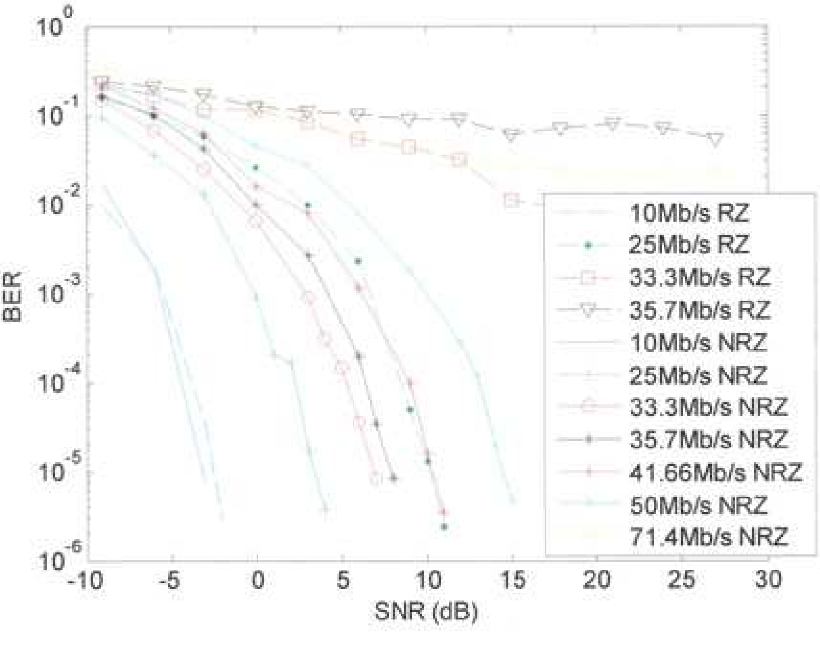


Fig. 6.6: BER against SNR for OOK NRZ/RZ with $0.6/T_b$ and $0.35/T_b$ low pass filtering

6.2.2 Coding

Coding techniques are concerned with the correction of errors within the detected data stream and not in enhancing the detection mechanism of the receiver itself. Such techniques rely on sending further information along with the intended data that allows an algorithm at the receiver end to detect and correct errors without retransmission; these methods being generically referred to as Forward Error Correction (FEC). This extra information serves no other purpose and is therefore redundant in a data sense and effectively reduces the data rate by some factor. Paradoxically, Claude Shannon showed in his work in the 1940s that the way to increase the data rate of a channel is by introducing these redundant bits.

Coding methods are broadly categorised into two types, block codes and convolution codes. In block codes a data word, an integer number of bits, is mapped onto a fixed size block consisting of the data or information bits and the check bits, each block being self contained and independent of other blocks. An (n,k) code has k data bits and n-k check bits; the rate of the code, defined as the proportion of information bits in the transmitted code sequence is then k/n [177]. The corrective power of a code is directly related to the number of check bits in the code word.

In a convolutional code the mapping of each code word depends not only on the current information block, but also on one or more previous blocks. This means the notion of a fixed length code word must be replaced by a code sequence, which could in theory be semi-infinite. Convolutional codes are described in a similar way to block codes, however, a third number is added to the description which relates to the number of input blocks that affect the current output block, this is known as the constraint length. Such schemes are very successful and widely implemented but add a processing delay and computational burden to the receiver.

A coding scheme can be added to virtually any communication system to improve performance, regardless of the receiver architecture, modulation type, filtering or other enhancement methods. The focus of this thesis is directed at primary detection at the receiver, therefore coding will not be considered in detail in this work.

6.2.3 Equalisation

Equalisation is concerned with compensating for imperfections in the channel. In the simplest case the equaliser filter has the inverse characteristics of the channel, essentially restoring the transmitted pulse shape at the receiver. There are two main methods and other related variations on equalisers that are frequently used in communication systems, these are the zero forcing equaliser (ZFE) and the decision feedback equaliser (DFE) both implemented as digital filters. A variation on the ZFE type of equalisation will form the benchmark for further work in this chapter.

6.2.3.1 The Zero Forcing Equaliser

The ZFE implemented as a transversal filter or FIR filter consisting of a taped delay line with input X_k, usually tapped at intervals of '*T*' seconds, where '*T*' is the symbol interval. The output of each tap is weighted by a variable gain factor a_N and each weighted output is summed to form the final output of the filter y_k at a particular time *kT*.

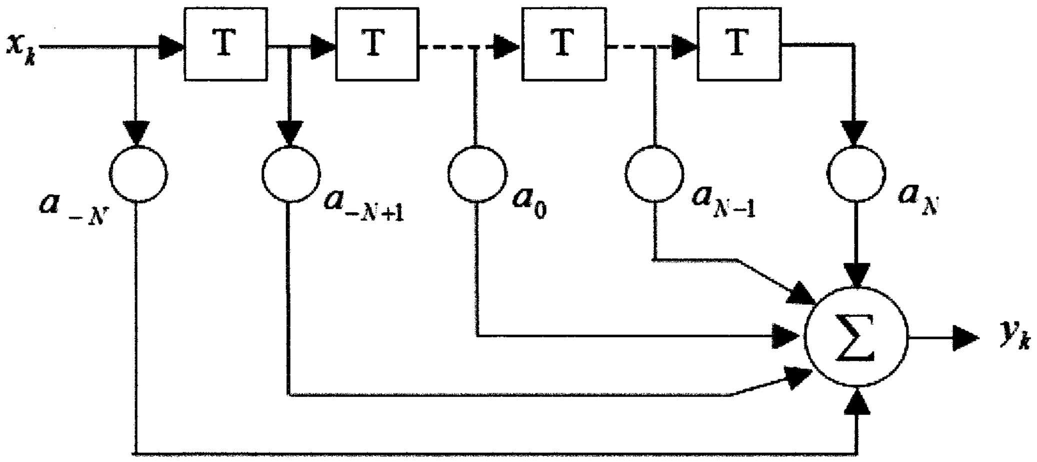


Fig. 6.7: Structure of the ZFE

There are (2*N*+1) taps where *N* is chosen large enough to span the ISI with corresponding weights $a_{-N} \dots a_0 \dots a_N$. Fig. 6.7 shows the basic structure of such a filter. The minimisation of ISI is achieved by considering only those inputs that appear at the correct sample times. For convenience $x(kT) = x_k$ and $y(kT) = y_k$. For zero ISI we require [173]:

$$y_k = \begin{cases} 1 \, for \; k = 0 \\ 0 \; elsewhere \end{cases} \tag{6.2}$$

The output y_k can be expressed in terms of the inputs and tap weights:

$$y_k = \sum_{n=-N}^{N} a_n x_{k-n} \tag{6.3}$$

Therefore:

$$y_k = \begin{cases} 1 \; when \; k = 0 \\ 0 \; when \; k = \pm 1, \pm 2, \dots \pm N \end{cases} \tag{6.4}$$

A transversal equaliser can force the output to go to zero at *N* points either side of the peak output and we can solve the (2*N* + 1) equations in matrix form as follows:

$$\begin{bmatrix} x_0 & x_{-N+1} & \cdots & x_{-N} & \cdots & x_{-2N+1} & x_{-2N} \\ x_{N-1} & x_0 & \cdots & x_{-N+1} & \cdots & x_{-N} & x_{-2N+1} \\ \vdots & \vdots & & & & & \\ x_N & x_{N-1} & \cdots & x_0 & \cdots & x_{-N+1} & x_{-N} \\ \vdots & \vdots & & & & & \\ x_{2N-1} & x_N & \cdots & x_{N-1} & \cdots & x_0 & x_{-N+1} \\ x_{2N} & x_{2N-1} & \cdots & x_N & \cdots & x_{N-1} & x_0 \end{bmatrix} \begin{bmatrix} a_{-N} \\ a_{-N+1} \\ \vdots \\ a_0 \\ \vdots \\ a_{N-1} \\ a_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{6.5}$$

$$\boldsymbol{a} = \boldsymbol{X}^T \boldsymbol{q} \tag{6.6}$$

Where $\boldsymbol{a}$ is the filter coefficient array, $\boldsymbol{X}$ is the sample point matrix and $\boldsymbol{q}$ is the output array.

A major drawback of the ZFE is that it ignores the presence of additive noise and its use may result in significant noise enhancement. This can be understood by noting that equaliser filter response is the inverse of the channel characteristics $Ge(f) = 1/C(f)$; so the equaliser places large gains where $C(f)$ is small, boosting noise in the process.

Relaxing the zero ISI condition and selecting the channel equaliser characteristic such that the combined power in the residual ISI and the additive noise at the equaliser output is minimised can improve the performance of a ZFE. An equaliser that is optimised based on the minimum mean square error (MMSE) criterion achieves this [178]. Although the characteristics of an indoor IR channel are quasi-static they are not uniform for any given receiver transmitter pair. For example moving a receiver from one part of a room to another will alter the impulse response $h(t)$ of the channel as seen by the receiver. We cannot therefore embody a common set of filter coefficients in the receiver and it is common for a training pulse to be transmitted so that the filter tap weights can be adjusted accordingly. Such a filter is usually referred to as a pre-set equaliser as the coefficients are calculated prior to transmission. A further variation on the ZFE is the adaptive equaliser where the tap weights are adjusted with time to compensate for fluctuations in channel response.

6.2.3.2 The Minimum Mean Square Error Equaliser

Relaxing the zero ISI condition and selecting the channel equaliser characteristic such that the combined power in the residual ISI and the additive noise at the equaliser output is minimised can improve the performance of a ZFE. Based on classical equalisation theory, the most common cost function is the mean squared error between the desired signal and the output of the equaliser [177]. An equaliser that minimises this cost function is therefore known as the MMSE as shown in Fig. 6.8.

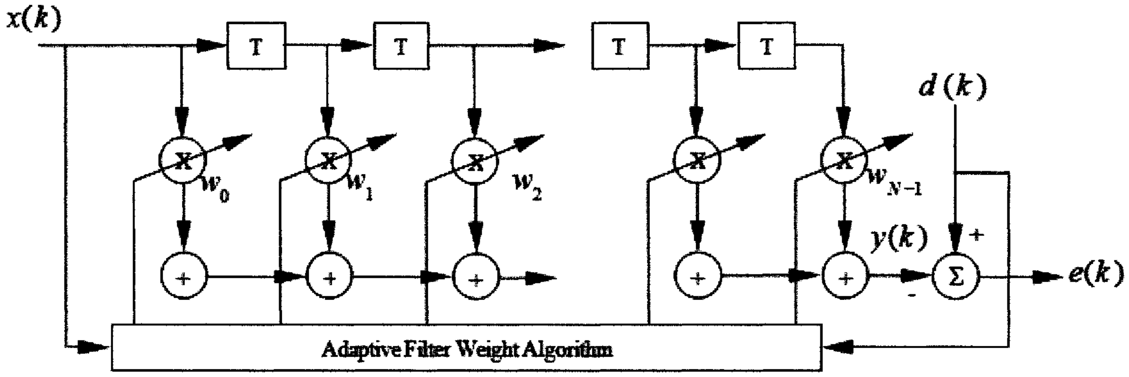


Fig.6.8: Structure of the minimum mean square error equaliser

Once again this type of equaliser relies on a known training sequence being periodically transmitted, this time so that the algorithm can minimise the mean square error.

The output from the equaliser is given as [181]:

$$y(k)=\sum_{n=0}^{N-1} w_n x(k-n) \tag{6.7}$$

Taking the expected '*E*' or mean values of the squared error between the output and training sequence, dropping the '*k*' for notational clarity and writing in terms of the correlation matrix **R** and the cross correlation vector **p** gives:

$$E[e^2(k)]=E[d^2(k)]+\mathbf{w}^T\mathbf{R}\mathbf{w}-2\mathbf{w}^T\mathbf{p} \tag{6.8}$$

Where

$$\mathbf{R}=E[\mathbf{x}\mathbf{x}^T]=E\begin{bmatrix} x_k \\ x_{k-1} \\ x_{k-2}\end{bmatrix}\begin{bmatrix} x_k & x_{k-1} & x_{k-2}\end{bmatrix}=E\begin{bmatrix} (x_k^2) & (x_k x_{k-1}) & (x_k x_{k-2}) \\ (x_{k-1}x_k) & (x_{k-1}^2) & (x_{k-1}x_{k-2}) \\ (x_{k-2}x_k) & (x_{k-2}x_{k-1}) & (x_{k-2}^2)\end{bmatrix} \tag{6.9}$$

And

$$\mathbf{p}=E[d_k\mathbf{x}_k]=E\begin{bmatrix} d_k x_k \\ d_k x_{k-1} \\ d_k x_{k-2}\end{bmatrix}=\begin{bmatrix} p_0 \\ p_1 \\ p_2\end{bmatrix} \tag{6.10}$$

In general for an 'N' weight filter:

$$\mathbf{p}=\begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{N-1}\end{bmatrix} \tag{6.11}$$

The optimum weight vector is then found by setting the partial derivative gradient vector to zero (i.e. $\nabla = 2\mathbf{R}\mathbf{w}-2\mathbf{p}=0$), therefore:

$$\mathbf{w}_{opt}=\mathbf{R}^{-1}\mathbf{p} \tag{6.12}$$

The MMSE in conjunction with the matched filter forms the basis for a number of traditional receivers used for performance comparison with the Wavelet-AI receivers in this chapter.

6.2.3.3 The Decision Feedback Equaliser

A possible improvement on the ZFE and MMSE is the Decision Feedback Equaliser (DFE). This non-linear device uses previous decisions to counter the ISI caused by the previously detected symbol on the current symbol to be detected. The DFE consists of two filters, a feed-forward filter (pre-filter) and a feed-back filter (ISI estimator) [177].

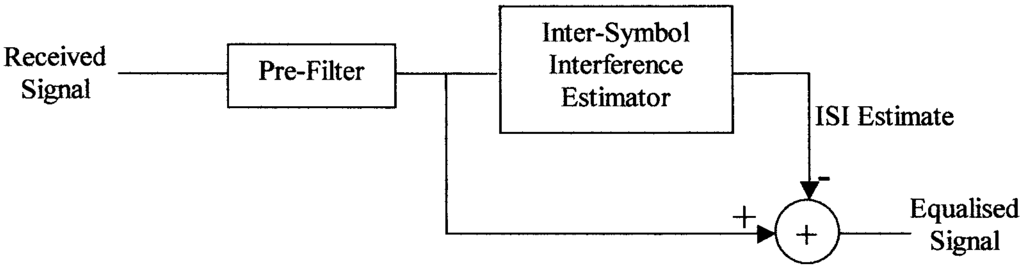


Fig. 6.9: Schematic DFE Structure

The feed-forward filter is generally a fractionally spaced FIR filter with adjustable tap coefficients and has identical form to the linear ZFE [178]. The feed-back filter is an FIR filter with symbol spaced taps having adjustable coefficients, its input being the set of previously detected symbols. The output of the feed-back filter is subtracted from the output of the feed-forward filter to form the input to the detector. Fig. 6.9 shows a block diagram representation of such a device.

The DFE does suffer performance degradation from feed-back errors; however, the errors do not cause catastrophic failure and are at relatively high BER, out of the usual operating area. Due to time limitations implementation of the DFE is not included in this work.

6.2.4 Diversity Techniques

Due to the problems of high ambient noise and multipath dispersion inherent in diffuse IR systems and the limitation of traditional techniques, researchers in the field are exploring alternative methods of mitigating these effects including; angle diversity [46, 52], multi-spot systems [38, 41, 51, 53, 54], sectored receivers [43 - 45] and code combing [179, 180] to name a few. Most of these schemes seek to eliminate noise or ISI by limiting the field of view of the optical receiver, cutting out noise and limiting the available paths to the receiver thus reducing ISI. Such techniques could potentially lead to severe shadowing if one transmitter – receiver pair was used in such systems. To overcome this it is usual to employ several transmitting beams or receiver elements. These schemes undoubtedly have the potential to offer performance improvements, but at the expense of system complexity, more difficult deployment, and additional hardware cost.

This work in this thesis is largely concerned with investigating alternative mitigation techniques in the electrical domain of diffuse optical system, and it is envisaged that these alternatives could be embodied in many diversity schemes to improve their performance further. It is therefore not the intention of this work to conduct a detailed review of diversity techniques.

6.2.5 Wavelet AI Receiver

The core of the Wavelet-AI receiver architecture does not directly rely on amplitude detection for threshold decision making and therefore inherently compensates for multipath induced ISI. Fig. 6.10 shows a 31.25 Mb/s OOK RZ signal subject to a D_{RMS} value of 8.19x10^{-9} s, and Fig. 6.11 is a three dimensional plot of the corresponding wavelet coefficients from a CWT with scales from 0 to 60. From observation it can be seen that some correlation exists between the two plots. The pattern recognition ability of a feed-forward back-propagation neural network is employed to map the features of the wavelet coefficients to a particular binary decision.

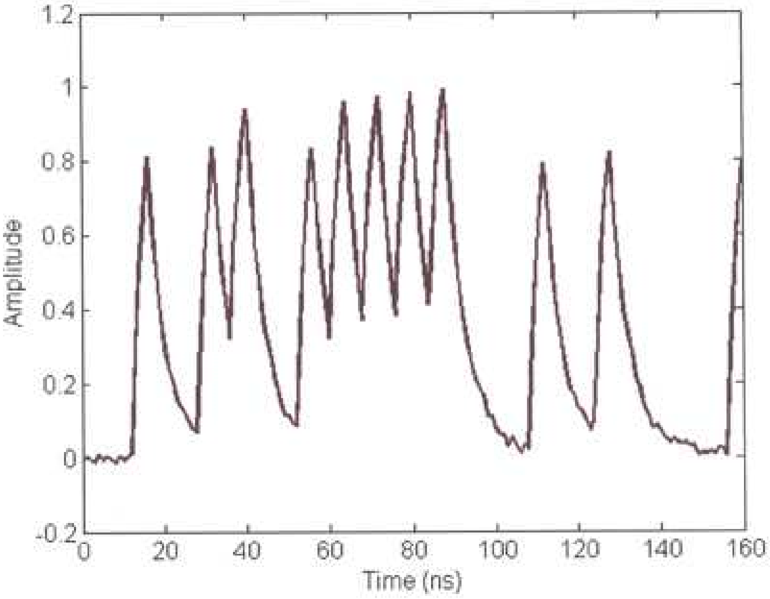


Fig. 6.10: 31.25 Mb/s OOK RZ signal with ISI

In chapter 5 sections 3.2 and 3.4 provide details of specific implementations of the Wavelet-AI architecture for OOK RZ and PPM, respectively. The same configurations are used here; the exception to this being the scales and number of training symbols used. The Wavelet-AI architecture is therefore universal for the scenarios modelled and simulated in this work

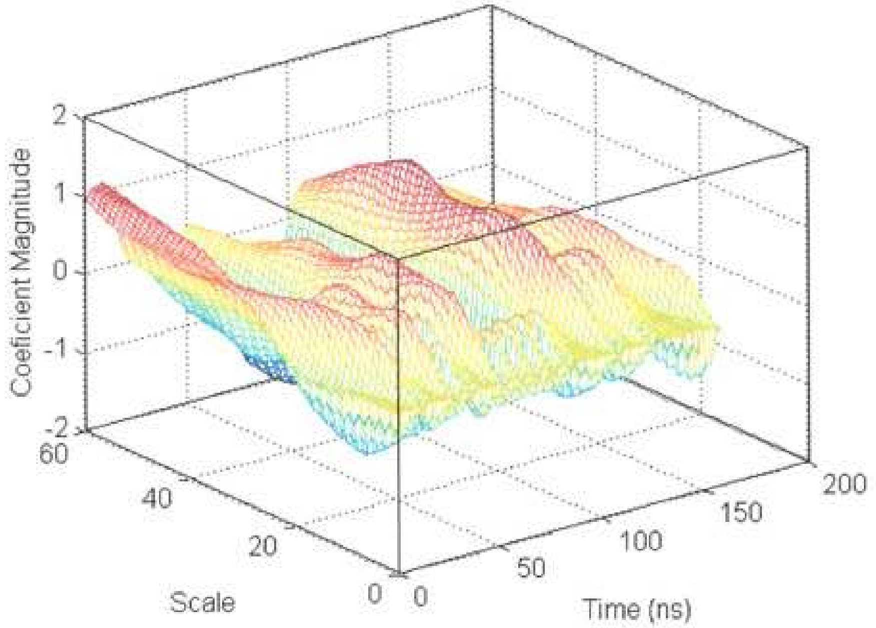


Fig. 6.11: Wavelet coefficients from a CWT of Fig. 6.10

6.3 The Diffuse Indoor IR Channel Model

The model used in simulations undertaken in this report is based on the well-cited work of Carruthers and Kahn [25] where a dimensionless parameter known as the normalised delay spread, D_T, (RMS delay spread / bit duration) is used as a measure of the ISI a data stream contains. This is a two stage modelling approach where firstly the delay spread seen by a co-located transmitter-receiver pair under an infinite plane is calculated. Secondly, a correction is made to model based on the actual separation of transmitter and receiver within a particular sized room.

The delay spread is given by:

$$D_{RMS}(h(t,a)) = \frac{a}{12}\sqrt{\frac{13}{11}} \tag{6.13}$$

Where

$$a = {}^{2H}\!\big/_{c} \tag{6.14}$$

and H is the height between Tx/Rx and ceiling, and c is the speed of light.

The correction of a is given by:

$$a(unshadowed) = 12\sqrt{{}^{11}\!\big/_{13}}(2.1 - 5.0s + 20.8s^2) \times D(h_1(t)) \tag{6.15}$$

Where s is related to the ratio of the distance between the receiver transmitter and a diagonal that intersects them both and extends to the walls of the room under consideration.

The Impulse Response is given by:

$$h(t,a) = \frac{6a^6}{(t+a)^7}u(t) \tag{6.16}$$

Where *u(t)* is the unit step response.

If we now consider the same multipath signal as in Fig. 6.1 and add noise providing a SNR of 12 dB as shown in Fig. 6.12, we can see that the task of bit detection by a simple threshold detector would be unrealistic for high data rates. The corresponding eye diagram of Fig. 6.13 confirms that the 'eye' has closed completely.

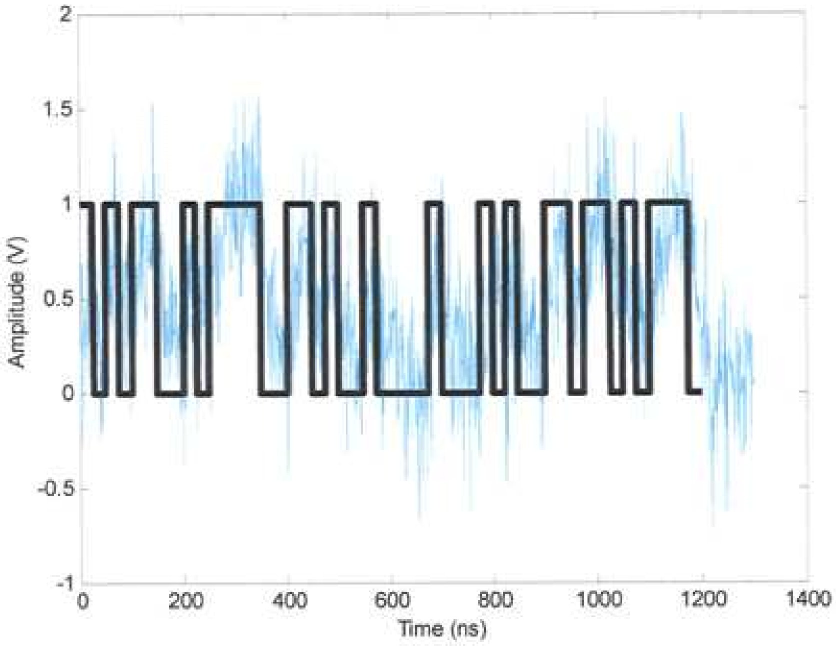


Fig. 6.12: Noise and multipath distortion

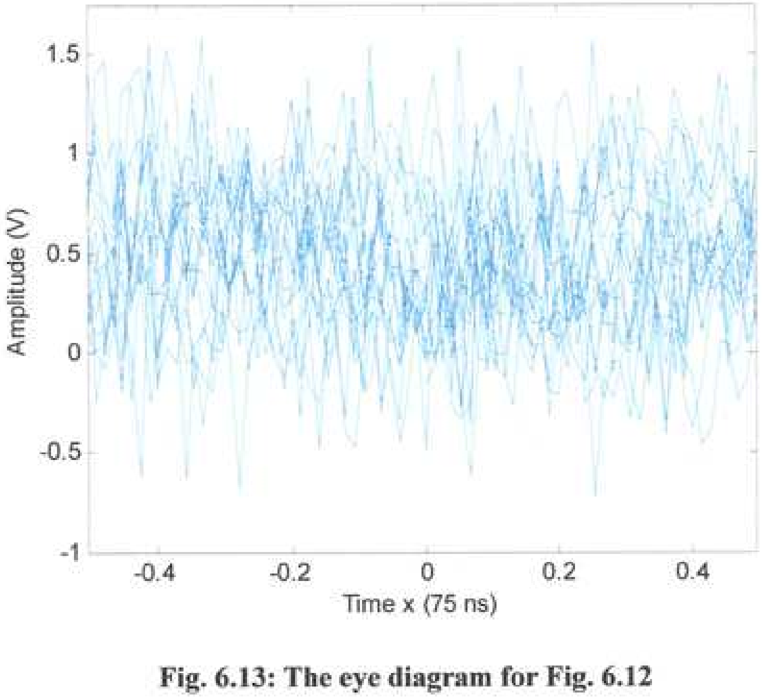


Fig. 6.13: The eye diagram for Fig. 6.12

For any given delay spread an increase in data rate, resulting in shorter pulses, will increase the effects of ISI and degrade the performance of a simple threshold detection receiver. The degradation caused by ISI will reach a point where the BER is irreducible regardless of the amount of transmission power. Fig. 6.3, shown in the introduction to this chapter, confirms that in comparison to OOK NRZ, the shorter pulse OOK RZ modulation scheme loses its power advantage for data rates beyond 10 Mb/s. Indeed, it is the NRZ case that shows a power advantage beyond 25 Mb/s thus negating the use of shorter duty modulation schemes in unequalised systems.

6.4 Unequalised Performance of OOK and PPM

This section looks at the unequalised performance of OOK RZ $\gamma = 0.5$, 4-PPM and 8-PPM at various data rates for a channel with and without multipath distortion. Where multipath distortion is present a D_{RMS} value of $8.19\text{x}10^{-9}$ s is used constantly throughout the remainder of this chapter. The optical power penalty is normalised to that power requirement of an OOK RZ $\gamma = 0.5$, matched filter receiver, shown in Fig. 6.14, to achieve a BER of 10^{-6} for a data rate of 2.5 Mb/s. All models throughout this chapter are based on a digital architecture and assumptions made in chapter 5, section .3.1, (i) are also valid in this section; the exception to this being that throughout this chapter the sampling interval employed by the ADC is 4 ns. In a similar manner to the assumption from chapter 5, section 3.2 (ii), the anti-alias filter of all models shown in this section have been omitted and band limited white noise is added using the Matlab AWGN function. This function, given the signal under consideration adds white Gaussian noise to the prescribed SNR level.

6.4.1 OOK Systems

For OOK the results are restricted to the OOK RZ case for both the Wavelet-AI and traditional receiver structures. The Traditional structure is shown in Fig. 6.14 and the Wavelet-AI structure is shown in Fig. 6.16. This is similar to the traditional model up to the ADC, and in both cases the multipath channel may be omitted.

6.4.1.1 Traditional OOK

The simulation model for the traditional system is shown in Fig. 6.14. The input bits (1 or 0) are passed to the transmitter filter that has a unit amplitude rectangular impulse response $p(t)$ with a duration $T_b/2$. The pulses are scaled by the transmitted optical signal power, in this case $4Pavg$, where $Pavg$ is the average transmitted signal power. The signal is then passed through the multipath channel with an impulse response $h(t)$, given by equation 6.16. The received optical power is scaled by the photodetector responsivity, taken to be unity in this chapter. White Gaussian noise is added before the

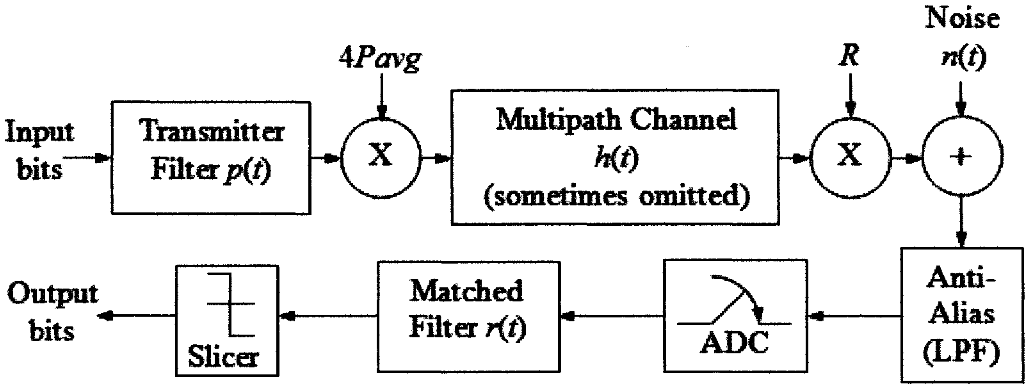


Fig. 6.14: OOK RZ traditional receiver schematic

signal is passed through an analogue anti-alias filter, which would be essential in a practical system. The signal is then passed to a unit energy matched filter, with impulse response $r(t)$ which is matched to $p(t)$ for the non-multipath case. The matched filter is sampled once at the end of the bit period and the result passed to a threshold detector, which outputs a one or zero depending on the level of the incoming signal. In the above case the matched filter $r(t)$ is only optimum for a non-multipath signal since increasing ISI would progressively reduce the matching between the $p(t)$ and $r(t)$. In the case where multipath dispersion is present the model changes and $r(t)$ is matched to the incoming pulse shape after being subjected to multipath distortion, thus helping preserve the performance of the traditional system.

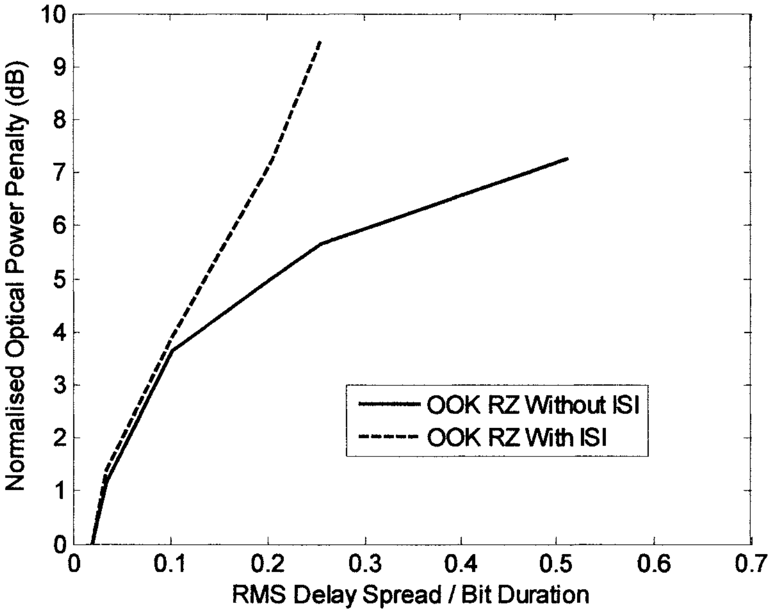


Fig. 6.15: Normalised optical power penalty against D_T for OOK RZ with and without ISI for a BER of 10^{-6}

The simulation results from the model of Fig. 6.14 are shown in Fig. 6.15. It can be observed that from a D_T of approximately 0.1 the curves become divergent, with performance of the OOK system being limited by the ISI. In fact individual simulations show that performance tails off rapidly and the BER quickly becomes irreducible regardless of the transmitted power and the matched filter being optimised to the incoming pulse shape. This phenomenon is perhaps more easily observed from Fig. 6.3, where BER curves for higher data rates do not improve despite an increase in SNR.

6.4.1.2 Wavelet AI OOK

In a similar manner to chapter 5 the Wavelet-AI receiver diagram of Fig. 6.16 looks superficially similar to its traditional counterpart; however, after the ADC the functionality of the two models differs markedly. The program flowchart for the simulation model is shown in Fig. 6.17 and the program listing in appendix D. The Wavelet-AI receiver works as described fully in chapter 5, section 3.2: The Wavelet analysis section extracts the coefficients of pre-determined scales from a sliding 3 bit window. The coefficients are then passed to a 2 layer 6 neuron feed-forward, back-propagation neural network with 1 output neuron. The output feeds into a slicer that forces a binary decision based on the level of the signal. Prior to data transmission a known series of bits are transmitted and the neural network output is trained provide a value close to one or close to zero to match the intended data. The 'Symlet 2' wavelet is used in all cases in this chapter and for this section there are 800 training bits for all values of D_T. The CWT scales employed for the simulations with respect to the various values of D_T are shown in Table 6.1.

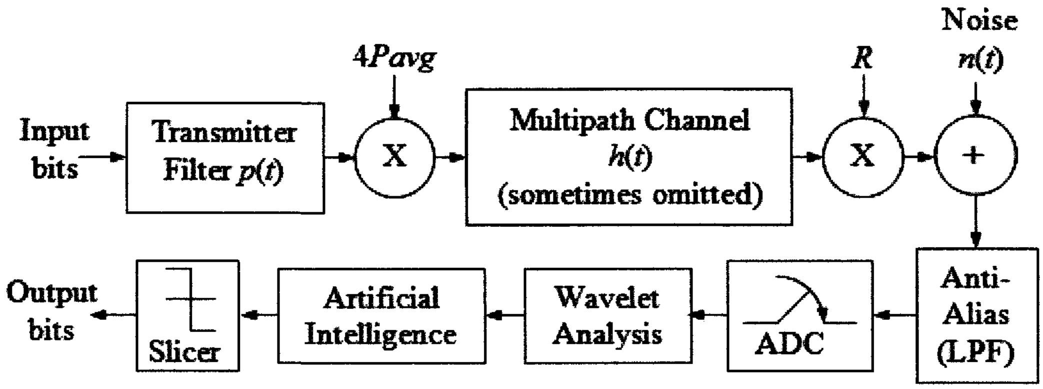


Fig. 6.16: OOK RZ Wavelet AI receiver schematic

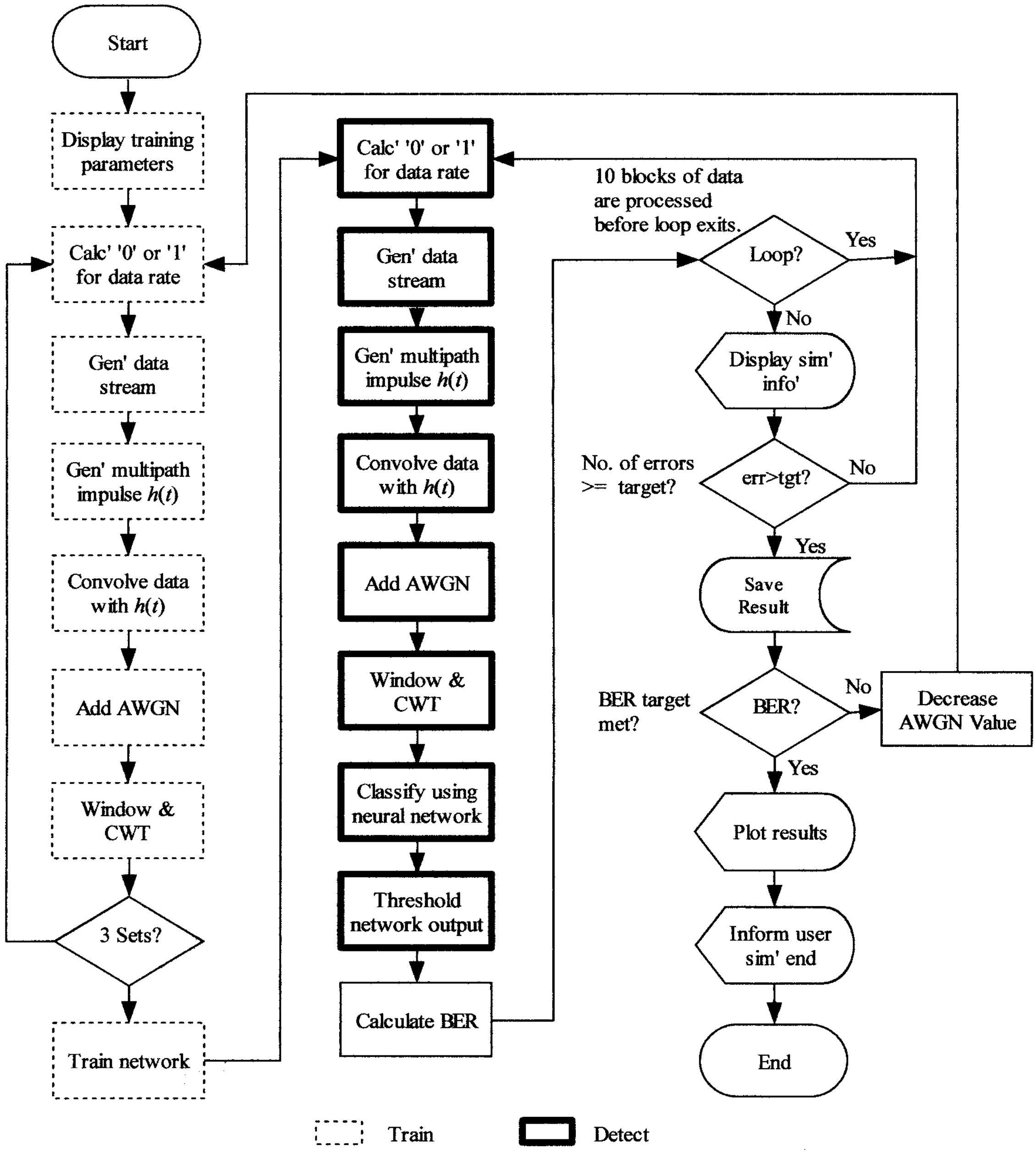


Fig 6.17: Flowchart for simulation model shown in Fig. 6.16

Table 6.1: Wavelet scales OOK RZ detection.

D_T	Scales
2.05×10^{-2}	76, 170, 270
3.41×10^{-2}	15, 30, 45, 60, 75, 90, 105, 125, 150, 175, 200
1.02×10^{-1}	5, 10, 15, 20, 25, 30, 35, 40, 50, 60, 70, 80, 90, 100
2.05×10^{-1}	3, 6, 9, 12, 15, 18, 20, 30, 40, 50, 60
2.56×10^{-1}	5, 10, 15, 20, 25, 30, 35, 40, 50, 75, 100, 125, 150, 175, 200
5.12×10^{-1}	3, 6, 9, 12, 15, 25, 35, 45, 55, 65, 75, 85, 95, 105

The simulation results from the Wavelet-AI model of Fig. 6.16 are shown in Fig. 6.18. In contrast with the results of the traditional system the performance curves do not diverge as rapidly and offer broadly similar performances for both the ISI and non ISI case up to a D_T of approximately 2.5. For higher values of D_T the curves diverge more noticeably.

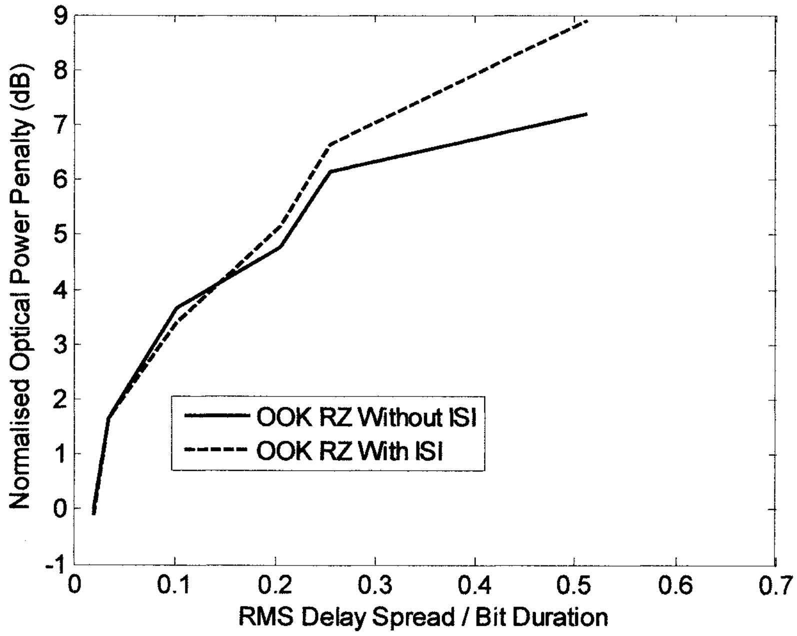


Fig. 6.18: Unequalised performance of Wavelet-AI OOK system for a BER of 10^{-6}

6.4.2 PPM Systems

In this section receiver architectures for detecting 4-PPM and 8-PPM are considered. For the traditional case the model is based on a PPM Threshold (TH) detector using a matched filter. For the Wavelet-AI case a model based on the Wavelet-AI Binary Symbol Detector (WBSD) described in chapter 5, section 3.4.1.

6.4.2.1 Traditional PPM

The traditional receiver model is shown in Fig. 6.19. In this system the PPM encoder converts each segment of Log_2L input bits onto one of the possible L symbols. The transmitter and channel are similar to those described for OOK; the transmitter filter having unit amplitude rectangular impulse response $p(t)$ with a duration of 1 slot T_s, where $T_s = T_b \log_2L/L$. The output of the transmitter is scaled by the peak transmitted optical signal power *LPavg* and passed through the multipath channel (sometimes omitted) $h(t)$. The received photocurrent is scaled by the photodetector responsivity, taken to be unity in this chapter. White Gaussian noise is added to the signal using the Matlab AWGN function. The signal is passed through a band limiting anti-alias filter before being passed to a unit energy filter with impulse response $r(t)$, matched to $p(t)$. In these simulations $r(t)$ is only matched to $p(t)$ for the case when $h(t)$ represents a perfect channel and introduces no multipath induced distortion. For the ISI case $r(t)$ is matched to the incoming pulse after being subjected to $h(t)$ in order to improve performance of the traditionally based receiver.

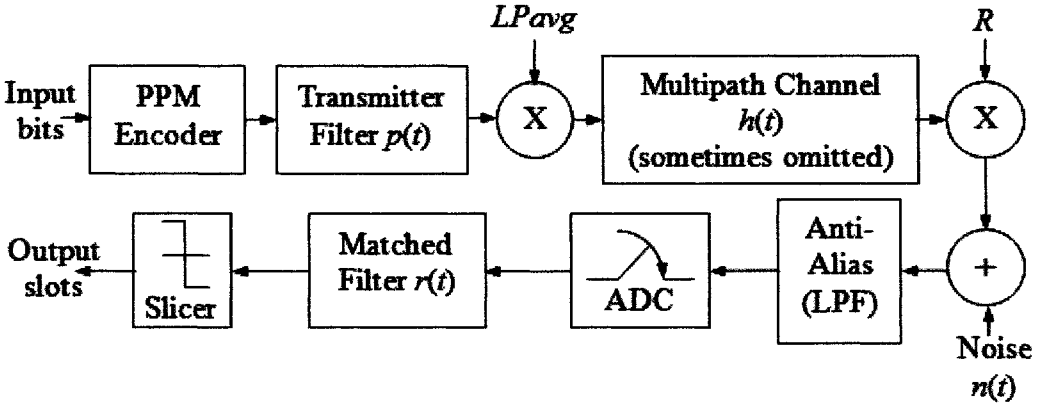


Fig. 6.19: *L*-PPM (TH) receiver schematic

The simulation results for various levels of D_T for both the 4-PPM and 8-PPM case are shown in Fig. 6.20. In a similar way to the OOK case it is immediately apparent that for both 4-PPM and 8-PPM the multipath distortion limits BER performance. In the PPM case the curves diverge earlier for OOK RZ and individual simulations show that the BER quickly becomes irreducible well below a D_T of 0.2. This outcome is entirely as expected from the power efficient but shorter pulse durations of PPM. Also as expected, the non ISI curves show that that 8-PPM is more power efficient, encoding 3 bits per symbol rather than the 2 bits for the 4-PPM case.

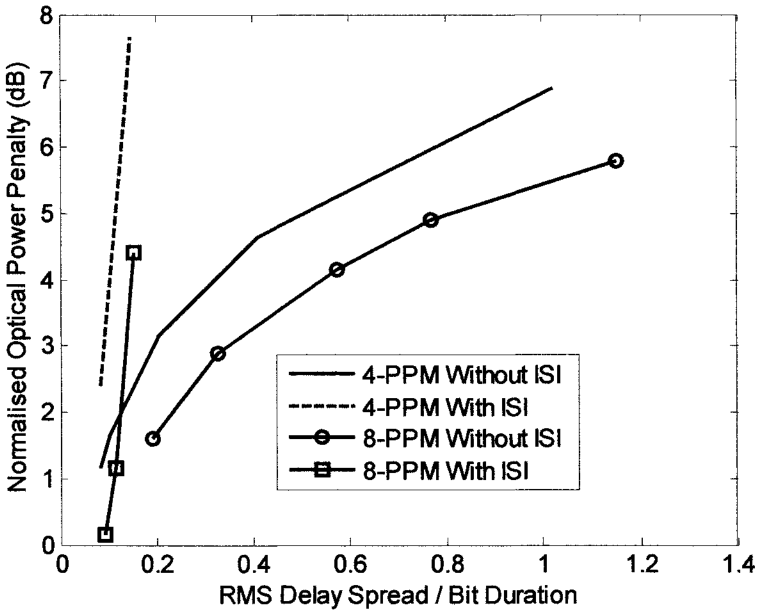


Fig. 6.20: Unequalised performance of 4-PPM and 8-PPM for a BER of 10^{-6}

6.4.2.2 PPM Wavelet AI Binary Symbol Detector (WBSD)

The PPM Wavelet-AI symbol detector model of Fig. 6.21 is very similar in operation to that described in chapter 5, section 3.4.1. The flowchart for the simulation model is shown in Fig. 6.22, and the program listing in appendix E. Up to the ADC, the model is identical to the traditional PPM case. Again the PPM encoder converts each segment of *Log₂L* input bits onto one of the possible *L* symbols. The transmitter filter has unit amplitude rectangular impulse response *p*(*t*), with a duration of 1 slot T_s where $T_s = T_b$ *log2L/L*. The output of the transmitter is scaled by the peak transmitted optical signal power *LPavg* and passed through the multipath channel *h*(*t*) (sometimes omitted). The received photocurrent is scaled by the photodetector responsivity, taken to be unity in all cases in this chapter. White Gaussian noise is added to the signal using the Matlab AWGN function. The signal is low pass anti-alias filtered before continuing to the

Wavelet Analysis section. Here, coefficients of pre-determined scales are produced using the 'Symlet 2' wavelet on a 3 window sample block as shown in Fig. 5.19, each window being 3 symbols long, the 3 window sample containing 9 symbols in total. The coefficients are passed to a neural network structured as Fig. 5.20, which determines the symbol value of the centre symbol of the centre window on the 3 window sample. For the 4-PPM case the network has 12 neurons in the first layer, and 2 in the output layer corresponding to the L=4 binary output '0 0', '0 1'...'1 1'. Prior to data transmission a known series of symbols are transmitted and the neural network output is trained provide a value close to one or close to zero to match the intended data. The scales used for the simulation are shown in Table 6.2.

Table 6.2: Wavelet scales 4-PPM detection.

D_T	**Scales**	**Training Symbols**
8.19X10^{-2}	10, 20, 30...300	1000
1.02x10^{-1}	10, 20, 30...300	1000
2.05x10^{-1}	5, 10, 15...300	1600
4.1x10^{-1}	3, 6, 9...18 + 20, 30, 40...60	1600
1.02	3, 6, 9...18 + 20, 25, 30...60	1600
2.05	1, 2, 3...20 + 25, 30, 35...50	1600

For the 8-PPM case the network has 12 neurons in the first layer, and 3 in the output layer corresponding to the 8-PPM binary output '0 0 0', '0 0 1'...'1 1 1'. Prior to data transmission a known series of symbols are transmitted and the neural network output is trained provide a value close to one or close to zero to match the intended data. The scales used for the simulation are shown in Table 6.3.

Table 6.3: Wavelet scales for 8-PPM detection.

D_T	Scales	Training Symbols
1.92×10^{-1}	30, 40, 50...210	1000
3.29×10^{-1}	7, 14, 21 + 35, 45, 55...145	1000
5.75×10^{-1}	15, 25, 35...90	1600
7.68×10^{-1}	1, 2, 3...61	1600
1.15	1, 2, 3...20 + 25, 30, 35...60	1600
2.3	1, 2, 3...20 + 25, 30, 35...60	1600

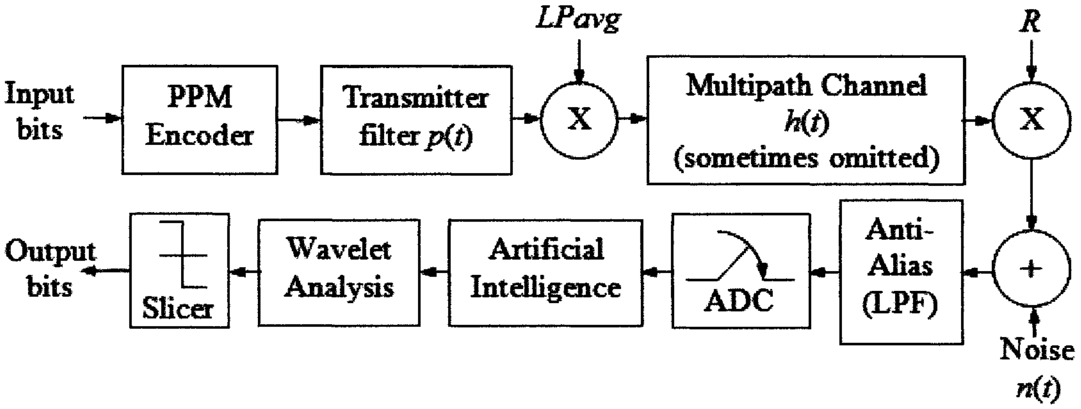


Fig. 6.21: PPM WBSD receiver schematic

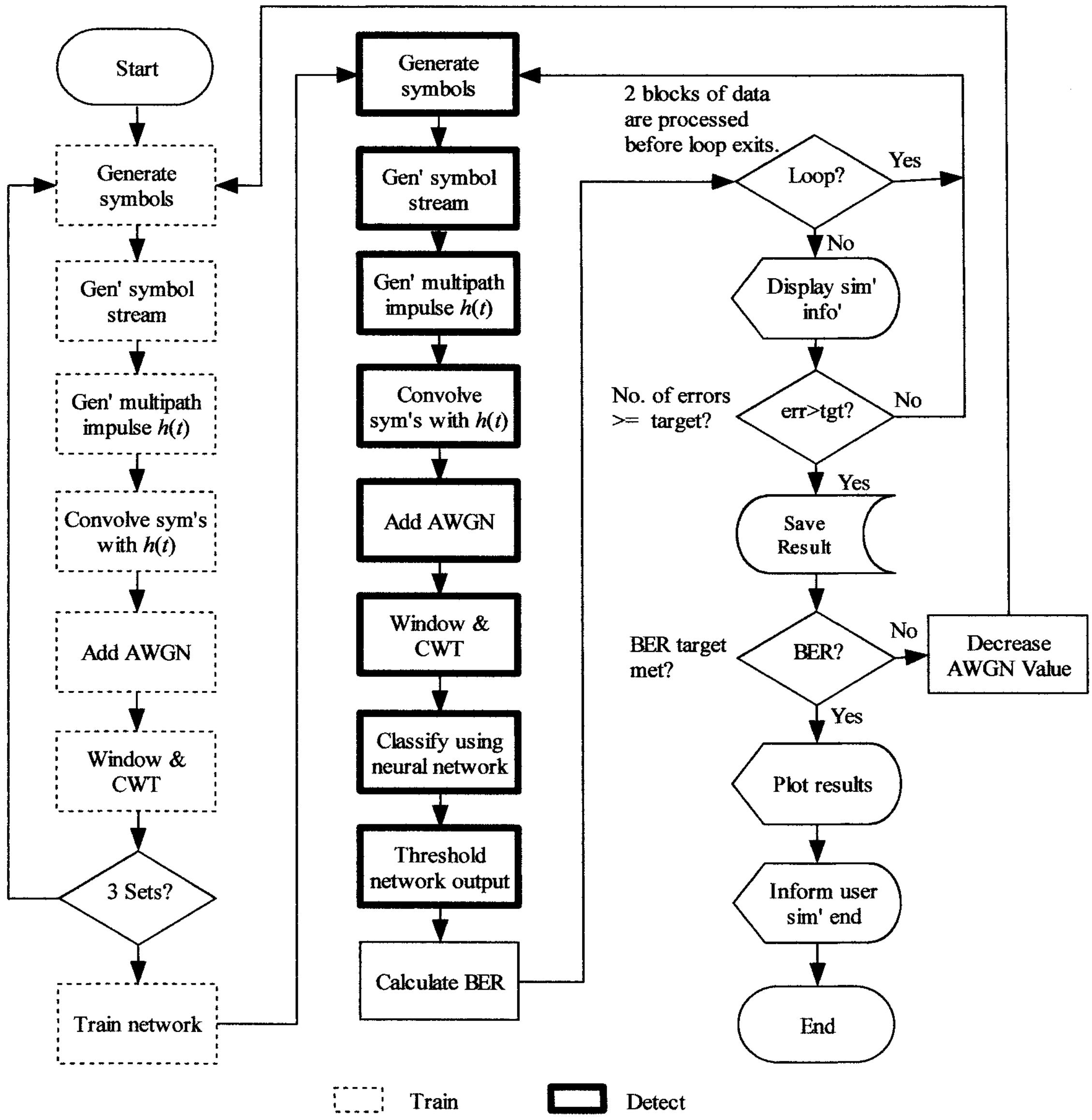


Fig. 6.22: flowchart for simulation model shown in Fig. 6.21

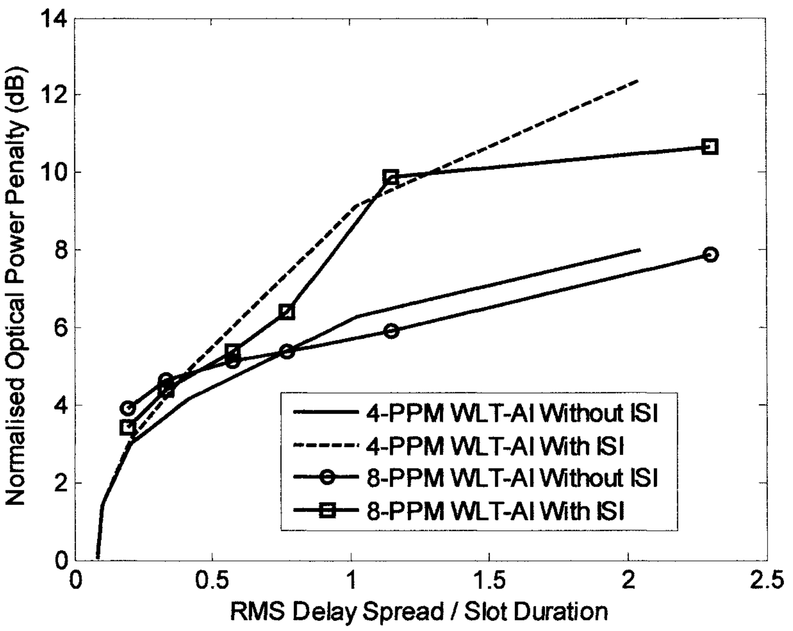


Fig. 6.23: Unequalised Performance of WLT-AI BSD for 4-PPM and 8-PPM for a BER of 10^{-6}

For cases without ISI, the plot of Fig. 6.23, shows a marginal improvement for the 8-PPM performance over the 4-PPM for D_T greater than approximately 0.7. Below this the 8-PPM case shows marginally worse performance. Compared to the traditional 4-PPM receiver performance of Fig. 6.20, the 4-PPM WBSD case shows an improvement approaching 1dB for a D_T Of 1 and a marginally better performance for 8-PPM when ISI is not present. For cases with ISI the performance of the WBSD receiver performance contrasts significantly with that of the traditional architecture. For D_T values well below 0.2, Fig. 6.20 shows the non-ISI and ISI curves diverging rapidly, with the ISI curves quickly rising indicating large optical power penalties for small increases in D_T. Individual simulations indicated that a small increase in data rate lead to irreducible BER figures regardless of the signal power used. The WBSD curves show divergence between the non-ISI and ISI cases happen much slower. Within the limits of D_T simulated the ISI performance curves do not show rapidly increasing optical power penalties for small increases in D_T, and no individual simulation indicating an irreducible BER. At a D_T of 0.2 the 8-PPM case with ISI shows an almost co-incident performance with the 4-PPM case, whilst most other points show some improvement. The cause of this anomaly is yet unproven, however, similar events are not uncommon in Wavelet-AI simulations and there are a number of candidate reasons, which will be discussed later. At high values of D_T there is an approximately 4 dB power penalty between the ISI and non ISI case for 4-PPM and approximately 3dB for 8-PPM.

6.5 Equalised Performance of OOK and PPM

This section essentially introduces traditional mitigation techniques into the simulation models for minimising the effects of ISI on BER. No further modifications are required to the Wavelet-AI model since the architecture inherently compensates for the variations resulting from multipath induced ISI. The remaining sections therefore compare the performance of equaliser enhanced traditional architectures with the previously simulated performance of the Wavelet-AI model subjected to ISI. In all cases the CWT scales used in the simulations were identical to the previous sections. It is therefore possible that further optimisation of scale selection for the ISI case could further enhance model performance.

6.5.1 OOK Systems (Traditional and Wavelet-AI)

The simulation model for the equalised version of the traditional OOK receiver architecture is shown in Fig. 6.24. The flowchart for the model is shown in Fig. 6.25 and the program listing on the accompanying CD. The operation is similar to that described earlier in the chapter. However, in this case the unit energy matched receiver is always matched to the incoming (sampled) pulse shape in all cases. The model also includes a MMSE as described in section 6.2.3.2 to diminish as far as possible the effects of the multipath channel. Filter coefficients were calculated for all simulations that involved changes in D_T or SNR, essentially a calculation for every simulation run. A sample impulse response of a filter used for 62.5Mb/s OOK is shown in Fig. 6.26. For all simulations in this chapter the equaliser filters have 19 coefficients.

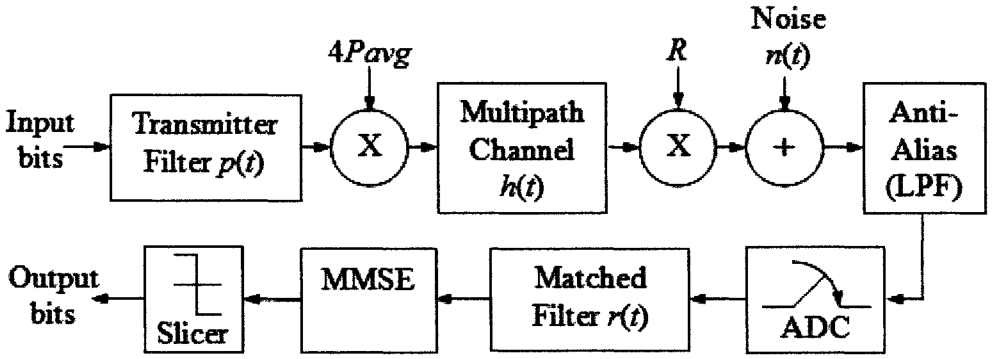


Fig. 6.24: Equalised OOK RZ schematic

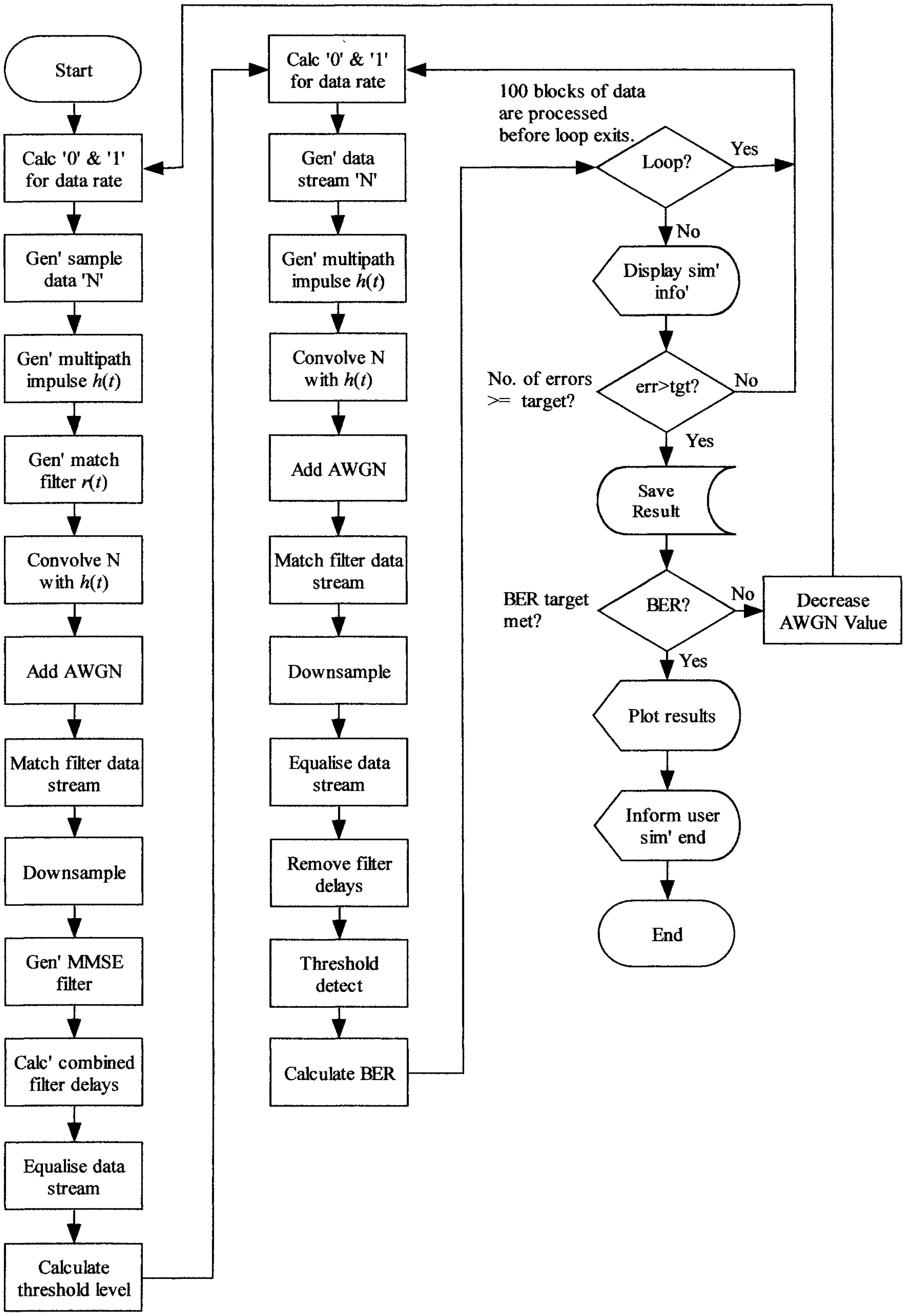


Fig. 6.25: Flowchart for the simulation model shown in Fig 6.24

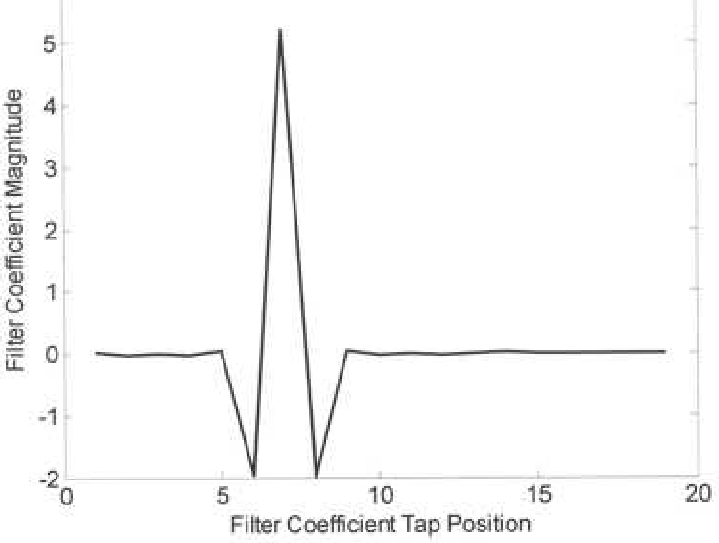


Fig. 6.26: MMSE filter impulse response for OOK RZ 62.5 Mb/s

The inclusion of the MMSE improves the performance of the system in the presence of ISI and its effects can easily be seen by inspection of the corresponding before and after 'eye diagrams' of Fig. 6.27 and Fig. 6.28, respectively. Fig. 6.28 depicts a largely restored 'eye opening' affording amplitude and time tolerance for reliable threshold detection

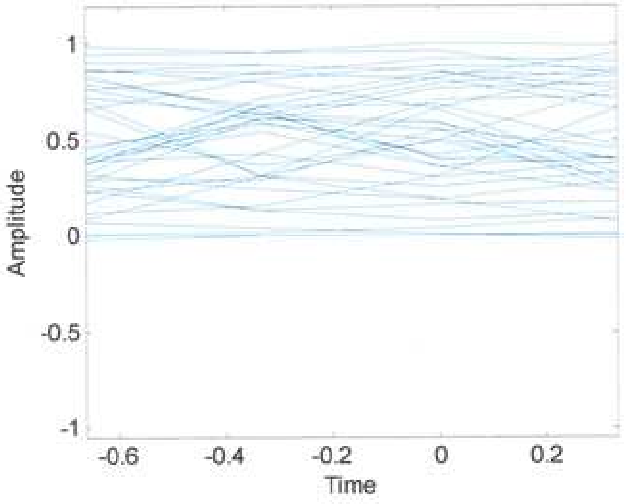


Fig. 6.27: The eye diagram for unequalised OOK RZ 62.5 Mb/s

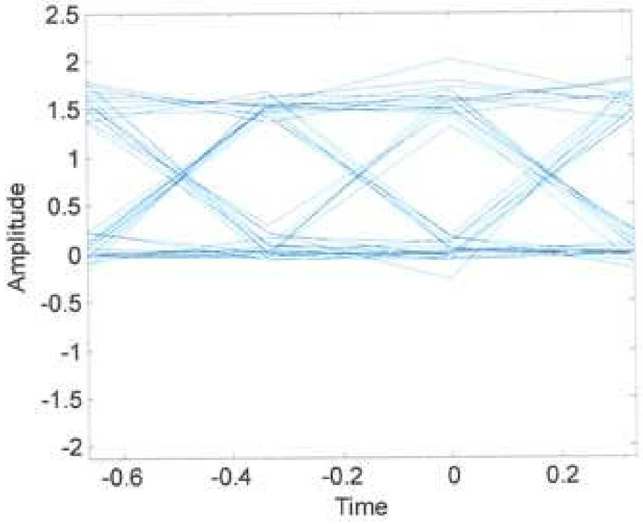


Fig. 6.28: The eye diagram for equalised (MMSE) OOK RZ 62.5 Mb/s

The results obtained form the simulations incorporating the MMSE are shown in Fig.6.29. The figure also incorporates the results from other simulations for comparison purposes. The results indicate that for ISI channels the inclusion of the MMSE significantly improves the traditional receiver performance. Irreducible BERs were not reached over the range of D_T simulated; however, a power penalty that increases with D_T is observable. The comparable Wavelet-AI architecture does not incorporate any equaliser algorithm; instead it relies on the feature extraction of the CWT section and the classification of the neural network as in the non-ISI case. The model is therefore the same as Fig. 6.16 and the results are repeated in Fig. 6.29 for comparison. It can be seen that for most cases values of D_T the Wavelet-AI receiver shows compatible to marginally better performance over the traditional-MMSE case. For D_T of 0.2 the equalised and Wavelet-AI systems provide a power advantage over the unequalised ISI case of a little less than 3 dB. At a D_T of 0.25 the power penalty incurred by the unequalised ISI case has risen to just over 3 dB; however, the curves are diverging and further simulations for the unequalised ISI case reveal irreducible BERs for D_T figures a little in excess of 0.25. At high levels of D_T there is an approximately 2 dB penalty for the equalised ISI case over the non ISI case.

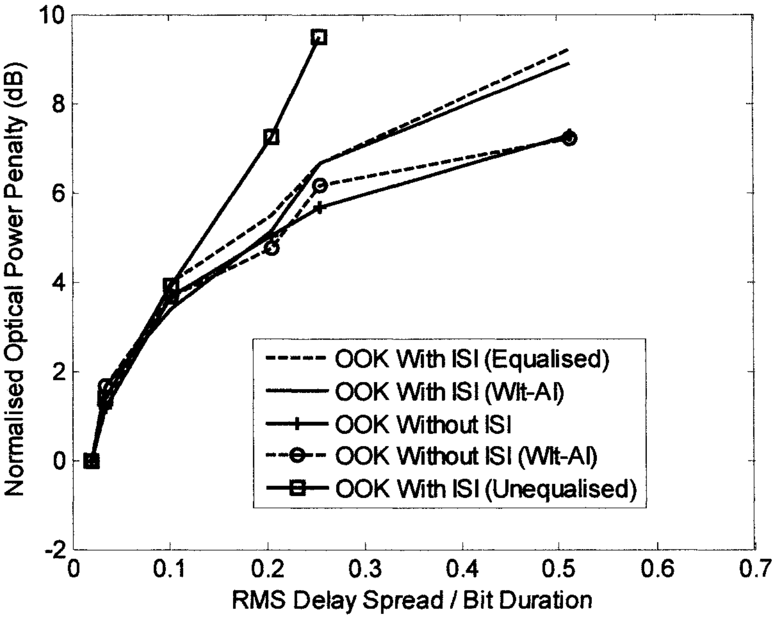


Fig. 6.29: Equalised and unequalised performance of OOK RZ with and without ISI for a BER of 10^{-6}

6.5.2 PPM Systems (Traditional and Wavelet-AI)

For equalised traditional PPM systems we consider 2 common architectures. The first is primarily as described in section 3.2.1 with the addition of a MMSE before the threshold detector as shown in Fig. 6.30. The flowchart for this scheme is shown in Fig. 6.31 and the program listing on the accompanying CD. The second model, as shown in Fig. 6.32, has an MMSE but also includes a soft decision decoder in place of the threshold detector as described in chapter 5, section 3.3. In the threshold case more than one slot per symbol can be above the threshold forcing the receiver to arbitrate in some manner. The soft decision decoder compares all slots within a symbol and chooses the largest; therefore, arbitration never takes place. Both schemes were used to simulate 4-PPM and 8-PPM modulation. The WBSD receiver does not change and the architecture is as described in section 4.2.2 and the results repeated here for comparison with the results of the traditional equalised receiver models.

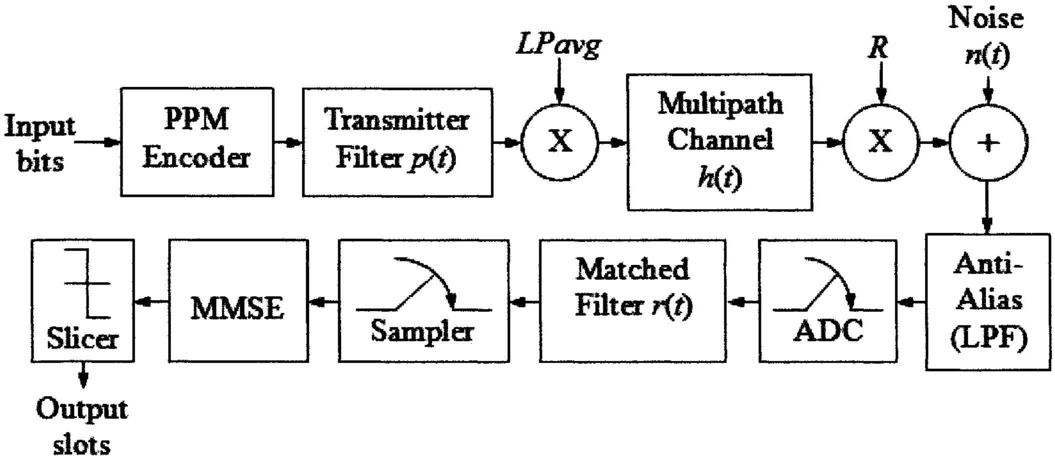


Fig. 6.30: Equalised 4-PPM TH receiver schematic

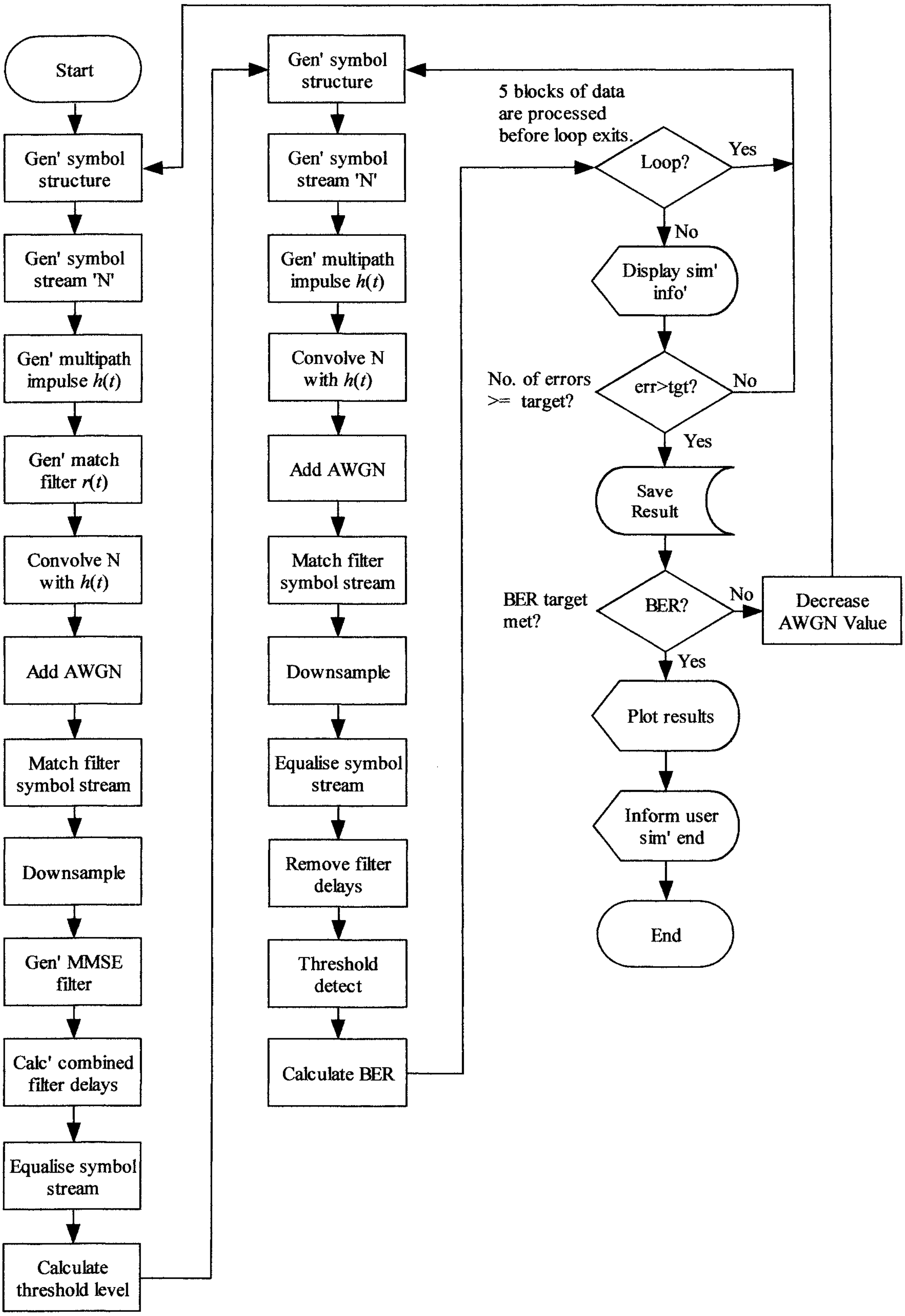


Fig. 6.31: Flowchart for simulation model shown in Fig. 6.30

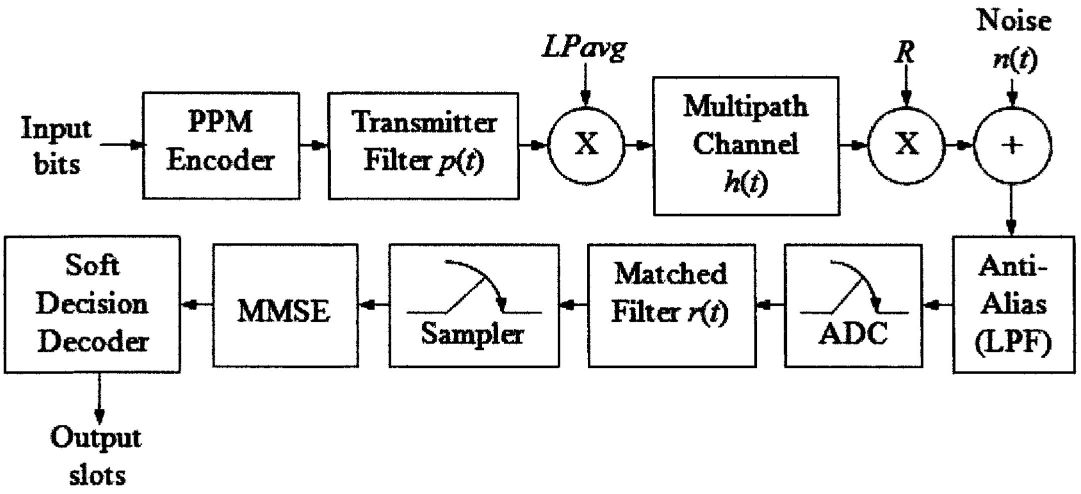


Fig. 6.32: Equalised PPM (MAP) receiver schematic

The results of the 4-PPM (TH), 4-PPM (MAP) and WBSD simulations are shown in Fig. 6.33. In all cases the performance is significantly improved over the non-equalised case shown in Fig. 6.20, where power penalties even with low values of D_T were significant. At a D_T of 0.5 the 4-PPM (TH) equalised case has an approximately 2 dB power penalty over the comparable non ISI performance curve; however, the Wavelet-AI receiver shows less than a 1 dB power penalty at this figure. The 4-PPM (MAP) equalised performance curve shows less than a 1 dB improvement over the 4-PPM (TH) case, with the Wavelet-AI case being comparable for a range of D_T values.

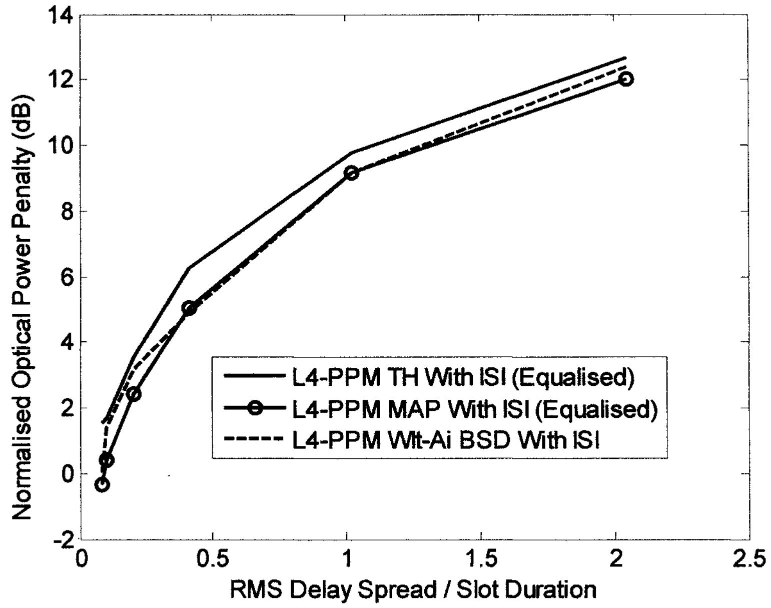


Fig.6.33: Equalised performance of 4-PPM (TH & MAP) and Wavelet-AI for a BER of 10^{-6}

6.5.3 8-PPM

As expected, the results of the equalised 8-PPM systems depicted in Fig. 6.34 show a dramatic improvement over the unequalised system performance shown in Fig. 6.20. The equalised 8-PPM performance of all the schemes show a little more variation than for the 4-PPM case. MAP detection provides an increase in performance over the threshold case of some 1.5 dB at a D_T of approximately 0.5, narrowing to less than 1 dB for the highest values of D_T simulated. The WBSD case whilst erratic shows the same trend as the other schemes and shows a power penalty of less than 1 dB over the traditional 8-PPM (MAP) case for higher values of D_T.

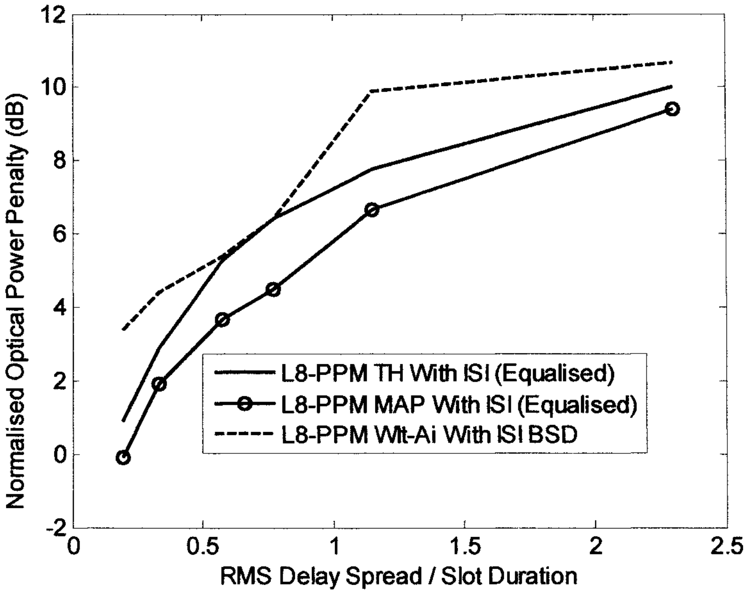


Fig.6.34: Equalised performance of 8-PPM (TH & MAP) and Wavelet-AI for a BER of 10^{-6}

6.6 Summary

The results presented show that the Wavelet-AI receiver architecture outperforms traditional matched filter receiver based receivers for OOK RZ, 4-PPM and 8-PPM even with the filter matched to the ISI affected pulse shape. The Wavelet-AI architecture is also capable of providing BER performance broadly comparable with the equalised forms of the traditional receiver structures; the significant caveat to this is the performance of the model used for 8-PPM that displayed wider variations in performance. However, there are a number of factors that begin to affect the basic operation of the Wavelet-AI structure for higher orders of PPM. Whilst sample rate and the number of effective samples per slot are likely to have some effect on both Wavelet-AI and traditional structures, the Wavelet-AI architecture is likely to need correspondingly longer lengths of training data for higher orders of PPM. This is due to the permutations of detected pulse positions and preceding pulse positions, as these previous pulse positions affect the detected position. It is therefore postulated that increasing the number of training samples should yield further improvements to performance for higher orders of PPM. However, in these simulations between 1000 and 1600 training symbols were used. Increasing the number of training symbols increases the computational burden both in terms of processor load and memory requirement; hence simulation times increase correspondingly. Unfortunately, increasing the number of training symbols would lead to unfeasible simulation times on current equipment and therefore such a strategy was not pursued.

Chapter 7

Conclusions

The overarching objective of research into indoor optical communications is to successfully harness the potentially large bandwidths available for high speed data transfer. The aspiration of this work is to support this fundamental objective by investigating novel methods that may yield improvements over traditional techniques. The primary objective of this work was to establish whether a Wavelet and Artificial Intelligence based receiver architecture had the potential to work successfully when employed in the detection of signals in the diffuse indoor IR environment. The motivation behind this is the limitations of existing systems to adequately overcome the demanding limitations imposed by the diffuse indoor IR channel, including multipath propagation and artificial light interference. Research has been particularly active with respect to finding modulation schemes that provide some performance advantage in this environment. However, as expected gains in one area generally lead to losses in another. It is hoped that the unique detection abilities of the receiver architecture presented in this work will allow a number of modulation schemes to overcome some of the limitations seen when traditional receiver structures are used. Unfortunately, this is the subject of further work and the focus in this thesis is limited to OOK and PPM.

To be able to progress the primary objective of this work it was necessary to investigate the underlying principles of the three main topic areas involved. Firstly, the diffuse indoor IR environment and its limitations were discussed in chapter 2. Secondly, signal feature identification, particularly with respect to the relatively recent technique of wavelet analysis was considered in chapter 3. And finally, artificial intelligence and its implementation in some relevant detection scheme were examined in chapter 4. This broad scope, essentially requiring knowledge of three specialist areas of technology had to be tackled before considering the implications of the target environment as detailed in chapter 5 and chapter 6. In addition to this benchmark simulation models were required so that the performance of any new receiver architectures could be compared on a like for like basis. This approach was deemed the most appropriate as mathematical analysis of the proposed architectures may not be possible, or would require research in its own right providing minimum contribution to the primary objective.

The breadth of this work has necessarily limited the time available to refine and test the architectures presented; however, the work successfully shows that Wavelet-AI architectures employed in the diffuse indoor IR environment perform broadly similar or in some cases better than the benchmark traditional systems. Further development work is required to more comprehensively determine the potential of such architectures, and suggestions for further work will be included later in this chapter.

Chapter 3 clearly indicates that wavelet analysis has the ability to temporally isolate signal features. A number of related techniques were presented in the chapter, however, the work here being fundamental focussed on the continuous wavelet transform and its multi-resolution ability. For comparison, Fourier analysis was briefly reviewed and compared to wavelet analysis underlining the differences between the two methods. The feature mapping ability of the continuous wavelet transform make it a good candidate to marry with the pattern classification abilities of artificial intelligence. The temporal relationship between an information signal and wavelet coefficients was demonstrated by the implementation of a receiver system based largely on a single scale coefficient produced by the continuous wavelet transform.

In chapter 4 artificial intelligence principles were introduced, along with practical implementations; in particular feed-forward back-propagation networks and training algorithms were discussed. Network architectures including the data 'windowing' scheme used extensively in the remainder of the work was detailed and the results of an initial direct detection by neural networks on OOK modulation simulation presented. The initial results indicated approximately a 4 dB variation in SNR for a particular BER for a 2 dB variation in SNR used to train the network, with individual simulation runs producing significant variations for a single SNR training figure. The chapter went on to consider implementing a low pass filter scheme prior to the network to in an attempt to reduce such variations. Results from the subsequent simulations indicated that a pre-filter strategy may yield improvements. In an attempt to improve computational performance the pre-filter and neural network architecture was amended to include down-sampling, where a single point per bit was passed to the neural network section. Simulation results of this scheme indicated a return to large variations in performance and the strategy was not pursued further.

The chapter continued by introducing the wavelet transform in place of the low-pass filter section and discussed the packaging and presentation structure for the subsequently produced wavelet coefficients. The results of initial simulations were presented and showed tight grouping of results within approximately 1 dB for a particular BER. However, computational burden was significant and simulation times prohibitive which lead to the rest of the chapter considering simplification strategies so that the receiver architecture could be simulated in a more realistic time frame. Reducing the data window size, the number of wavelet coefficient scales and reducing the number of neurons in the neural network were all considered and simulated. The results showed that significant simplifications could be made whilst maintaining similar performance for the particular conditions used. Simulation times were reduced but remained significant, with a single simulation taking several days in many cases.

The work went on to include simulations considering the effect of ambient light sources, particularly fluorescent lights driven by electronic ballasts. Chapter 5 presented the fluorescent light model to be used in the simulations and went on to consider the effect of interference produced by this model on traditional and Wavelet-AI receivers for both OOK RZ and PPM schemes. As expected the traditional OOK RZ receiver suffered severe performance degradation, with results indicating an almost constant 18 dB optical power penalty compared to the non-interference case. In contrast the Wavelet-AI suffered less than an approximately 1 dB optical power penalty over the data rate simulated. The traditional PPM receiver also suffered severe performance degradation; however, the MAP implementation of PPM showed more resilience, but at 2.5 Mb/s the difference between the interference and non-interference case for the 'best' result was still in excess of 10 dB. The chapter progressed to the introduction of a

Wavelet-AI binary symbol detector, a Wavelet-AI soft slot detector and their simulations under the same channel conditions. The key point form the simulation results being that the interference and non-interference curves were not divergent as in the traditional case, but showed a trend to having some offset between the two cases. In summary, the Wavelet-AI models outperformed the traditional receiver structures in all cases with interference and were comparable or slightly better without interference. The largest variation resulted from the use of the WSSD, which produced an approximately 3.5 dB power penalty between the interference and non-interference case at low data rates.

The chapter progressed to briefly consider the effects of baseline wander on OOK detection for both traditional and Wavelet-AI architectures. As detailed, such a situation is purely hypothetical in the Wavelet-AI case as deployment with filtering is not envisaged for practical applications. However, the inclusion of a HPF in the Wavelet-AI system allows direct comparison to traditional architectures employing HPF for mitigating artificial light interference. The simulation results using OOK RZ show that the Wavelet-AI architecture is significantly more tolerant to baseline wander than its traditional counterparts being able to successfully tolerate filter cut on frequencies at the fundamental data frequency. For a 'cut on frequency / data rate' of 10^{-1} the Wavelet-AI system offered over a 4 dB optical power improvement on the best performing traditional receiver incorporating a HPF.

The work of chapter 5 concluded by considering the effect of fluorescent light interference with electrical high pass filtering. This was contained to the traditional receiver systems and the results compared with those previously obtained from simulations with Wavelet-AI receivers. For OOK RZ the inclusion of a HPF significantly reduced the effects of fluorescent light interference; however, at data rates below approximately 10 Mb/s the improvement starts to diminish with optical power penalties gradually increasing. The Wavelet-AI system does not suffer the same effects and its optical power penalty continued to decrease in line with the data rate; offering over a 2 dB improvement over the best HPF case. The results of PPM stimulations with inclusive HPF's showed similar trends with traditional systems showing sudden increases in optical power penalty at data rates below about 5 Mb/s, whilst the performance of the Wavelet-AI systems continued to improve in line with the reduced data rate. The Wavelet-AI soft slot detector provided results inferior to all other cases except at the lowest data rates simulated, however, the Wavelet-AI binary symbol detector gave comparable performance to the PPM(TH) case at higher data rates and approximately 1.5 dB worse then the PPM(MAP) case.

Finally, chapter 6 looked at the effects of multipath propagation and showed that data rates are severely limited unless effective mitigation is employed. Popular compensation techniques were introduced, concentrating on a detailed description of the subsequently deployed minimum mean square equaliser. Following this, a multipath channel model incorporated in the simulations was introduced.

The unequalised performance of OOK and PPM schemes for both OOK and PPM was discussed. As expected, traditional receiver architectures experienced significant performance degradation in the presence of multipath induced ISI. The traditional OOK RZ scheme suffered an optical power penalty of 4 dB for a D_T of 0.25, with rapidly degrading performance for the ISI case. In contrast the Wavelet-AI architecture showed less than 1 dB variation between the ideal and ISI case at a D_T of 0.3 and less than 2 dB at a D_T of 0.5.

For traditional PPM receivers the results were more dramatic with 4-PPM showing a power penalty of almost 7 dB at a D_T of approximately 0.15 and approximately 4 dB in the 8-PPM case. In both instances the curves of the ISI and ideal cases were aggressively divergent for all values of D_T simulated. Again in stark contrast, the Wavelet-AI receiver did not show uncontrolled divergence between the ideal and ISI case, but a trend towards a constant or mildly increasing power penalty. The ISI case show little discernable difference from the ideal case for a D_T up to approximately 0.5 for both 4-PPM and 8-PPM.

For OOK and PPM traditional systems their performance in the presence of ISI improved markedly with the inclusion of equalisers. In broad terms the performance of the Wavelet-AI systems was very similar to these cases with the exception of 8-PPM where the performance was more erratic and showed a power penalty of approximately 1 dB at higher values of D_T.

In conclusion, the results of simulations show that the Wavelet-AI architectures described in this work offer significantly improved performance over uncompensated traditional receivers for both fluorescent light interference and multipath induced ISI. For conventionally compensated systems the Wavelet-AI architecture showed an improvement in performance for lower data rates with a signal subjected to fluorescent light interference. For higher data rates and equalised receivers minor variations exist as detailed in the various chapters. However, broadly similar performance was evident.

Whilst specific compensating methods are required for standard receiver systems to overcome particular interferers or channel imperfections, the same is not the case for

Wavelet-AI. The simulation evidence indicates that provided the correct wavelet scales are chosen and training given, the Wavelet-AI receiver will compensate for multiple deficiencies in the received signal providing the flexibility for a 'standard' scheme to be deployed in numerous environments. Due to the nature of the simulations and limitations in performance of the computer hardware used in this work further refinement of the results is possible. However, the broader comparisons for the conditions indicated in this work will remain the same.

The IM/DD, base-band schemes employed in the indoor diffuse environment mean that direct digitisation of the detected signal is possible with currently available hardware. Given this, software intensive schemes such as Wavelet-AI will be easier to deploy, adapt and upgrade than their traditional hardware only predecessors. Further enhancements may see schemes like Wavelet-AI 'choosing' the best modulation scheme for a given environment, tuning the system for optimum performance.

A particular optimised implementation of the Wavelet-AI scheme could be embedded on a custom chip, or perhaps partially deployed on an FPGA. Unfortunately, such a hardware biased implementation may not allow the true flexibility of the Wavelet-AI system to be realised. If the scheme was deployed on fast adaptable processors it would potentially be able to dynamically configure a wide range of parameters 'on the fly', adapting to the changing environment (day-time, night-time, lights on, lights off etc) and choosing the best wavelet, scales, network or modulation scheme to optimise performance in near real time.

Chapter 8

Further Work

Whilst the objectives listed in chapter 1 have been achieved, there is still potential for much more research to be done. With such large topic areas and relatively new techniques it was never envisaged that this work could be self contained and comprehensive; the amount of work and time required being beyond the scope of the thesis. However, it was envisaged that this work, should its results be encouraging, be the starting point for other aspects in development this technique to further enhance and refine it.

As a straight forward progression of the work presented here simulations into the effects of combined fluorescent light interference and multipath propagation could prove useful. Whilst the individual effects are potentially confined to high or low data rates the combined effect does mirror a practical deployment. The results would hopefully further confirm the ability of the Wavelet-AI structure to reject or limit the effect of complex signal distortions.

One of the major problems encountered in this work was the availability of computational effort for the simulations. The PC hardware of the day, combined with the inherent way Matlab™ functions slowed the pace of work at stages to a tedious pace. Given that the work here shows the viability of the Wavelet-AI architecture any further researchers should consider spending the time and effort in converting the appropriate Matlab™ code in to a host processor high level native language such as 'C'. It is envisaged that whilst the effort to do this may be significant it would yield the performance enhancements required for further work to be continued in a timely manner. Further enhancements could be made by the division of computational effort amongst several computer platforms.

The choice of wavelet coefficient scales to use during a simulation was carried out empirically from time-scale-intensity plots. This is clearly not optimum and the performance of the architectures presented could conceivably be improved by an algorithm to select the most suitable contributions. For deployed implementations such an algorithm would obviously be essential. Further to this, bespoke wavelet designs, tailored to the channel conditions by algorithms 'on-the-fly' could also yield further improvements; whilst using an alternative wavelet algorithm may diminish computational burden.

Neural network, type, size, training algorithms and activation functions provide an almost inexhaustible list of permutations to overcome when implementing a system such as Wavelet-AI. However, as indicated, neural networks can also show very little performance variations for what seem to be large changes in network architecture. One straightforward piece of work would be to replace the linear activation function with the 'Hard limit function' in the output layers. If successful this would obviate the need for a slicer in the Wavelet-AI architecture. More fundamental investigations in to the use of probablistic networks and recurrent networks could prove fruitful.

Whilst OOK and 4-PPM proved reasonably consistent through simulation, 8-PPM showed more variation in results. It is postulated that this is largely due to the increased permutations being classified by the network. Research could be directed in this area to identify and overcome the possible effects associated with higher order PPM and the Wavelet-AI architecture.

Many modulation schemes have been proposed for use in the indoor infrared environment. This work, by the very nature of having to restrict scope, concentrated on the well known and used OOK and PPM schemes. However, other techniques show particular advantages under certain conditions imposed by the indoor environment, but disadvantages under other conditions. The flexibility and inherent compensating nature of a detection scheme like Wavelet-AI could potentially lead to a number of candidate modulation schemes outperforming their implementation on traditional receiver hardware.

Given that hardware for direct digitisation at speeds required for reasonable data rates is commercially available, a concrete implantation of the Wavelet-AI system, and benchmark software defined traditional system is achievable. With code written in 'C' this would be a major step toward fully understanding system implications and practical deployment.

Appendix A

Matlab™ Listing for: Fig. 5.7.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% wlt_nn_fl_OOK_RZ %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This function trains neural networks to detect an OOK signal with
% fluorescent light interference under different noise conditions and
% then tests them against unknown data using wavelet coefficients as the
% information source.
%
% This file contains a number of embedded functions
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function wlt_nn_fl_OOK_RZ

% Turns the interference on or off.
flInterference = 1; % Fluorescent light interference (1 = on).
matchFilter = 0; % Match filter detection (1 = on). 'NOT USED'

% The number of samples defines the data rate, must be divisible by 2
samplesPerSymbol = 400;
temp = 1; save results_2p5Mbps_wlt_nn_fl_OOK_RZ temp; % Open a results file

%What an OOK '0' and '1' look like.
zero=zeros(1,samplesPerSymbol);
one=[zeros(1,samplesPerSymbol/2) ones(1,samplesPerSymbol/2)];

SNR = -18;  %The starting SNR -3
index = 1;  %Used to keep track of which BER and SNR we are recording
runBER = 1; %Running total of the BER when we are in the While loop
totErr = 0; %Keeps track of the total errors for a particular SNR
training=0;%We are not training here
wordBlocks = 2;%The number of times the program goes round the while loop.
numSymbols=1000; %The number of symbols each time round the while loop
numDecimations=numSymbols; %The number of 'answers' we expect.

%While the BER is less than 1e-5 and the SNR is less than 60dB, keep going.
while(runBER>1e-5 && SNR<60)

    if(runBER>5e-3)
        increment = 3;% Slow increment down as BER gets lower (curve steep)
    else
        increment = 0.5;
    end

    SNR = SNR + increment;% SNR value

    %Train the neural network
    [net, minp, maxp]=trainNetwork(samplesPerSymbol, zero, one, ...
    flInterference, SNR, matchFilter);
    totErr = 0; %Reset
    totBits = 0; %Reset

    %While the total errors are less than 5 keep going
    while(totErr<5)
        for ctr=1:wordBlocks

            % Generate the OOK signal
            [OOK, data]=generateOOK(numSymbols, samplesPerSymbol, SNR,...
                zero, one, training, flInterference, matchFilter);

            % Condition it: Use CWT and package it properly for NN
            [conditionedSig]=sigConditioner(OOK, samplesPerSymbol, ...
                numDecimations);

            % Pass the NN the conditioned signal and detect it.
            Net_res=[];
            [res]=AI_MP_WLT_OOK_SIM(net, conditionedSig, minp, maxp);
            Net_res=[Net_res res];

            %Gets rid of the leading and trailing zeros, which we haven't
            %really detected (due to windowing, ie cwt of 3 bits).
            data=data(2:(end-1));

            totErr = totErr + sum(abs(Net_res-data));%Calcualte tot errors

            totBits = totBits + numSymbols;%Calculate tot number of bits

            runBER = totErr/totBits;%Calculate the running BER
        end

        %Display information regarding SNR and BER
        info=strcat('Samples per symbol_',num2str(samplesPerSymbol),...
        '_noise level_', num2str(SNR),'_Total BITS_', num2str(totBits),...
        '_Total errors_', num2str(totErr), '_BER_', num2str(runBER));
        disp(info);

    end

    %Record the results
    BER(index) = runBER;
    snr(index) = SNR;
    index = index + 1;

    % Save the results
    saveResults(BER, snr);
end

% Inform the user the simulation has finished
msgbox('Simulation finished','Information','help');

% At the end of the simulation plot SNR v BER.
semilogy(snr,BER);
xlabel('SNR (dB)');
ylabel('BER');
title('SNR against BER');
return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%EMBEDDED FUNCTIONS%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% trainNetwork %%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function trains the neural network.
%
%INPUTS:-
%samplesPerSymbol: The number of 1 nanosecond samples that make up the
%symbol. Eg 100Mbps OOK requires 10 x 1 nanosecond samples.
%
%zero: What a zero pulse looks like
%
%one: What a one pulse looks like
%
%flinterference: Whether interference is 'on or 'off'
%
%SNR: The value of the added noise
```

```
%
%matchFilter: Whether a match filter is used (not used)
%
%OUTPUTS:-
%netInfo: Information string on the network
%
%net: The trained neural network
%
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [net, minp, maxp]=trainNetwork(samplesPerSymbol, zero, one, ...
    flInterference, SNR, matchFilter)

numSymbols = 1500;% The amount of symbols used to train the network.

% Make the data to train with, we need 3 streams for training.
[trainingInput1, data1]=condTrainingSets(numSymbols, samplesPerSymbol, ...
    zero, one, flInterference, SNR, matchFilter);

[trainingInput2, data2]=condTrainingSets(numSymbols, samplesPerSymbol, ...
    zero, one, flInterference, SNR, matchFilter);

[trainingInput3, data3]=condTrainingSets(numSymbols, samplesPerSymbol, ...
    zero, one, flInterference, SNR, matchFilter);

% Get rid of leading & trailing zeros we added then train.
[net,minp,maxp]=AI_MP_WLT_OOK_Train(trainingInput1, ...
    data1(2:(length(data1)-1)),trainingInput2,data2(2:(length(data2)-1))...
    ,trainingInput3,data3(2:(length(data3)-1)));

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%% condTrainingSets %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function makes training data for the neural network.
%
%INPUTS:-
%numSymbols: The number of OOK symbols required
%
%samplesPerSymbol: The number of 1 nanosecond samples that make up the
%symbol. Eg 100Mbps OOK requires 10 x 1 nanosecond samples.
%
%zero: What a zero pulse looks like
%
%one: What a one pulse looks like
%
%flinterference: Whether interference is 'on or 'off'
%
%SNR: The value of the added noise
%
%matchFilter: Whether a match filter is used (not used)
%
%OUTPUTS:-
%trainingInput: Data suitably formatted for input to a neural network
%
%data: The binary representation of the trainingInput.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

function [trainingInput, data]=condTrainingSets(numSymbols, ...
    samplesPerSymbol, zero, one, flInterference, SNR, matchFilter)

training = 1; % need to tell 'generateOOK that we are training'

% Generate the training data
[OOK, data]=generateOOK(numSymbols, samplesPerSymbol, SNR,...
    zero, one, training, flInterference, matchFilter);

numDecimations = numSymbols;

[trainingInput]=sigConditioner(OOK, samplesPerSymbol, numDecimations);

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% generateOOK %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function produces the data stream.
%
%INPUTS:-
%numSymbols: The number of OOK symbols required
%
%samplesPerSymbol: The number of 1 nanosecond samples that make up the
%symbol. Eg 100Mbps OOK requires 10 X 1 nanosecond samples.
%
%zero: What a zero pulse looks like
%
%one: What a one pulse looks like
%
%Flinterference: Whether interference is 'on or 'off'
%
%SNR: The value of the added noise
%
%matchFilter: Whether a match filter is used (not used)
%
%training: Whether we are generating data for training or simulation
%
%OUTPUTS:-
%OOK: The prepared symbol stream.
%
%dataRate: The data rate based on the samples per sybol passed in
%
%data: The binary representation of the data stream
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [OOK, data]=generateOOK(numSymbols, samplesPerSymbol, SNR,...
    zero, one, training, FlInterference, matchFilter);

sampInt = 1e-9;
windowSize=3;

dutyCycle = 0.5; % 1 for NRZ 0.5 for RZ 0.5 etc (ONLY 0.5 USED)

if (FlInterference)
    FlI=2e-6; % Fluorescent light current = 2 microamps.
    Ib=FlI;
    ShI=200e-6; % Shot noise current = 200 microamps.
else
    ShI=200e-6; % Shot noise current = 200 microamps.
    FlI = 0;
end

```

```
% Work out the noise power
q=1.602e-19;
systemBandWidth = 500e6; % Based on 1 ns sample time
noisePower = q*(ShI + FlI)*systemBandWidth; % Need to get a power value.

% Generate the basic OOK symbolStream
symbolStream=[];
n=numSymbols+(windowSize-1);
data=randint(1,n);
for symbol=1:n
    if(data(symbol)==1)
        symbolStream=[symbolStream one];
    else
        symbolStream=[symbolStream zero];
    end
end

% If fluorescent light interference is on generate the interference
if (FlInterference)
    % The low cycle is 20mS long so start at any time in that period
    interferenceLength = (numSymbols+(windowSize-1))...
        *samplesPerSymbol*sampInt;

    startTime = abs(rand(1,1)) * 20e-3;
    endTime = startTime + interferenceLength;

    %Make the Fl interference (external function)
    [i_elect]=fl_model(Ib, sampInt, startTime, endTime);
    i_elect=i_elect(1:(end-1));
end

% If training add random noise to the symbol stream.
if(training)
    % Scale the signal power to get correct SNR given the noise power
    % (shot noise only here)
    sigPower=noisePower*10^(SNR/10);
    sigAmplitude=sqrt(sigPower/(dutyCycle/2)); %Duty cycle/2 for OOK
    OOK=awgn((symbolStream*sigAmplitude),SNR,'measured');%Add the noise
end

% If training with interference add the interference to the symbol stream
if(training & FlInterference) % Trg with fl interference
    OOK=OOK + i_elect; % Add interference to random noise

% If training without interference, do nothing
elseif(training & ~FlInterference)
    %OOK = OOK - do nothing

% If simulating with interfernce, scale symbol stream and add interference
elseif(~training & FlInterference)
    sigPower =noisePower*10^(SNR/10);
    sigAmplitude=sqrt(sigPower/(dutyCycle/2)); %Duty cycle/2 for OOK
    sigPlusShotNoise = awgn((symbolStream * sigAmplitude),SNR,'measured');
    OOK=sigPlusShotNoise + i_elect;

% If simulating with interfernce, scale symbol stream
elseif(~training & ~FlInterference)
    %Find out what signal power we want (shot noise only here)
    sigPower=noisePower*10^(SNR/10);
    sigAmplitude=sqrt(sigPower/(dutyCycle/2)); %Duty cycle/2 for OOK
    OOK = awgn((symbolStream * sigAmplitude),SNR,'measured');

```

```
else
    %ERROR, should never get here!
    disp('An error has occurred in function: GenerateOOK');
end

% If we are using a match filter (NOT USED)
if(matchFilter)
    matchFilt=fliplr(one)*(1/sqrt(samplesPerSymbol*1e-9*dutyCycle));
    OOK=conv(matchFilt,OOK);
end


return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%% sigConditioner %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function takes the symbol stream and formats it for
%input to the neural network.
%
%INPUTS:-
%signal: The input symbol stream
%
%samplesPerSymbol: The number of 1 nanosecond samples that make up the
%symbol. Eg 100Mbps OOK requires 10 x 1 nanosecond samples.
%
%numDecimations: This is the number of 'windows' we require and is the
%same as the number of bits in the data stream.
%
%OUTPUTS:-
%conditionedSig: A matrix of conditioned data for the neural network
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [conditionedSig]=sigConditioner(signal, samplesPerSymbol, ...
         numDecimations)

numSamplesPerDecimation = 3; %5 for a 5 bit window

% Keep track of where we are in the symbol stream
start = 1;
index = samplesPerSymbol;
finish=samplesPerSymbol * numSamplesPerDecimation;


% Arrange the signal in a 3 window by numDecimations matrix
for decimation=1:numDecimations

    % Cut out the samples from the signal and perform a CWT
    % Rem, scales are chosen to match the data rate
    coef=cwt(signal(start:finish),[250 750 1000 1250 1500],'sym2');

    % Sort the coefficients in to a single column
    [r,c]=size(coef);
    sig=[];
    for row=1:r
        sig=[sig coef(row,:)];
    end

    %Pre define the holding matrix
    if(decimation==1)
        conditionedSig=zeros(length(sig),numDecimations);
    end

    % Add the column to a matrix where each column represents the
    % coefficients from one 'window'
    conditionedSig(:,decimation)=sig';

    % Index to the next window
    start = start + index;
    finish = finish + index;
end

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% saveResults %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function saves all parameters and results associated with the
%simulation.
%
%INPUT:-
%BER: The BER for a particular SNR
%
%SNR: The corresponding SNR
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function saveResults(BER, SNR);

save -append results_2p5Mbps_wlt_nn_fl_OOK_RZ BER;
save -append results_2p5Mbps_wlt_nn_fl_OOK_RZ SNR;

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AI_MP_WLT_OOK_Train %%%%%%%%%%%%%%%%%%%%%%%%
%
%This function carries out network training training.
%
%INPUTS:-
%Net_input: The values the network is trained on
%
%True_values: The true binary representation of the Net_input
%
%Val_impt: Validation data
%
%Val_R: The true binary representation of the Validation data
%
%Tst_impt: Test data
%
%Tst_R: The true binary representation of the Test data
%
%OUTPUTS:-
%Net: The trained neural network
%
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Net,minp,maxp]=AI_MP_WLT_OOK_Train(Net_input,True_values,...
          Val_impt,Val_R,Tst_impt,Tst_R);


%Normalise the input and targets. Matlab function returning the
%normalisation matrices and the normalised data 'pn'.
%Training set
[pn,minp,maxp] = premnmx(Net_input);
```

```
tn=True_values;

%Validation set. Matlab function normalising the validation set.
val.P = tramnmx(Val_impt,minp,maxp);
val.T = Val_R:

%Test set. Matlab function normalising the test set.
tst.P = tramnmx(Tst_impt,minp,maxp);
tst.T = Tst_R;

%Create the network. Matlab function creating a feedforward network.
%5 first layer, 1 output layer: 'purelin', and 'logsig' are activation
%functions and 'trainscg' is the training algorithm.
net=newff(minmax(pn),[5,1],{'logsig','purelin'},'trainscg');

net=init(net);%Matlab function initialising the network weights

%Adjust the training parameters
net.trainParam.show = 10;
net.trainParam.lr = 0.05;
net.trainParam.epochs = 250;
net.trainParam.goal = 1e-14;
net.trainParam.min_grad = 1e-9;

% Matlab function to train the network
[Net,tr]=train(net, pn, tn,[],[],val,tst);

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AI_MP_WLT_OOK_SIM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function uses the nural network to determine the binary value of the
%centre bit from a 5 bit window. Please read the INPUTS & OUTPUTS
%
%Inputs:
%Net - The neural network to use.
%
%Net_input - The data stream to be processed.
%
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%Outputs:
%Net_res - The thresholded network output.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Net_res]=AI_MP_WLT_OOK_SIM(Net, Net_input, minp, maxp);

%%Normalise the input in accordance with the trained network. Matlab
%function
pn = tramnmx(Net_input,minp,maxp);

%Simuate the network. Matlab function
R=sim(Net, pn);

%Step through the simulation results and threshold
[r,c]=size(R);

for i=1:c
    for j=1:r
        temp = R(j,i);
        if(temp>0.5)
            R(j,i)=1;
        else
            R(j,i)=0;
        end
    end
end

Net_res=R;

return;

```

Appendix B

Matlab™ listing for: Fig. 5.18

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Wlt_AI_BSD %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This function simulates a Wavelet AI Binary Symbol Detector
% for L4-PPM with and without fluorescent light interference and
% different noise conditions
%
% This file contains a number of embedded functions and uses external
% functions for fluorescent light interference
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Wlt_AI_BSD

flInterference = 1; % Fluorescent light interference (1 = on).
matchFilter = 0;    % Match filter detection (1 = on) NOT USED.

samplesPerSymbol = 80;%Number of 10 ns pulses per symbol.
temp = 1; save results; % Open a results file

%What an PPM '0' and '1' look like.
zero=zeros(1,samplesPerSymbol/4);
one=[ones(1,samplesPerSymbol/4)];

SNR = -12;  %The starting SNR
index = 1;  %Used to keep track of which BER and SNR we are recording
runBER = 1; %Running total of the BER when we are in the While loop
totErr = 0; %Keeps track of the total errors for a particular SNR
training = 0; %We are not training here
wordBlocks = 2;%The number of times the program goes round the while loop.
numSymbols = 500; %The number of symbols each time round the while loop
numDecimations = numSymbols; %The number of 'answers' we expect.

%While the BER is less than 5e-6 and the SNR.
while(runBER>5e-6)

    % Slow increment down as BER gets lower (curve steep)
    if(runBER>1e-3)
        increment = 3;
    else
        increment = 0.5;
    end

    SNR = SNR + increment;% SNR value

    % Train the neural network
    [net, minp, maxp]=trainNetwork(samplesPerSymbol, zero, one, ...
    flInterference, SNR, matchFilter);
    totErr = 0; %Reset
    totBits = 0; %Reset

    % While the total errors are less than 5 keep going
    while(totErr<5)
        for ctr=1:wordBlocks

            % Generate the OOK signal
            [OOK, data, symbols]=generatePPM(numSymbols,...
                samplesPerSymbol, SNR, zero, one, training, ...
                flInterference, matchFilter);

            % Condition it: Use CWT and package it properly for NN
            [conditionedSig]=sigConditioner(OOK, samplesPerSymbol, ...
                numDecimations);

            % Pass the NN the conditioned signal and detect it.
            Net_res=[];
            [res]=AI_WLT_PPM_SIM(net, conditionedSig, minp, maxp);
            Net_res=[Net_res res];

            %Gets rid of the leading and trailing zeros, which we haven't
            %really detected (due to windowing, ie cwt of 3 bits).
            data=data(:,2:(end-1));

            %Calcualte tot errors
            totErr = totErr + sum(sum(abs(Net_res-data)));

            %Calculate tot number of bits 2 bits for L4 PPM
            totBits = totBits + numSymbols * 2;

            runBER = totErr/totBits; %Calculate the running BER
        end
        %Display information regarding SNR and BER
        info=strcat('Samples per symbol_',num2str(samplesPerSymbol),...
        '_noise level_', num2str(SNR),'_Total BITS_', num2str(totBits),...
        '_Total errors_', num2str(totErr), '_BER_', num2str(runBER));
        disp(info);
    end

    %Record the results
    BER(index) = runBER;
    snr(index) = SNR;
    index = index + 1;

    %Save the results
    saveResults(BER, snr);
end

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%EMBEDED FUNCTIONS%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% trainNetwork %%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function trains the neural network.
%
%INPUTS:-
%samplesPerSymbol: The number of 10 nanosecond samples that make up the
%symbol.
%
%zero: What a zero pulse looks like
%
%one: What a one pulse looks like
%
%flinterference: Whether interference is 'on or 'off'
%
%SNR: The value of the added noise
%
%matchFilter: Whether a match filter is used (not used)
%
%OUTPUTS:-
%netInfo: Information string on the network
%
%net: The trained neural network
%
```

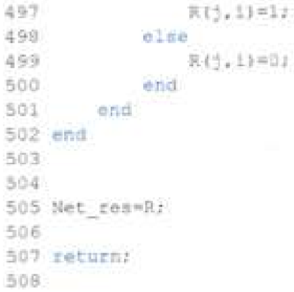
```
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [net, minp, maxp]=trainNetwork(samplesPerSymbol, zero, one, ...
    flInterference, SNR, matchFilter)

numSymbols = 1000;% The amount of symbols used to train the network.

%Make the data to train with, we need 3 streams for training.
[trainingInput1, data1]=condTrainingSets(numSymbols, samplesPerSymbol, ...
    zero, one, flInterference, SNR, matchFilter);

[trainingInput2, data2]=condTrainingSets(numSymbols, samplesPerSymbol, ...
    zero, one, flInterference, SNR, matchFilter);

[trainingInput3, data3]=condTrainingSets(numSymbols, samplesPerSymbol, ...
    zero, one, flInterference, SNR, matchFilter);

% Get rid of leading & trailing zeros we added then train.
[net,minp,maxp]=AI_MP_WLT_OOK_Train(trainingInput1, ...
    data1(:,2:(length(data1)-1)),trainingInput2,data2...
    (:,2:(length(data2)-1)),trainingInput3,data3(:,2:(length(data3)-1)));

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%% condTrainingSets %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function makes training data for the neural network.
%
%INPUTS:-
%numSymbols: The number of PPM symbols required
%
%samplesPerSymbol: The number of 10 nanosecond samples that make up the
%symbol.
%
%zero: What a zero pulse looks like
%
%one: What a one pulse looks like
%
%flInterference: Whether interference is 'on or 'off'
%
%SNR: The value of the added noise
%
%matchFilter: Whether a match filter is used (not used)
%
%OUTPUTS:-
%trainingInput: Data suitably formatted for input to a neural network
%
%data: The binary representation of the trainingInput.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [trainingInput, data]=condTrainingSets(numSymbols, ...
    samplesPerSymbol, zero, one, flInterference, SNR, matchFilter)

training=1;%need to tell 'generatePPM that we are training'
[OOK, data, symbols]=generatePPM(numSymbols, samplesPerSymbol, SNR,...
    zero, one, training, flInterference, matchFilter);

```

```
numDecimations = numSymbols;

[trainingInput]=sigConditioner(OOK, samplesPerSymbol, numDecimations);

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% generatePPM %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function produces the data stream.
%
%INPUTS:-
%numSymbols: The number of OOK symbols required
%
%samplesPerSymbol: The number of 10 nanosecond samples that make up the
%symbol.
%
%zero: What a zero pulse looks like
%
%one: What a one pulse looks like
%
%Flinterference: Whether interference is 'on or 'off'
%
%SNR: The value of the added noise
%
%matchFilter: Whether a match filter is used (not used)
%
%training: Whether we are generating data for training or simulation
%
%OUTPUTS:-
%PPM: The prepared symbol stream.
%
%symbols: a number between 1 and 4 representing the pulse position
%
%data: The binary representation of the data stream
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [PPM, data, symbols]=generatePPM(numSymbols, samplesPerSymbol,...
    SNR, zero, one, training, FlInterference, matchFilter);

sampInt = 1e-8;
windowSize=3;

L = 4; % 4 for 4PPM

ZERO_ZERO = [one zero zero zero];
ZERO_ONE = [zero one zero zero];
ONE_ZERO = [zero zero one zero];
ONE_ONE = [zero zero zero one];

if (FlInterference)
    FlI=2e-6; % Fluorescent light current = 2 microamps.
    Ib=FlI;
    ShI=200e-6; % Shot noise current = 200 microamps.
else
    ShI=200e-6; % Shot noise current = 200 microamps.
    FlI = 0;
end

q=1.602e-19;
systemBandWidth = 50e6; %% Based on 10 ns sample time
noisePower = q*(ShI + FlI)*systemBandWidth; % Need to get a power value.

% Generate the basic PPM symbolStream
```

```
symbolStream=[];
data=[];
n=numSymbols+(windowSize-1);
symbols=randint(1,n,[1 4]);
for symbol=1:n
    switch symbols(symbol)
        case 1
            symbolStream=[symbolStream ZERO_ZERO];
            data=[data [0 0]'];
        case 2
            symbolStream=[symbolStream ZERO_ONE];
            data=[data [0 1]'];
        case 3
            symbolStream=[symbolStream ONE_ZERO];;
            data=[data [1 0]'];
        case 4
            symbolStream=[symbolStream ONE_ONE];
            data=[data [1 1]'];
        otherwise
            disp('ERROR');
    end
end


% If fluorescent light interference is on
if (FlInterference)
    % The low cycle is 20mS long so start at any time in that period
    interferenceLength = (numSymbols+(windowSize-1))...
        *samplesPerSymbol*sampInt;

    startTime = abs(rand(1,1)) * 20e-3;
    endTime = startTime + interferenceLength;

    %Make the Fl interference
    [i_elect]=fl_model(Ib, sampInt, startTime, endTime);
    i_elect=i_elect(1:(end-1));
end

if(training) % If training add random noise.
    % Find out what signal power we want
    sigPower=noisePower*10^(SNR/10);
    sigAmplitude=sqrt(sigPower*L);
    PPM=awgn((symbolStream*sigAmplitude),SNR,'measured');%Add the noise
end

if(training & FlInterference) % Training with fl interference
    PPM=PPM + i_elect; % Add interference to random noise

elseif(training & ~FlInterference) % Training without fl interference
    %PPM = PPM - do nothing

elseif(~training & FlInterference) % Simulation with fl interference
    sigPower =noisePower*10^(SNR/10);
    sigAmplitude=sqrt(sigPower*L);
    sigPlusShotNoise = awgn((symbolStream * sigAmplitude),SNR,'measured');
    PPM=sigPlusShotNoise + i_elect;

elseif(~training & ~FlInterference) % Simulation without fl interference
    %Find out what signal power we want (shot noise only here)
    sigPower=noisePower*10^(SNR/10);
    sigAmplitude=sqrt(sigPower*L);
    PPM = awgn((symbolStream * sigAmplitude),SNR,'measured');
```

```

else
    %ERROR, should never get here!
    disp('An error has occurred in function: generatePPM');
end

% If we are using a match filter (NOT USED)
if(matchFilter)
    matchFilt=fliplr(one)*(1/sqrt(samplesPerSymbol*1e-9*dutyCycle));
    PPM=conv(matchFilt,PPM);
end


return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% sigConditioner %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function takes the symbol stream and formats it for
%input to the neural network.
%
%INPUTS:-
%signal: The input symbol stream
%
%samplesPerSymbol: The number of 10 nanosecond samples that make up the
%symbol.
%
%numDecimations: This is the number of 'windows' we require and is the
%same as the number of bits in the data stream.
%
%OUTPUTS:-
%conditionedSig: A matrix of conditioned data for the neural network
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [conditionedSig]=sigConditioner(signal, samplesPerSymbol, ...
         numDecimations)

numSamplesPerDecimation = 3; %5 for a 5 bit window

% Keep track of where we are in the symbol stream
start = 1;
index = samplesPerSymbol;
finish=samplesPerSymbol * numSamplesPerDecimation;


%Cut out the samples from the signal
for decimation=1:numDecimations

    % Cut out the samples from the signal and perform a CWT
    % Rem, scales are chosen to match the data rate
    coef=cwt(signal(start:finish),[30 45 60 75 90 120],'sym2');

    % Sort the coefficients in to a single column
    [r,c]=size(coef);
    sig=[];
    for row=1:r
        sig=[sig coef(row,:)];
    end

    %Pre define the holding matrix
    if(decimation==1)
        conditionedSig=zeros(length(sig),numDecimations);
    end

    % Add the column to a matrix where each column represents the
    % coefficients from one 'window'
    conditionedSig(:,decimation)=sig';

    % Index to the next window
    start = start + index;
    finish = finish + index;
end

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% saveResults %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function saves all parameters and results associated with the
%simulation.
%
%INPUT:-
%BER: The BER for a particular SNR
%
%SNR: The corresponding SNR
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function saveResults(BER, SNR);

save -append results BER;
save -append results SNR;

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AI_MP_WLT_OOK_Train %%%%%%%%%%%%%%%%%%%%%%%%
%
%This function carries out network training training.
%
%INPUTS:-
%Net_input: The values the network is trained on
%
%True_values: The true binary representation of the Net_input
%
%Val_impt: Validation data
%
%Val_R: The true binary representation of the Validation data
%
%Tst_impt: Test data
%
%Tst_R: The true binary representation of the Test data
%
%OUTPUTS:-
%Net: The trained neural network
%
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Net,minp,maxp]=AI_MP_WLT_OOK_Train(Net_input,True_values,...
          Val_impt,Val_R,Tst_impt,Tst_R);

%Normalise the input and targets. Matlab function returning the
%normalisation matrices and the normalised data 'pn'.
%Training set
[pn,minp,maxp] = premnmx(Net_input);
```

```
tn=True_values;

%Validation set. Matlab function normalising the validation set.
val.P = tramnmx(Val_impt,minp,maxp);
val.T = Val_R;

%Test set. Matlab function normalising the test set.
tst.P = tramnmx(Tst_inpt,minp,maxp);
tst.T = Tst_R;

%Create the network. Matlab function creating a feedforward network.
net=newff(minmax(pn),[12,2],{'logsig','purelin'},'trainscg');
%'purelin', and 'logsig' are activation functions
%'trainscg' is the training algorithm.

net=init(net);%Matlab function initialising the network weights

%Adjust the training parameters
net.trainParam.show = 10;
net.trainParam.lr = 0.05;
net.trainParam.epochs = 250;
net.trainParam.goal = 1e-14;
net.trainParam.min_grad = 1e-9;

%Train the network
[Net,tr]=train(net, pn, tn,[],[],val,tst);

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AI_WLT_PPM_SIM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function uses the nural network to determine the binary value of the
%signal
%
%Inputs:
%Net - The neural network to use.
%
%Net_input - The data stream to be processed.
%
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%Outputs:
%Net_res - The thresholded network output.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Net_res]=AI_WLT_PPM_SIM(Net, Net_input, minp, maxp);


%Normalise the input in accordance with the trained network
pn = tramnmx(Net_input,minp,maxp);%Matlab function

%Simuate the network
R=sim(Net, pn);%Matlab function

%Step through the simulation results and threshold
[r,c]=size(R);
for i=1:c
    for j=1:r
        temp = R(j,i);
        if(temp>0.5)
```

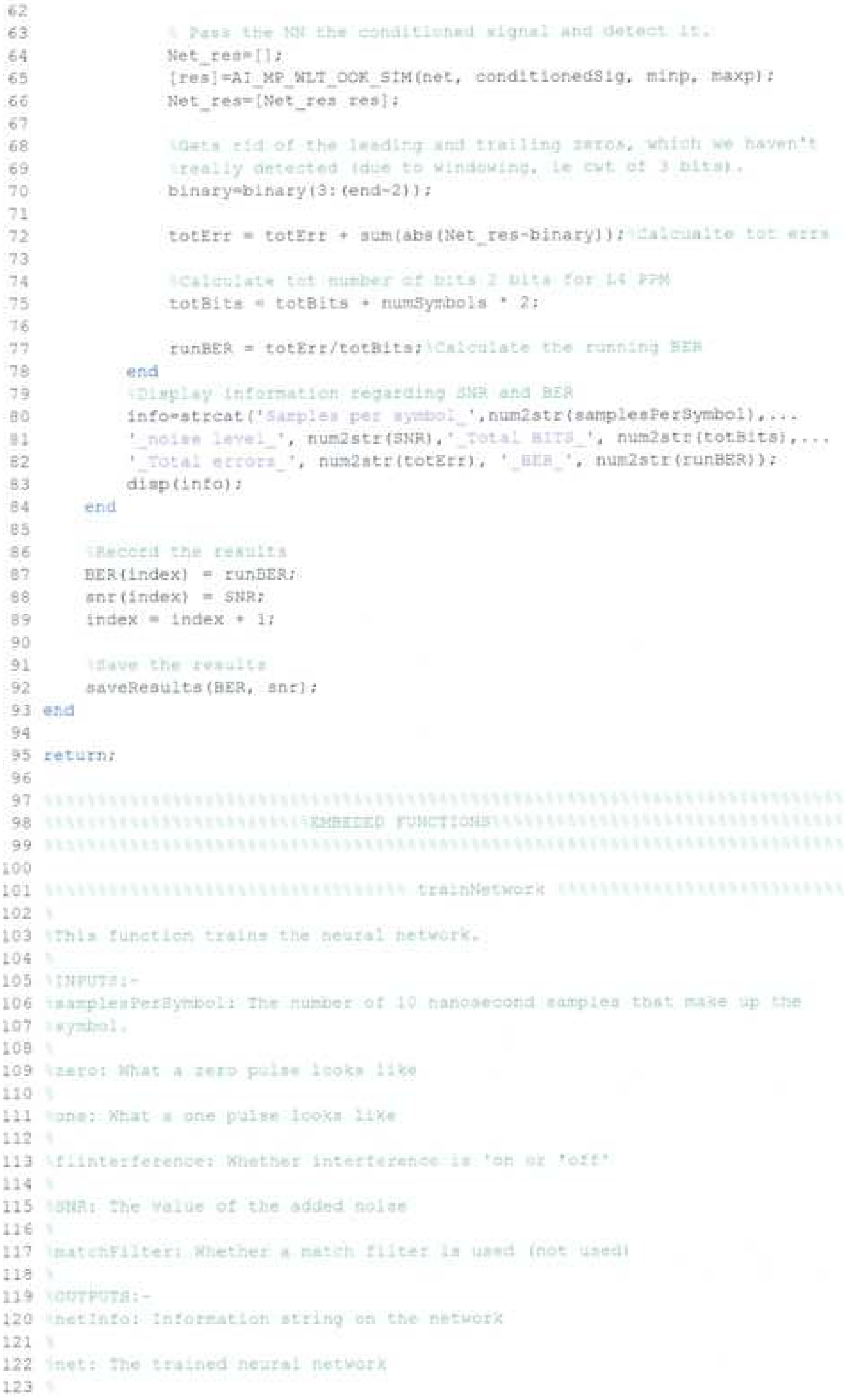


```
            R(j,1)=1;
        else
            R(j,1)=0;
        end
    end
end


Net_res=R;

return;

```

Appendix C

Matlab™ Listing for: Fig. 5.21

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Wlt_AI_SSD %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This function simulates a Wavelet AI Soft Slot Detector
% for L4-PPM with and without fluorescent light interference and
% different noise conditions
%
% This file contains a number of embedded functions and uses external
% functions for fluorescent light interference
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Wlt_AI_SSD

L=4;%4PPM

flInterference = 1; % Fluorescent light interference (1 = on).
matchFilter = 0; % Match filter detection (1 = on) NOT USED.

samplesPerSymbol = 80;%Number of 10 ns pulses per symbol.
temp = 1; save results temp; % Open a results file

%What an PPM '0' and '1' look like.
zero=zeros(1,samplesPerSymbol/L);
one=[ones(1,samplesPerSymbol/L)];

SNR = -12;  %The starting SNR -3
index = 1;  %Used to keep track of which BER and SNR we are recording
runBER = 1; %Running total of the BER when we are in the While loop
totErr = 0; %Keeps track of the total errors for a particular SNR
training=0;%We are not training here
wordBlocks = 2;%The number of times the program goes round the while loop.
numSymbols=250; %The number of symbols each time round the while loop
numDecimations=numSymbols; %The number of 'answers' we expect.

%While the BER is less than 5e-6 and the SNR is less than 35dB, keep going.
while(runBER>5e-6)

    if(runBER>2.5e-3)
        increment = 3;% Slow increment down as BER gets lower (curve steep)
    else
        increment = 0.5;
    end

    SNR = SNR + increment;% SNR valuie

    %Train the neural network
    [net, minp, maxp]=trainNetwork(samplesPerSymbol, zero, one, ...
    flInterference, SNR, matchFilter);
    totErr = 0; %Reset
    totBits = 0; %Reset

    %While the total errors are less than 5 keep going
    while(totErr<5)
        for ctr=1:wordBlocks

            % Generate the OOK signal
            [PPM, slots, binary]=generatePPM(numSymbols, samplesPerSymbol,↙
SNR,...
                zero, one, training, flInterference, matchFilter);

            % Condition it: Use CWT and package it properly for NN
            [conditionedSig]=sigConditioner(PPM, samplesPerSymbol, ...
                numDecimations);
```



```
            % Pass the NN the conditioned signal and detect it.
            Net_res=[];
            [res]=AI_MP_WLT_OOK_SIM(net, conditionedSig, minp, maxp);
            Net_res=[Net_res res];

            %Gets rid of the leading and trailing zeros, which we haven't
            %really detected (due to windowing, ie cwt of 3 bits).
            binary=binary(3:(end-2));

            totErr = totErr + sum(abs(Net_res-binary));%Calcualte tot errs

            %Calculate tot number of bits 2 bits for L4 PPM
            totBits = totBits + numSymbols * 2;

            runBER = totErr/totBits;%Calculate the running BER
        end
        %Display information regarding SNR and BER
        info=strcat('Samples per symbol_',num2str(samplesPerSymbol),...
        '_noise level_', num2str(SNR),'_Total BITS_', num2str(totBits),...
        '_Total errors_', num2str(totErr), '_BER_', num2str(runBER));
        disp(info);
    end

    %Record the results
    BER(index) = runBER;
    snr(index) = SNR;
    index = index + 1;

    %Save the results
    saveResults(BER, snr);
end

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%EMBEDED FUNCTIONS%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% trainNetwork %%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function trains the neural network.
%
%INPUTS:-
%samplesPerSymbol: The number of 10 nanosecond samples that make up the
%symbol.
%
%zero: What a zero pulse looks like
%
%one: What a one pulse looks like
%
%flinterference: Whether interference is 'on or 'off'
%
%SNR: The value of the added noise
%
%matchFilter: Whether a match filter is used (not used)
%
%OUTPUTS:-
%netInfo: Information string on the network
%
%net: The trained neural network
%
```

```
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [net, minp, maxp]=trainNetwork(samplesPerSymbol, zero, one, ...
    flInterference, SNR, matchFilter)

numSymbols = 1000;% The amount of symbols used to train the network.

%Make the data to train with, we need 3 streams for training.
[trainingInput1, slots1]=condTrainingSets(numSymbols, samplesPerSymbol, ...
    zero, one, flInterference, SNR, matchFilter);

[trainingInput2, slots2]=condTrainingSets(numSymbols, samplesPerSymbol, ...
    zero, one, flInterference, SNR, matchFilter);

[trainingInput3, slots3]=condTrainingSets(numSymbols, samplesPerSymbol, ...
    zero, one, flInterference, SNR, matchFilter);

% Get rid of leading & trailing slots as these are not detected.
[net,minp,maxp]=AI_MP_WLT_OOK_Train(trainingInput1, ...
    slots1(2:(length(slots1)-1)),trainingInput2,slots2(2:(length(slots2)-1))...
    ,trainingInput3,slots3(2:(length(slots3)-1)));

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%% condTrainingSets %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function makes training data for the neural network.
%
%INPUTS:-
%numSymbols: The number of PPM symbols required
%
%samplesPerSymbol: The number of 10 nanosecond samples that make up the
%symbol.
%
%zero: What a zero pulse looks like
%
%one: What a one pulse looks like
%
%flInterference: Whether interference is 'on or 'off'
%
%SNR: The value of the added noise
%
%matchFilter: Whether a match filter is used (not used)
%
%OUTPUTS:-
%trainingInput: Data suitably formatted for input to a neural network
%
%slots: The true slot representation of the trainingInput.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [trainingInput, slots]=condTrainingSets(numSymbols, ...
    samplesPerSymbol, zero, one, flInterference, SNR, matchFilter)

training=1;%need to tell 'generateOOK that we are training'
[PPM, slots, binary]=generatePPM(numSymbols, samplesPerSymbol, SNR,...
    zero, one, training, flInterference, matchFilter);

```

```
numDecimations = numSymbols;

[trainingInput]=sigConditioner(PPM, samplesPerSymbol, numDecimations);

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% generatePPM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function produces the data stream.
%
%INPUTS:-
%numSymbols: The number of OOK symbols required
%
%samplesPerSymbol: The number of 10 nanosecond samples that make up the
%symbol.
%
%zero: What a zero pulse looks like
%
%one: What a one pulse looks like
%
%FIinterference: Whether interference is 'on or 'off'
%
%SNR: The value of the added noise
%
%matchFilter: Whether a match filter is used (not used)
%
%training: Whether we are generating data for training or simulation
%
%OUTPUTS:-
%PPM: The prepared symbol stream.
%
%slots: The true slot representation of the data stream
%
%data: The true binary representation of the data stream
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [PPM, slots, binary]=generatePPM(numSymbols, samplesPerSymbol,...
    SNR, zero, one, training, FlInterference, matchFilter);

sampInt = 1e-8;
windowSize=3;

L = 4; % 4 for 4PPM

ZERO_ZERO = [one zero zero zero];
ZERO_ONE = [zero one zero zero];
ONE_ZERO = [zero zero one zero];
ONE_ONE = [zero zero zero one];

if (FlInterference)
    FlI=2e-6; % Fluorescent light current = 2 microamps.
    Ib=FlI;
    ShI=200e-6; % Shot noise current = 200 microamps.
else
    ShI=200e-6; % Shot noise current = 200 microamps.
    FlI = 0;
end

q=1.602e-19;
systemBandWidth = 50e6; %Based on 10 ns sample time
noisePower = q*(ShI + FlI)*systemBandWidth; % Need to get a power value.

% Generate the basic PPM symbolStream
```

```
symbolStream=[];
slots=[];
binary=[];
n=numSymbols+(windowSize-1);
symbols=randint(1,n,[1 4]);
for symbol=1:n
    switch symbols(symbol)
        case 1
            symbolStream=[symbolStream ZERO_ZERO];
            slots=[slots 1 0 0 0];
            binary=[binary 0 0];
        case 2
            symbolStream=[symbolStream ZERO_ONE];
            slots=[slots 0 1 0 0];
            binary=[binary 0 1];
        case 3
            symbolStream=[symbolStream ONE_ZERO];;
            slots=[slots 0 0 1 0];
            binary=[binary 1 0];
        case 4
            symbolStream=[symbolStream ONE_ONE];
            slots=[slots 0 0 0 1];
            binary=[binary 1 1];
        otherwise
            disp('ERROR');
    end
end


% If fluorescent light interference is on
if (FlInterference)
    % The low cycle is 20mS long so start at any time in that period
    interferenceLength = (numSymbols+(windowSize-1))...
        *samplesPerSymbol*sampInt;

    startTime = abs(rand(1,1)) * 20e-3;
    endTime = startTime + interferenceLength;

    %Make the Fl interference
    [i_elect]=fl_model(Ib, sampInt, startTime, endTime);
    i_elect=i_elect(1:(end-1));
end

if(training) % If training add random noise.
    % Find out what signal power we want (shot noise only here)
    sigPower=noisePower*10^(SNR/10);
    sigAmplitude=sqrt(sigPower*L);
    PPM=awgn((symbolStream*sigAmplitude),SNR,'measured');%Add the noise
end

if(training & FlInterference) % Trg with fl interference
    PPM=PPM + i_elect; % Add interference to random noise

elseif(training & ~FlInterference) % Trg without fl interference
    %PPM = PPM - do nothing

elseif(~training & FlInterference) % Simulation with fl interference
    sigPower =noisePower*10^(SNR/10);
    sigAmplitude=sqrt(sigPower*L);
    sigPlusShotNoise = awgn((symbolStream * sigAmplitude),SNR,'measured');
    PPM=sigPlusShotNoise + i_elect;

```

```
elseif(~training & ~F1Interference) % Simulation without f1 interference
    %Find out what signal power we want (shot noise only here)
    sigPower=noisePower*10^(SNR/10);
    sigAmplitude=sqrt(sigPower*L);
    PPM = awgn((symbolStream * sigAmplitude),SNR,'measured');

else
    %ERROR, should never get here!
    disp('An error has occurred in function: generatePPM');
end

% If we are using a match filter (NOT USED)
if(matchFilter)
    matchFilt=fliplr(one)*(1/sqrt(samplesPerSymbol*1e-9*dutyCycle));
    PPM=conv(matchFilt,PPM);
end


return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% sigConditioner %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function takes the symbol stream and formats it for
%input to the neural network.
%
%INPUTS:-
%signal: The input symbol stream
%
%samplesPerSymbol: The number of 10 nanosecond samples that make up the
%symbol.
%
%numDecimations: This is the number of 'windows' we require and is the
%same as the number of bits in the data stream.
%
%OUTPUTS:-
%conditionedSig: A matrix of conditioned data for the neural network
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [conditionedSig]=sigConditioner(signal, samplesPerSymbol, ...
         numDecimations)

numDecimations = (numDecimations * 4)- 6;
numSamplesPerDecimation = 3; %5 for a 5 bit window

% Keep track of where we are in the symbol stream
start = 1;
index = samplesPerSymbol/4;
finish=samplesPerSymbol/4 * numSamplesPerDecimation;


%Cut out the samples from the signal
for decimation=1:numDecimations

    % Cut out the samples from the signal and perform a CWT
    % Sam, scales are chosen to match the data rate
    coef=cwt(signal(start:finish),[30 45 60 75 90 120],'sym2');

    % Sort the coefficients in to a single column
    [r,c]=size(coef);
    sig=[];
    for row=1:r
        sig=[sig coef(row,:)];
    end

    %Pre define the holding matrix
    if(decimation==1)
        conditionedSig=zeros(length(sig),numDecimations);
    end

    % Add the column to a matrix where each column represents the
    % coefficients from one 'window'
    conditionedSig(:,decimation)=sig';

    % Index to the next window
    start = start + index;
    finish = finish + index;
end

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% saveResults %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function saves all parameters and results associated with the
%simulation.
%
%INPUT:-
%BER: The BER for a particular SNR
%
%SNR: The corresponding SNR
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function saveResults(BER, SNR);

save -append results BER;
save -append results SNR;

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AI_MP_WLT_OOK_Train %%%%%%%%%%%%%%%%%%%%%%%%
%
%This function carries out network training training.
%
%INPUTS:-
%Net_input: The values the network is trained on
%
%True_values: The true binary representation of the Net_input
%
%Val_impt: Validation data
%
%Val_R: The true binary representation of the Validation data
%
%Tst_impt: Test data
%
%Tst_R: The true binary representation of the Test data
%
%OUTPUTS:-
%Net: The trained neural network
%
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Net,minp,maxp]=AI_MP_WLT_OOK_Train(Net_input,True_values,...
           Val_impt,Val_R,Tst_impt,Tst_R);

%Normalise the input and targets. Matlab function returning the
%normalisation matrices and the normalised data 'pn'.
%Training set
[pn,minp,maxp] = premnmx(Net_input);
tn=True_values;

%Validation set. Matlab function normalising the validation set.
val.P = tramnmx(Val_impt,minp,maxp);
val.T = Val_R;

%Test set. Matlab function normalising the test set.
tst.P = tramnmx(Tst_impt,minp,maxp);
tst.T = Tst_R;

%Create the network. Matlab function creating a feedforward network.
net=newff(minmax(pn),[6,1],{'logsig','purelin'},'trainscg');
%'purelin', and 'logsig' are activation functions
%'trainscg' is the training algorithm.

net=init(net); %Matlab function initialising the network weights

%Adjust the training parameters
net.trainParam.show = 10;
net.trainParam.lr = 0.05;
net.trainParam.epochs = 250;
net.trainParam.goal = 1e-14;
net.trainParam.min_grad = 1e-9;

%Train the network
[Net,tr]=train(net, pn, tn,[],[],val,tst);

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AI_WLT_PPM_SIM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function uses the nural network to determine the binary value of the
%signal
%
%Inputs:
%Net - The neural network to use.
%
%Net_input - The data stream to be processed.
%
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%Outputs:
%Net_res - The thresholded network output.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Net_res]=AI_MP_WLT_OOK_SIM(Net, Net_input, minp, maxp);

%Normalise the input in accordance with the trained network
pn = tramnmx(Net_input,minp,maxp); %Matlab function

%Simuate the network
R=sim(Net, pn); %Matlab function

%Step through the slots per symbol and find the largest
```

```
R=R(4:end-3);
[r,c]=size(R);
data=[];
L=4;
word=[];
symbol=[];
ctr=1;
while(ctr<c)
    for bit=1:L
        word(bit)=R(ctr);
        ctr=ctr+1;
    end
    index=find(word==max(word));
    switch index
        case 1
            symbol=[symbol 0 0];
        case 2
            symbol=[symbol 0 1];
        case 3
            symbol=[symbol 1 0];
        case 4
            symbol=[symbol 1 1];
    end
end


Net_res=symbol;

return;

```

Appendix D

Matlab™ Listing for: Fig. 6.16

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% wlt_nn_mp_OOK_RZ %%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This function trains neural networks to detect an OOK signal with
% multipath induced interference under different noise conditions and
% then tests them against unknown data using wavelet coefficients as the
% information source.
%
% This file contains a number of embedded functions
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function wlt_nn_mp_OOK_RZ

global sampleInterval;
sampleInterval = 4e-9;

global isiOn;
isiOn = 1;  % For an ISI stream set to '1' for a none ISI stram set to '0'

samplesPerSymbol=4; %The number of 4 ns samples per OOK pulse.
blocks = 10; %The number of times the program goes round the while loop.
SNR = -3;    %The starting SNR
index = 1;   %Used to keep track of which BER and SNR we are recording
runBER = 1;  %Running total of the BER when we are in the While loop
totErr = 0;  %Keeps track oof the total errors for a particular SNR
numSymbols=100; %The number of symbols each time round the while loop
numDecimations=numSymbols; %The number of 'answers' we expect.
save results samplesPerSymbol;

%While the BER is less than 1e-5 and the SNR is less than 50 dB, keep going
while(runBER>1e-5 && SNR<50)

    % If the BER is mor than 1e-4 increment the SNR by 3dB, otherwise
    % increment by 0.5dB
    if(runBER>1e-4)
        increment = 3;
    else
        increment = 1;
    end

    SNR = SNR + increment;% Increment the SNR
    totErr = 0; %Reset
    totBits = 0; %Reset

    info=strcat('SNR_',num2str(SNR));%Information string
    disp(strcat('Samples per symbol_',num2str(samplesPerSymbol),...
        '_Training SNR_', num2str(SNR)));%Display an information string

    %Train the neural network
    [netInfo, net, minp, maxp]=trainNetwork(samplesPerSymbol, SNR);

    %While the total errors are less than 5 keep going
    while(totErr<5)
        for ctr=1:blocks

            % Generate the OOK signal
            [mpOokStream, ookStream, dataRate, data]=ookSource(...
                numSymbols, samplesPerSymbol, SNR);

            % Condition it: Use CWT and package it properly for NN
            [conditionedSig]=sigConditioner(mpOokStream,...
                samplesPerSymbol, numDecimations);

            % Pass the NN the conditioned signal and detect it.
            Net_res=[];
            [res]=AI_MP_WLT_OOK_SIM(net, conditionedSig, minp, maxp);
            Net_res=[Net_res res];

            % Gets rid of the leading and trailing zeros, which we haven't
            % really detected (due to windowing, ie cwt of 3 bits).
            data=data(2:(end-1));

            % Calcualte the total errors
            totErr = totErr + sum(abs(Net_res-data));

            % Calculate the total number of bits
            totBits = totBits + numSymbols;

            % Calculate the running BER
            runBER = totErr/totBits;
        end

        %Display information regarding SNR and BER
        info=strcat('Samples per symbol_',num2str(samplesPerSymbol),...
            '_Net_',num2str(SNR),'_dB_','_noise level_', num2str(SNR),...
            '_Total BITS_', num2str(totBits), '_Total errors_',...
            num2str(totErr), '_BER_', num2str(runBER));
        disp(info);

    end

    % Record the results
    BER(index) = runBER;
    snr(index) = SNR;
    index = index + 1;

    % Save the results
    saveResults(BER, snr);

end

graph(BER, snr);% Construct a results plot

%Use pop-up box to notify simulation end
msgbox('All training and simulation has finished','SIMULATION','help');

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%EMBEDED FUNCTIONS%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% trainNetwork %%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function trains the neural network.
%
%INPUTS:-
%samplesPerSymbol: The number of 1 nanosecond samples that make up the
%symbol. Eg 100Mbps OOK requires 10 x 1 nanosecond samples.
%
%SNR: The value of the added noise
%
%
%OUTPUTS:-
%netInfo: Information string on the network
```

```
%
%net: The trained neural network
%
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [netInfo, net, minp, maxp]=trainNetwork(samplesPerSymbol, SNR)

numSymbols = 800;% The amount of symbols used to train the network.

%Text string regarding network parameters
netInfo=strcat('Samples per symbol_',num2str(samplesPerSymbol),'_',...
    'Number of symbols_',num2str(numSymbols),'_','SNR_',num2str(SNR));

% Make the data to train with, we need 3 streams for training.
[trainingInput1, data1]=condTrainingSets(numSymbols, samplesPerSymbol,...
    SNR);

[trainingInput2, data2]=condTrainingSets(numSymbols, samplesPerSymbol,...
    SNR);

[trainingInput3, data3]=condTrainingSets(numSymbols, samplesPerSymbol,...
    SNR);

% Get rid of leading & trailing zeros we added then train.
[net,minp,maxp]=AI_MP_WLT_OOK_Train(trainingInput1,data1...
    (2:(length(data1)-1)),trainingInput2,data2(2:(length(data2)-1)),...
    trainingInput3,data3(2:(length(data3)-1)));

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%% condTrainingSets %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function makes training data for the neural network.
%
%INPUTS:-
%numSymbols: The number of OOK symbols required
%
%samplesPerSymbol: The number of 1 nanosecond samples that make up the
%symbol. Eg 100Mbps OOK requires 10 x 1 nanosecond samples.
%
%SNR: The value of the added noise
%
%matchFilter: Whether a match filter is used (not used)
%
%OUTPUTS:-
%trainingInput: Data suitably formatted for input to a neural network
%
%data: The binary representation of the trainingInput.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [trainingInput, data]=condTrainingSets(numSymbols,...
    samplesPerSymbol, SNR)

% Generate the training data
[mpOokStream,ookStream, dataRate, data]=ookSource(numSymbols,...
    samplesPerSymbol, SNR);

numDecimations = numSymbols;

```

```
[trainingInput]=sigConditioner(mpOokStream, samplesPerSymbol,...
    numDecimations);

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ookSource %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function produces the data stream.
%
%INPUTS:-
%numSymbols: The number of OOK symbols required
%
%samplesPerSymbol: The number of 1 nanosecond samples that make up the
%symbol. Eg 100Mbps OOK requires 10 x 1 nanosecond samples.
%
%
%SNR: The value of the added noise
%
%
%OUTPUTS:-
%mpOokStream: The prepared symbol stream.
%
%ookStram: The none ISI symbol stream
%
%dataRate: The data rate based on the samples per symbol passed in
%
%data: The binary representation of the data stream
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [mpOokStream,ookStream, dataRate, data]=ookSource(numSymbols,...
    samplesPerSymbol, SNR)

global sampleInterval;

%Work out the actual data rate
dataRate=(1/(samplesPerSymbol*sampleInterval));

%work out what a '1'and '0' pulse looks like
one=[zeros(1,(samplesPerSymbol/2)) ones(1,(samplesPerSymbol/2))];
zero=zeros(1,samplesPerSymbol);

data=randint(1,numSymbols); %Create the random data stream

% Add leading 0s for window, we detect the centre bit from a 3 bit sliding
% window, so the first and last bit don't count
data=[0 data 0];
ookStream=[]; %Make and empty array to hold the OOK pulse stream

%Populate the pulse stream based on the random data
for symbol=1:length(data)
    if(data(symbol)==1)
        ookStream=[ookStream one];
    else
        ookStream=[ookStream zero];
    end
end

[t,h1,h2,Drms1,Drms2]=Multipath_IR;%Calculate the channel characteristics

%Convolve the ideal signal with the chanel impulse response.
mpOokStream=conv(h2,ookStream);

mpOokStream=awgn(mpOokStream,SNR,'measured');%Add the noise
```

```
mpOokStream = mpOokStream/max(mpOokStream);%Normalise to 1

return

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% sigConditioner %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function takes the symbol stream and formats it for
%input to the neural network.
%
%INPUTS:-
%signal: The input symbol stream
%
%samplesPerSymbol: The number of 1 nanosecond samples that make up the
%symbol. Eg 100Mbps OOK requires 10 x 1 nanosecond samples.
%
%numDecimations: This is the number of 'windows' we require and is the
%same as the number of bits in the data stream.
%
%OUTPUTS:-
%conditionedSig: A matrix of conditioned data for the neural network
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [conditionedSig]=sigConditioner(signal, samplesPerSymbol, ...
    numDecimations)

numSamplesPerDecimation = 3; %5 for a 5 bit window

% Keep track of where we are in the symbol stream
start = 1;
index = samplesPerSymbol;
finish=samplesPerSymbol * numSamplesPerDecimation;

% Arrange the signal in a 3 window by numDecimations matrix
for decimation=1:numDecimations

    % Cut out the samples from the signal and perform a CWT
    % Rem, scales are chosen to match the data rate
    coef=cwt(signal(start:finish),[(3:3:12) (15:10:105)],'sym2');

    % Sort the coefficients in to a single column
    [r,c]=size(coef);
    sig=[];
    for row=1:r
        sig=[sig coef(row,:)];
    end

    % Pre define the holding matrix
    if(decimation==1)
        conditionedSig=zeros(length(sig),numDecimations);
    end

    % Add the column to a matrix where each column represents the
    % coefficients from one 'window'
    conditionedSig(:,decimation)=sig';

    % Index to the next window
    start = start + index;
    finish = finish + index;
end

return;
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% saveResults %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function saves all parameters and results associated with the
%simulation.
%
%INPUT:-
%BER: The BER for a particular SNR
%
%snr: The corresponding SNR
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function saveResults(BER, snr);

save -append results BER;
save -append results snr;

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AI_MP_WLT_OOK_Train %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function carries out network training training.
%
%INPUTS:-
%Net_input: The values the network is trained on
%
%True_values: The true binary representation of the Net_input
%
%Val_impt: Validation data
%
%Val_R: The true binary representation of the Validation data
%
%Tst_impt: Test data
%
%Tst_R: The true binary representation of the Test data
%
%OUTPUTS:-
%Net: The trained neural network
%
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Net,minp,maxp]=AI_MP_WLT_OOK_Train(Net_input,True_values,...
    Val_impt,Val_R,Tst_impt,Tst_R);

%Normalise the input and targets. Matlab function returning the
%normalisation matrices and the normalised data 'pn'.
%Training set
[pn,minp,maxp] = premnmx(Net_input);
tn=True_values;

%Validation set. Matlab function normalising the validation set.
val.P = tramnmx(Val_impt,minp,maxp);
val.T = Val_R;

%Test set. Matlab function normalising the test set.
tst.P = tramnmx(Tst_impt,minp,maxp);
tst.T = Tst_R;

%Create the network. Matlab function creating a feedforward network.
```

```
%5 first layer, 1 output layer; 'purelin', and 'logsig' are activation
%functions and 'trainscg' is the training algorithm.
net=newff(minmax(pn),[5,1],{'logsig','purelin'},'trainscg');


net=init(net);%Matlab function initialising the network weights

%Adjust the training parameters
net.trainParam.show = 10;
net.trainParam.lr = 0.05;
net.trainParam.epochs = 250;
net.trainParam.goal = 1e-14;
net.trainParam.min_grad = 1e-9;

%Train the network
[Net,tr]=train(net, pn, tn,[],[],val,tst);

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AI_MP_WLT_OOK_SIM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function uses the nural network to determine the binary value of the
%centre bit from a 5 bit window. Please read the INPUTS & OUTPUTS
%
%Inputs:
%Net - The neural network to use.
%
%Net_input - The data stream to be processed.
%
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%Outputs:
%Net_res - The thresholded network output.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Net_res]=AI_MP_WLT_OOK_SIM(Net, Net_input, minp, maxp);

% Normalise the input in accordance with the trained network. Matlab
% function
pn = tramnmx(Net_input,minp,maxp);

%Simuate the network. Matlab function
R=sim(Net, pn);

%Step through the simulation results and threshold
[r,c]=size(R);

for i=1:c
    for j=1:r
        temp = R(j,i);
        if(temp>0.5)
            R(j,i)=1;
        else
            R(j,i)=0;
        end
    end
end

Net_res=R;
```

```

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Multipath_IR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function determines the channel impulse response using the Carruthers
%& Kahn ceiling bounce method. Please read the INPUTS & OUTPUTS
%
%OUTPUTS:-
%h1: is the impulse response of an infinate plane.
%
%h2 is the corrected response due to room size and Tx, Rx location.
%
% t is the time base vector;
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [t,h1,h2,Drms1,Drms2]=Multipath_IR;

global isiOn;
global sampleInterval;
t=0:(sampleInterval):(1e-7); %time base

H=2; %Height of room above Tx&Rx

c=2.998*10^8; %The speed of light

%The time it takes for the light to get from the Tx to the Rx
%for a co-located Tx and Rx
a=(2*H)/c;

%The impulse response for the co-located Tx&Rx for an infinate plane
%ceiling
h1=(6*(a^6))*((t+a).^-7);

h1=h1/trapz(t,h1);

%RMS Delay Spread
Drms1=(a/12)*sqrt(11/13);

%Correction factor for 'a' where Tx and Rx are not co-located:
%Tx-Rx diagonal approx 5.5m, based on the hypotenuse of an approx
%4 x 4 sqaure in the centre of the room.
%Tx & Rx are 4m appart
s=4/5.5;

%new estimate of 'a'.
a=12*(sqrt(11/13)*(2.1-(5*s)+(20.8*s^2)))*Drms1;

%New Estimate of RMS Delay Spread
Drms2=(a/12)*sqrt(11/13);

%Corrected h.
h2=(6*(a^6))*((t+a).^-7);

h2=h2/sum(h2);

% If the ISI is not required
if (isiOn == 0)
    h2 = 1;
end

return;
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% graph %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function plots a graph of all available BER vs SNR for each Neural
%network.
%
%INPUTS:-
%
%BER: An array of BERs
%
%snr, Corresponding array of SNRs
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function graph(BER, snr)


semilogy(snr, BER);


return;
```

Appendix E

Matlab™ Listing for: Fig. 6.21

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% wlt_nn_mp_BSD %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This function simulates a Wavelet AI Binary Symbol Detector
% for L4-PPM with and without ISI and different noise conditions
%
% This file contains a number of embedded functions.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function wlt_nn_mp_BSD

global sampleInterval;
sampleInterval = 4e-9;

global samplesPerSymbol;
samplesPerSymbol = 4;

global isiOn;
isiOn = 1;  % For an ISI stream set to '1' for a none ISI stram set to '0'

temp = 1; save results temp; % Open a results file

%What an PPM '0' and '1' look like.
zero=zeros(1,samplesPerSymbol/4);
one=[ones(1,samplesPerSymbol/4)];

SNR = -3;  %The starting SNR -3
index = 1;  %Used to keep track of which BER and SNR we are recording
runBER = 1; %Running total of the BER when we are in the While loop
totErr = 0; %Keeps track of the total errors for a particular SNR
training=0;%We are not training here
wordBlocks = 2;%The number of times the program goes round the while loop.
numSymbols=500; %The number of symbols each time round the while loop
numDecimations=numSymbols; %The number of 'answers' we expect.

%While the BER is less than 5e-6 and the SNR is less than 35dB, keep going.
while(runBER>5e-6)

    % Slow increment down as BER gets lower
    if(runBER>1e-3)
        increment = 3;
    else
        increment = 0.5;
    end

    SNR = SNR + increment;

    %Train the neural network
    [net, minp, maxp]=trainNetwork(zero, one, SNR);
    totErr = 0; %Reset
    totBits = 0; %Reset

    %While the total errors are less than 5 keep going
    while(totErr<5)
        for ctr=1:wordBlocks

            % Generate the OOK signal
            [PPM, data, symbols]=generatePPM(numSymbols, SNR, zero, one);

            % Condition it: Use CWT and package it properly for NN
            [conditionedSig]=sigConditioner(PPM, numDecimations);

            % Pass the NN the conditioned signal and detect it.
            Net_res=[];
            [res]=AI_MP_WLT_PPM_SIM(net, conditionedSig, minp, maxp);
            Net_res=[Net_res res];

            %Gets rid of the leading and trailing zeros, which we haven't
            %really detected (due to windowing, ie cwt of 3 bits).
            data=data(:,2:(end-1));

            %Calcualte tot errors
            totErr = totErr + sum(sum(abs(Net_res-data)));

            %Calculate tot number of bits 2 bits for L4 PPM
            totBits = totBits + numSymbols * 2;

            runBER = totErr/totBits;%Calculate the running BER

        end

        %Display information regarding SNR and BER
        info=strcat('Samples per symbol_',num2str(samplesPerSymbol),...
        '_noise level_', num2str(SNR),'_Total BITS_', num2str(totBits),...
        '_Total errors_', num2str(totErr), '_BER_', num2str(runBER));
        disp(info);

    end
    %Record the results
    BER(index) = runBER;
    snr(index) = SNR;
    index = index + 1;

    % Save the results
    saveResults(BER, snr);

end

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%EMBEDED FUNCTIONS%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% trainNetwork %%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function trains the neural network.
%
%INPUTS:-
%
%zero: What a zero pulse looks like
%
%one: What a one pulse looks like
%
%SNR: The value of the added noise
%
%
%OUTPUTS:-
%netInfo: Information string on the network
%
%net: The trained neural network
%
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
```

```
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [net, minp, maxp]=trainNetwork(zero, one, SNR)

numSymbols = 1600;% The amount of symbols used to train the network.

%Make the data to train with, we need 3 streams for training.
[trainingInput1, data1]=condTrainingSets(numSymbols, zero, one, SNR);

[trainingInput2, data2]=condTrainingSets(numSymbols, zero, one, SNR);

[trainingInput3, data3]=condTrainingSets(numSymbols, zero, one, SNR);

% Get rid of leading & trailing zeros we added then train.
[net,minp,maxp]=AI_MP_WLT_PPM_Train(trainingInput1, ...
    data1(:,2:(length(data1)-1)),trainingInput2,data2(:,2:(length(data2)-1))...
    ,trainingInput3,data3(:,2:(length(data3)-1)));

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%% condTrainingSets %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function makes training data for the neural network.
%
%INPUTS:-
%numSymbols: The number of PPM symbols required
%
%zero: What a zero pulse looks like
%
%one: What a one pulse looks like
%
%SNR: The value of the added noise
%
%
%OUTPUTS:-
%trainingInput: Data suitably formatted for input to a neural network
%
%data: The binary representation of the trainingInput.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [trainingInput, data]=condTrainingSets(numSymbols, zero, one, SNR);

[OOK, data, symbols]=generatePPM(numSymbols, SNR, zero, one);

numDecimations = numSymbols;

[trainingInput]=sigConditioner(OOK, numDecimations);

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% generatePPM %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function produces the data stream.
%
%INPUTS:-
%numSymbols: The number of OOK symbols required
%
%zero: What a zero pulse looks like
%
%one: What a one pulse looks like
```

```
%
%SNR: The value of the added noise
%
%OUTPUTS:-
%PPM: The prepared symbol stream.
%
%symbols: a number between 1 and 4 representing the pulse position
%
%data: The binary representation of the data stream
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [PPM, data, symbols]=generatePPM(numSymbols, SNR, zero, one);

global sampleInterval;
windowSize = 3;

L = 4; % 4 for 4PPM

% the symbols
ZERO_ZERO = [one zero zero zero];
ZERO_ONE = [zero one zero zero];
ONE_ZERO = [zero zero one zero];
ONE_ONE = [zero zero zero one];

% Generate the basic PPM symbolStream
symbolStream=[];
data=[];
n=numSymbols+(windowSize-1);
symbols=randint(1,n,[1 4]);
for symbol=1:n
    switch symbols(symbol)
        case 1
            symbolStream=[symbolStream ZERO_ZERO];
            data=[data [0 0]'];
        case 2
            symbolStream=[symbolStream ZERO_ONE];
            data=[data [0 1]'];
        case 3
            symbolStream=[symbolStream ONE_ZERO];;
            data=[data [1 0]'];
        case 4
            symbolStream=[symbolStream ONE_ONE];
            data=[data [1 1]'];
        otherwise
            disp('ERROR');
    end
end

[t,h1,h2,Drms1,Drms2]=Multipath_IR;% Calculate the channel characteristics

% Convolve the ideal signal with the chanel impulse response.
PPM=conv(h2,symbolStream);

PPM=awgn(PPM,SNR,'measured');% Add the noise

PPM = PPM/max(PPM);% Normalise to 1

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%% sigConditioner %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function takes the symbol stream and formats it for
%input to the neural network.
```

```
%
%INPUTS:-
%signal: The input symbol stream
%
%numDecimations: This is the number of 'windows' we require and is the
%same as the number of bits in the data stream.
%
%OUTPUTS:-
%conditionedSig: A matrix of conditioned data for the neural network
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [conditionedSig]=sigConditioner(signal, numDecimations)

global samplesPerSymbol;

numSamplesPerDecimation = 3; %5 for a 5 bit window

% Keep track of where we are in the symbol stream
start = 1;
index = samplesPerSymbol;
finish=samplesPerSymbol * numSamplesPerDecimation;


%Cut out the samples from the signal
for decimation=1:numDecimations

    % Cut out the samples from the signal and perform a CWT
    % Rem, scales are chosen to match the data rate
    coef=cwt(signal(start:finish),[1:1:20 25:5:50],'sym2');

    % Sort the coefficients in to a single column
    [r,c]=size(coef);
    sig=[];
    for row=1:r
        sig=[sig coef(row,:)];
    end

    %Pre define the holding matrix
    if(decimation==1)
        conditionedSig=zeros(length(sig),numDecimations);
    end

    % Add the column to a matrix where each column represents the
    % coefficients from one 'window'
    conditionedSig(:,decimation)=sig';

    % Index to the next window
    start = start + index;
    finish = finish + index;

end

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% saveResults %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function saves all parameters and results associated with the
%simulation.
%
%INPUT:-
%BER: The BER for a particular SNR
%
```

```
%SNR: The corresponding SNR
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function saveResults(BER, SNR);

save -append results BER;
save -append results SNR;

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AI_MP_WLT_PPM_Train %%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function carries out network training training.
%
%INPUTS:-
%Net_input: The values the network is trained on
%
%True_values: The true binary representation of the Net_input
%
%Val_impt: Validation data
%
%Val_R: The true binary representation of the Validation data
%
%Tst_impt: Test data
%
%Tst_R: The true binary representation of the Test data
%
%OUTPUTS:-
%Net: The trained neural network
%
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Net,minp,maxp]=AI_MP_WLT_PPM_Train(Net_input,True_values,...
          Val_impt,Val_R,Tst_impt,Tst_R);

%Normalise the input and targets. Matlab function returning the
%normalisation matrices and the normalised data 'pn'.
%Training set
[pn,minp,maxp] = premnmx(Net_input);
tn=True_values;

%Validation set. Matlab function normalising the validation set.
val.P = tramnmx(Val_impt,minp,maxp);
val.T = Val_R;

%Test set. Matlab function normalising the test set.
tst.P = tramnmx(Tst_impt,minp,maxp);
tst.T = Tst_R;

%Create the network. Matlab function creating a feedforward network.
net=newff(minmax(pn),[12,2],{'logsig','purelin'},'trainscg');
%'purelin', and 'logsig' are activation functions
%'trainscg' is the training algorithm.

net=init(net);%Matlab function initialising the network weights

%Adjust the training parameters
net.trainParam.show = 10;
net.trainParam.lr = 0.05;
net.trainParam.epochs = 250;
net.trainParam.goal = 1e-14;
net.trainParam.min_grad = 1e-9;

%Train the network
[Net,tr]=train(net, pn, tn,[],[],val,tst);

return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AI_WLT_PPM_SIM %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function uses the nural network to determine the binary value of the
%signal
%
%Inputs:
%Net - The neural network to use.
%
%Net_input - The data stream to be processed.
%
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%Outputs:
%Net_res - The thresholded network output.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Net_res]=AI_MP_WLT_PPM_SIM(Net, Net_input, minp, maxp);
%Inputs:
%Net - The neural network to use.
%
%Net_input - The data stream to be processed.
%
%minp: A normalisation matrix used to prepare network inputs
%
%maxp: A normalisation matrix used to prepare network inputs
%
%Outputs:
%Net_res - The thresholded network output.

%Normalise the input in accordance with the trained network
pn = tramnmx(Net_input,minp,maxp);%Matlab function

%Simuate the network
R=sim(Net, pn);%Matlab function

%Step through the simulation results and threshold
[r,c]=size(R);
for i=1:c
    for j=1:r
        temp = R(j,i);
        if(temp>0.5)
            R(j,i)=1;
        else
            R(j,i)=0;
        end
    end
end


Net_res=R;

```

```
return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Multipath_IR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%This function determines the channel impulse response using the Carruthers
%& Kahn ceiling bounce method. Please read the INPUTS & OUTPUTS
%
%OUTPUTS:-
%h1: is the impulse response of an infinate plane.
%
%h2 is the corrected response due to room size and Tx, Rx location.
%
% t is the time base;
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [t,h1,h2,Drms1,Drms2]=Multipath_IR;

global isiOn;
global sampleInterval;
t=0:(sampleInterval):(1e-7); %time base

H=2; %Height of room above Tx&Rx

c=2.998*10^8; %The speed of light

%The time it takes for the light to get from the Tx to the Rx
%for a co-located Tx and Rx
a=(2*H)/c;

%The impulse response for the co-located Tx&Rx for an infinate plane
%ceiling
h1=(6*(a^6))*((t+a).^-7);

h1=h1/trapz(t,h1);

%RMS Delay Spread
Drms1=(a/12)*sqrt(11/13);

%%Correction factor for 'a' where Tx and Rx are not co-located:
%Tx-Rx diagonal approx 5.5m, based on the hypotenuse of an approx
%4 x 4 sqaure in the centre of the room.
%Tx & Rx are 4m appart
s=4/5.5;

%New estimate of 'a'.
a=12*(sqrt(11/13)*(2.1-(5*s)+(20.8*s^2)))*Drms1;

%New Estimate of RMS Delay Spread
Drms2=(a/12)*sqrt(11/13);

%Corrected h.
h2=(6*(a^6))*((t+a).^-7);

h2=h2/sum(h2);

% If the ISI is not required
if (isiOn == 0)
    h2 = 1;
end

return;
```

References

[1] Michael Stich, "Why hybrid architectures make sense for home networking", June 2006. http://www.embedded.com/showArticle.jhtml?articleID=189500623. 20/04/2007.

[2] IEEE 802.11b-1999, Institute of Electrical and Electronic Engineers, 1999.

[3] IEEE 802.11g-2003, Institute of Electrical and Electronic Engineers, 2003.

[4] www.etsi.org. 20/04/2007.

[5] http://en.wikipedia.org/wiki/HIPERLAN. 20/04/2007

[6] "Status of Project IEEE 802.11n," http://grouper.ieee.org/groups/802/11/Reports/tgn_update.htm. 20/04/2007.

[7] C. Nobel, "5 Things You Need To Know About 802.11n," GovernmentVAR, September 2006. http://www.crn.com/networking/192600474?queryText=802.11n. 20/04/2007.

[8] ZigBee Alliance, http://www.zigbee.org/en/index.asp. 20/04/2007.

[9] M. Galeev, "Home networking with Zigbee," Embeded System Design, April 2004. www.embedded.com/showArticle.jhtml?articleID=18902431,'Home networking with Zigbee'. 20/04/2007.

[10] Infrared Data Association Releases IrDA Global Market Report 2006. http://www.irda.org/displaycommon.cfm?an=1&subarticlenbr=17. 20/04/2007.

[11] www.irda.org. 20/04/2007.

[12] http://irda.org/displaycommon.cfm?an=1&subarticlenbr=7. 20/04/2007.

[13] http://en.wikipedia.org/wiki/TEMPEST. 20/04/2007.

[14] D. Robertson, "MoD seeks battlefield innovators," *The Times*, 18th October 2006.

[15] S. Hagihira, M. Takashina, T. Mori, N. Taenaka, T. Mashimo and I. Yoshiya, "Infrared transmission of electronic information via LAN in the operating room," *Journal of Clinical Monitoring and Computing*. vol. 16. no. 3, pp. 171-175, 2000.

[16] D. Shin, K. Shin, I. Kim, K. Park, T. Lee, S. Kim, K. Lim and S. Huh, "low-power hybrid wireless network for monitoring infant incubators," *Medical Engineering & Physics*, vol. 27, iss. 8, pp. 713-716, 2005.

[17] S. Jivkov and S. Jivkova, "Compact low-power IR data link for EEG systems," *9th World Multi-Conference On Systemics, Cybernetics and Informatics*, vol. 5, pp. 47-50, 2005.

[18] T. Komine, S. Haruyama and M. Nakagawa , "A study of shadowing on indoor visible-light wireless communication utilizing plural white LED lightings," *Wireless Personal Communications*, vol. 34, no. 1-2, July 2005 .

[19] L. Zeng and D. O'Brien, "Indoor optical wireless communications with visible LED light," The Institute of Physics Optical Group Meeting*: The Optics Of Free Space Optical Communications*, December 2006.

[20] P. Amirshahi and M. Kaverhad, "Broadband access over medium and low voltage power-lines and use of white light emitting diodes for indoor communications," *3rd IEEE Consumer Communication and Networking Conference*, CCNC 2006.

[21] C. Petrie, "Defending FPGAs," *Electronics Weekly*, no. 2284, p. 21, 2007.

[22] J. R. Barry, *Wireless Infrared Communications*. Kluwer Academic, U.S.A., 1994.

[23] J. M. Kahn and J. R. Barry, "Wireless infrared communications," *Proceedings of the IEEE,* vol. 85, no. 2, pp. 265-298, February 1997.

[24] S. Hranilovic, "On the design of bandwidth efficient signalling for indoor wireless optical channels," *International Journal of Communication Systems*, vol. 18, iss. 3, pp. 205-228, 2005

[25] J. B. Carruthers and J. M. Kahn, "Modeling of nondirected wireless infrared channels," *IEEE Transactions on Communications*, vol. 45, no. 10, pp. 1260-1268, October 1997.

[26] A. J. C. Moreira, A. M. R. Tavares, R. T. Valadas and A. M. de Oliveira Duarte, "Modulation methods for wireless infrared transmission systems: performance under ambient light noise and interference," in *Proceedings of SPIE Conference on Wireless Data Transmission*, Philadelphia, U.S.A., vol. 2601, pp. 226-237, October 1995,

[27] E. A. Lee and D. G. Messerschmitt, *Digital Communication*, Second Edition. Kluwer Academic, U.S.A., 1994.

[28] A. M. Street, P. N. Stavrinou, D. C. O'Brien and D. J. Edwards, "Indoor optical wireless systems – a review," *Optical and Quantum Electronics*, vol. 29, pp. 349-378, 1997.

[29] P. P. Smyth, P. L. Eardley, K. T. Dalton, D. R. Wisely, P. McKee and D. Wood, "Optical wireless: a prognosis," in *Proceedings of SPIE Conference on Wireless Data Transmission*, Philadelphia, U.S.A., vol. 2601, pp. 212-225, October 1995

[30] O. Bouchet, H. Sizun, C. Boisrobert, F. Fornel and P. Favennec, *Free-Space Optics – Propagation and Communication*, ISTE Ltd,U.K., 2006. ISBN: 1905209029.

[31] www.sagentia.com/Sectors/Telecoms_media/Broadband_3G_and_WLAN. Aspx. 06/04/2007.

[32] www.jvcvictor.co.jp/english/pro/online/palcatalog/PALcataloghome.html. 07/04/2007.

[33] F. R. Gfeller, H. R. Muller and P. Vettiger, "Infrared communication for in-house applications," in *Proceedings of IEEE Conference on Computer Communication*, Washington DC, U.S.A., pp. 132-138, September 1978.

[34] F. R. Gfeller and U. H. Bapst, "Wireless in-house data communication via diffuse infrared radiation," *Proceedings of the IEEE*, vol. 67, no. 11, pp. 1474-1486, November 1979.

[35] J. M. Kahn, W. J. Krause and J. B. Carruthers, "Experimental characterization of non-directed indoor infrared channels," *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 1613-1623, February/March/April 1995.

[36] R. Ramirez-Iniguez and R. Green, "Optical antenna design for indoor optical wireless communication systems," *International Journal of Communication Systems*, vol. 18, no. 3, pp. 229-245, 2005.

[37] E. Mutafungwa and L. Yong, "Performance of indoor infrared wireless CDMA systems with angle diversity," *International Journal of Millimeter Waves*, vol. 25, no. 2, pp. 365-381, 2004.

[38] G. Yun and M. Kavehrad, "Spot diffusing and fly-eye receivers for indoor infrared wireless communications," in *Proceedings of the 1992 IEEE Conference on Selected Topics in Wireless Communications,* Vancouver, Canada, pp. 286-292, June 1992.

[39] A. Al-Ghamdi and J. Elmirghani, "Performance analysis of mobile optical wireless systems employing a novel beam clustering method and diversity detection," *IEE Proceedings: Optoelectronics*, vol. 151, no. 4, pp. 223-231, 2004.

[40] A. Sivabalan and J. John, "Improved power distribution in diffuse indoor optical wireless systems employing multiple transmitter configurations," *Optical and Quantum Electronics*, vol. 38, no. 8, pp. 711-725, 2006.

[41] A. Al-Ghamdi and J. Elmirghani "Analysis of diffuse optical wireless channels employing spot-diffusing techniques, diversity receivers, and combining schemes," *IEEE Transactions on Communications*, vol. 52, no. 10, pp. 1622-1631, 2004.

[42] J. M. Kahn, P. Djahani, A. G. Weisbin, K. T. Beh, A. P. Tang and R. You, "Imaging diversity receivers for high-speed infrared wireless communication," *IEEE Communications Magazine*, vol. 36, no. 12, pp. 88-94, December 1998.

[43] R. T. Valadas and A. M. de Oliveira Duarte, "Sectored receivers for indoor wireless optical communication systems," in *Proceedings of the 5th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, The Hague, Netherlands, pp. 1090-1095, September 1994.

[44] C. R. A. T. Lomba, R. T. Valadas and A. M. de Oiveira Duarte, "Sectored receivers to combat the multipath dispersion of the indoor optical channel," in *Proceedings of the 6th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Toronto, Canada, 27-29, pp. 321-325, September 1995.

[45] A. M. R. Tavares, R. J. M. T. Valadas and A. M. de Olveira Duarte, "Performance of an optical sectored receiver for indoor wireless communication systems in presence of artificial and natural noise sources," in *Proceedings of SPIE Conference on Wireless Data Transmission*, Philadelphia, U.S.A., vol. 2601, pp. 264-273, October 1995.

[46] R.T. Valadas, A.R. Tavares and A.M. de Olveira Duarte, "Angle diversity to combat the ambient noise in indoor optical wireless communication systems," *International Journal of Wireless Information Networks*, vol. 4, no. 4, pp. 275-288, 1997.

[47] K. L. Sterckx, J. M. H. Elmirghani and R. A. Cryan, "Pyramidal fly-eye detection antenna for optical wireless systems," in *Proceedings of IEE Colloquium on Optical Wireless Communications*, London, U.K., pp. 5/1-5/6, June 1999.

[48] K. L. Sterckx, J. M. H. Elmirghani and R. A. Cryan, "Sensitivity assessment of a three-segment pyramidal fly-eye detector in a semidisperse optical wireless communication link," *IEE Proceedings - Optoelectronics*, vol. 147, no. 4, pp. 286–294, August 2000.

[49] K. L. Sterckx, J. M. H. Elmirghani and R. A. Cryan, "Three-segment pyramidal fly-eye detection antenna for optical wireless communication systems under the constraint of ambient noise introduced by highly directive spotlights," *International Journal of Communication Systems*, vol. 13, no. 7 & 8, pp. 577-588, November-December 2000.

[50] A. P. Tang, J. M. Kahn and K. P. Ho, "Wireless infrared communication links using multi-beam transmitters and imaging receivers," in *Proceedings of IEEE International Conference on Communications*, Dallas, U.S.A., 23-27 June 1996, pp. 180-186.

[51] S. Jivkova and M. Kavehrad, "Indoor wireless infrared local access, multi-spot diffusing with computer generated holographic beam-splitters," in *Proceedings of IEEE International Conference on Communications*, Vancouver, Canada, vol. 1, pp. 604-608, June 1999.

[52] J. B. Carruthers and J. M. Kahn, "Angle diversity for nondirected wireless infrared communication," *IEEE Transactions on Communications*, vol. 48, no. 6, pp. 960-969, June 2000.

[53] S. Jivkova and M. Kavehrad, "Multispot diffusing configuration for wireless infrared access," *IEEE Transactions on Communications*, vol. 48, no. 6, pp. 970-978, June 2000.

[54] P. Djahani and J.M. Kahn, "Analysis of infrared wireless links employing multibeam transmitters and imaging diversity receivers," *IEEE Transactions on Communications*, vol. 48, no. 12, pp. 2077-2088, December 2000.

[55] M. Castill-Vázquez and A. Puerta-Notario, "Single-channel imaging receiver for optical wireless communications," *IEEE Communications Letters*, vol. 9, no. 10, pp. 897-899. 2005.

[56] D. Takase and T. Ohtsuki, "Spatial multiplexing in optical wireless MIMO communications over indoor environment," *IEICE Transactions of Communications*, vol. E89-B, no. 4, pp.1364-1371, 2006.

[57] D. R. Wisely, "A 1Gbit/s optical wireless tracked architecture for ATM delivery," in *Proceedings of IEE Colloquium on Optical Free Space Communication Links*, London, U.K., pp. 14/1-14/7, February 1996.

[58] J. B. Carruthers and J. M. Kahn, "Modeling of nondirected wireless infrared channels," in *Proceedings of IEEE International Conference on Communications*, Dallas, U.S.A., vol. 2, pp. 1227-1231, June 1996.

[59] J. R. Barry, J. M. Kahn, W. J. Krause, E. A. Lee and D. G. Messerschmitt, "Simulation of Multipath Impulse Response for Indoor Optical Channels," *IEEE Journal on Selected Areas in Communications*, vol. 11, issue 3, pp. 367-379, 1993.

[60] C. R. Lomba, R. T. Valadas and A. M. de Oliveira Duarte, "Efficient simulation of the indoor wireless optical channel," *International Journal of Communication Systems*, vol 13, issue 7-8, pp. 537-549, 2000.

[61] J. B. Carruthers and P. Kannan, "Iterative Site-Based Modelling for Wireless Infrared Channels," *IEEE Transactions on Antennas and Propagation*, vol. 50, issue 5, pp. 759-766, May 2002.

[62] V. Jungnickel, V. Pohl, S. Nönnig and C. von Helmolt, "A Physical Model of the Wireless Infrared Communication Channel", *IEEE Journal On Selected Areas in Communications,* vol. 20, No. 3, pp. 631-640, April 2002.

[63] A. Sivabalan and J. John, "Modelling and simulation of indoor optical wireless channels: a review," *IEEE Proceedings - Conference on Convergent Technologies for the Asia-Pacific Region*, vol. 2, pp. 1082-1085, 2003.

[64] N. Hayasaka and T. Ito, "Channel modelling of nondirected wireless infrared indoor diffuse link," *Electronics and Communications in Japan, Part I: Communications*, vol. 90, no. 6, pp. 9-19, 2007.

[65] D. Biosca Rojas and J. Lopez, "Generalisation of monte carlo ray tracing algorithm for the calculation of the impulse response on indoor wireless infrared channels," *Universidad, Ciencia y Technoligia*, vol. 9, no. 33, pp. 17-25, 2005.

[66] A. Mihaescu, M. Otesteanu, L. Ghise, M. Telescu and B. Pascal, "Reduced size model method for diffuse optical indoor wireless channel characterization," *WSEAS Transactions on communications*, vol. 5, no. 2, pp. 155-160, 2006.

[67] S. Arumugam and J. John, "Effect of transmitter positions on channel bandwidth in diffuse indoor multi-transmitter optical wireless systesm," *Proceedings of SPIE – Optical Transmission, Switching and Subsystems III*, vol. 6021 I, 2005.

[68] O. Gonzalez, S. Rodriguez, R. Perez-Jimenez, R. Mendoza and A. Ayala, "Error analysis of the simulated impulse response on indoor wireless optical channels using a Monte Carlo-based ray-tracing algorithm," *IEEE Transactions on Communications*, vol. 53, no. 1, pp. 124-130, 2005.

[69] J. Carruthers, S. Carroll and P. Kannan, "Propagtion modelling for indoor optical wireless communication using fast multi receiver channel estimation," *IEE proceedings – Optoelectronics*, vol. 150, no. 5, 2003.

[70] J. Carruthers and S. Carroll, "Statisitical impulse response models for indoor optical wireless channels," *International Journal of Communication Systems*, vol. 18, no. 3, 2005.

[71] Vishay Semiconductors Document No. 82502, Rev. 1.2, July 03, "*Eye Safety of Diode Emitters*". www.vishay.com/docs/82502/82502.pdf. 06/04/07

[72] A. R. Hayes, "Digital Pulse Interval Modulation for Indoor Optical Wireless Communication Systems," Ph.D. Thesis, Sheffield Hallam University, Sheffield, U.K., 2002.

[73] Vishay Semiconductors Document No. 80113, February 02, "*Physics and Technology*". http://www.vishay.com/docs/80113/80113.pdf. 06/04/07

[74] A. J. C. Moreira, R. T. Valadas and A. M. De Oliveira Duarte, "Characterisation and modelling of artificial light interference in optical wireless communication systems," in *Proceedings of the 6th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Toronto, Canada, pp. 326-331, September 1995.

[75] A. J. C. Moreira, R. T. Valadas and A. M. de Oliveira Duarte, "Optical interference produced by artificial light," *Wireless Networks*, vol. 3, no. 2, pp. 131-140, 1997.

[76] A. J. C. Moreira, R. T. Valadas and A. M. de Oliveira Duarte, "Performance of infrared transmission systems under ambient light interference," *IEE Proceedings - Optoelectronics*, vol. 143, no. 6, pp. 339-346, December 1996.

[77] R. Narasimhan, M. D. Audeh and J. M. Kahn, "Effect of electronic-ballast fluorescent lighting on wireless infrared links," *IEE Proceedings - Optoelectronics*, vol. 143, no. 6, pp. 347-354, December 1996.

[78] A. J. C. Moreira, R. T. Valadas and A. M. de Oliveira Duarte, "Reducing the effects of artificial light interference in wireless infrared transmission systems," in *Proceedings of IEE Colloquium on Optical Free Space Communication Links*, London, U.K., pp. 5/1-5/10, 1996.

[79] R. Narasimhan, M. D. Audeh and J. M. Kahn, "Effect of electronic-ballast fluorescent lighting on wireless infrared links," in *Proceedings of IEEE International Conference on Communications*, Dallas, U.S.A., pp. 1213-1219, June 1996.

[80] K. Samaras, A. M. Street, D. C. O'Brien and D. J. Edwards, "Error rate evaluation of wireless infrared links," in *Proceedings of IEEE International Conference on Communications*, Atlanta, U.S.A., vol. 2, pp. 826-831, June 1998.

[81] A. M. R. Tavares, R. J. M. T. Valadas and A. M. de Oliveira Duarte, "Performance of wireless infrared transmission systems considering both ambient light interference and inter-symbol interference due to multipath dispersion," in *Proceedings of SPIE Conference on Optical Wireless Communications*, Boston, U.S.A., vol. 3532, pp. 82-935, November 1998.

[82] K. K. Wong, T. O'Farrell and M. Kiatweerasakul, "Infrared wireless communication using spread spectrum techniques," *IEE Proceedings - Optoelectronics*, vol. 147. no. 4, pp. 308-314, August 2000.

[83] K. K. Wong, T. O'Farrell and M. Kiatweerasakul, "The performance of optical wireless OOK, 2-PPM and spread spectrum under the effects of multipath dispersion and artificial light interference," *International Journal of Communication Systems*, vol. 13, no. 7 & 8, pp. 551-576, November-December 2000.

[84] S. Lee, "Reducing the effects of ambient noise light in an indoor optical wireless system using polarizers," *Microwave and Optical Technology Letters*, vol. 40, no. 3, 2004.

[85] A. R. Hayes, Z. Ghassemlooy, N. L. Seed and R. McLaughlin, "Baseline-wander effects on systems employing digital pulse-interval modulation," *IEE Proceedings - Optoelectronics*, vol. 147. no. 4, pp. 308-314, August 2000.

[86] N. M. Aldibbiat, Z. Ghassemlooy and R. McLaughlin, "Dual header pulse interval modulation for dispersive indoor optical wireless communication systems," *IEE Proceedings – Circuits Devices and Systems*, Vol. 149. No. 3, pp. 187-192, 2002.

[87] Z. Ghassemlooy and A. R. Hayes, "Digital pulse interval modulation for IR communication systems – a review," *International Journal of Communication Systems*, Vol. 13, No. 7-8, pp. 519 – 536, 2000.

[88] T. O'Farrell and K. K. Wong, "Complementary sequence inverse keying for indoor wireless infrared channels," *Electronics Letters*, Vol. 40, No. 4, pp. 257–259, 2004.

[89] U. Sethakaset and T. A. Gulliver, "Differential Amplitude Pulse-Position Modulation for Indoor Wireless Optical Channels," *GLOBECOM - IEEE Global Telecommunications Conference*, vol. 3, pp 1867-1871, 2004.

[90] D. Shiu, J. M. Kahn, "Differential Pulse Position Modulation For Power-Efficient Wireless Infrared Communication," *IEEE Global Telecommunications Conference*, vol. 1, pp. 219-224, 1998.

[91] T. Lueftner, C. Kroepl,. M. Huemer, R. Hagelauer and R. Weigel, "Edge-position modulation for high-speed wireless infrared communications," *IEE Proceedings -- Optoelectronics*, vol. 150, issue 5, pp. 427-437, 2003.

[92] L. W. Couch II, *Digital and Analog Communication Systems*, Fifth Edition. Prentice-Hall, U.S.A., 1997.

[93] Vishay Semiconductors Document No. 82513, Rev. 1.4, December 05, "*Infrared Data Communication According the IrDA® Standard*". www.vishay.com/docs/82513/82513.pdf. 06/04/07

[94] M. D. Audeh and J. M. Kahn, "Performance evaluation of *L*-pulse-position modulation on non-directed indoor infrared channels," in *Proceedings of IEEE International Conference on Communications*, New Orleans, U.S.A., 1-5, vol. 2, pp. 660-664, May 1994.

[95] J. Garrido-Balsells, A. Garcia-Zambrana and A. Puerta-Notario, "Variable weight MPPM technique for rate-adaptive optical wireless communications," *Electronics Letters*, vol. 42, no. 1, 2006.

[96] R. Cryan and M. Menon, "Spectral characterisation of n^k pulse position modulation format," *Electronics Letters*, vol. 41, no. 23, 2005.

[97] Y. Zeng, R. Green and M. Leeson, "Adaptive pulse amplitude and position modulation for optical wireless channels," *2nd IET International Conference on Access Technologies*, pp. 13-16, 2006.

[98] Z. Xu, J. Wang and L. Shen, "A novel soft-decision decoding technology in diffuse optical wireless networks employing DH-PIM," *Proceedings of SPIE*, vol. 5640, pp. 504-512, 2005.

[99] Y. Liao, C. Zhang and X. Lin, "T-ZCZ keying for infrared wireless communications," *Proceedings of SPIE*, vol. 5640, pp. 524-532, 2005.

[100] H. Park and J. Barry, "Trellis-coded multiple-pulse-position modulation for wireless infrared communication," *IEEE Transactions on Communications*, vol. 52, no. 4, pp. 643-651, 2004.

[101] T. Satoh and T. Hashimoto, "Performance evaluation of chip-synchronous, code-asynchronous optical CDMA systems for indoor infrared wireless communications with diffuse link," *Electronics and Communications in Japan*, vol. 86, n0. 4, pp. 789-801, 2003.

[102] F. Delgado, R. Pérez Jiménez and O. González, "FHSS Optical Wireless transceiver for short range low-speed indoor sensor interconnection," *Microwave and Optical Technology Letters*, vol. 48, no. 11, 2006.

[103] V. Vitsa and A. Boucouvalas, "Packet level acknowledgement and go-back-n protocol performance in infrared wireless LANs," *International Journal of Communication Systems*, vol. 16, no. 2, pp. 171-191, 2003.

[104] M. Menon and R Cryan, "Optical wireless communications utilizing a dicode PPM PIN-BJT receiver," *Microwave and Optical Technology Letters*, vol. 45, no. 4, 2005.

[105] V. Vitsas and A. Boucouvalas, "Performance analysis of the advanced Infrared (Air) CSMA/CA MAC protocol for wireless LANs," *Wireless Networks*, vol. 9, no. 5, pp. 495-507, 2003.

[106] J. Rabadán, M. Bacallado, F. Delgado, S. Perez, and R. Pérez-Jiménez, "Experimental characterisation of a direct-sequence spread-spectrum optical wireless system based on pulse-conformation techniques for in-house communications," *IEEE transactions on Consumer Electronics*, vol. 50, no. 2, pp. 484-490, 2004.

[107] H. P. Hsu, *Applied Fourier Analysis*, Publisher: Harcourt Brace Jovanovich, U.S.A., 1984. ISBN: 0156016095.

[108] E. Ifeachor and B. Jervis, *Digital Signal Processing – A Practical Approach*, Publisher: Addison Wesley Longman, U.K, 1998. ISBN: 9780201544138.

[109] R. Polikar, "The Engineers Ultimate Guide To Wavelet Analysis," http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html, 10/05/07

[110] www.mathworks.com/access/helpdesk/help/toolbox/wavelet, 04/05/07

[111] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Pattern Anal. and Machine Intell.*, vol. 11, no. 7, pp. 674-693, 1989.

[112] B. Hubbard, *The World According to Wavelets: The Story of a Mathematical Technique in the Making*, Publisher: A K Peters, U.S.A, May 1998. ISBN: 1568810725.

[113] T. Keitt and J. Fischer, "Detection of scale-specific community dynamics using wavelets," *Ecology*, vol. 8, iss. 11, pp. 2895-2905, 2006.

[114] F. In and S. Kim, "The hedge ratio and the empirical relationship between the stock and futures markets: a new approach using wavelet analysis," *The Journal of Business*, vol. 79, iss. 2, pp. 799-821, 2006.

[115] J. Issartel, L. Marin, P. Gaillot and T. Bardainne, “A practical guide to time-frequency analysis in the study of human motor behavior: the contribution of wavelet transform,” *Journal of Motor Behavior*, vol. 38, iss. 2, pp. 139-160, 2006.

[116] A. Murata, “An attempt to evaluate mental workload using wavelet transform of EEG,” *Human Factors*, vol.47, iss. 3, pp. 498-509, 2005.

[117] M. Jiang and D. Crookes, “Area-efficient high-speed 3D DWT processor architecture,” *Electronics Letters*, vol. 43, p. 502, 2007.

[118] S. Smorfa and M. Olivieri, “HW-SW optimisation of JPEG2000 wavelet transform for dedicated multimedia processor architectures,” *IET Comput. Digit. Tech.*, vol. 1, p. 137, 2007.

[119] M. Florkowski and B. Florkowska , “Wavelet-based partial discharge image denoising,” *IET Gener. Transm. Distrib.*, vol. 1, p. 340, 2007.

[120] Y. Zhang, J. Jiang and F. Feroze, “Wavelet-based denoising for improving nonlinear dynamic analysis of pathological voices,” *Electronics Letters*, vol. 41, no. 16, pp. 930-931, 2005.

[121] www.c3.lanl.gov/~brislawn/FBI/FBI.html. 10/05/07.

[122] http://www.jpeg.org/jpeg2000/index.html. 10/05/07.

[123] S. Yea and W. Pearlman, “A Wavelet-Based Two-Stage Near-Lossless Coder,” *IEEE Transactions On Image Processing*, Vol. 15, No. 11, pp. 3488-3500, 2006.

[124] M. Akho-Zahieh and O. Ugweje, “Wavelet packet based MC/MCD-CDMA communication system,” *Electronics Letters*, Vol. 42, No. 11, pp. 644-645, 2006

[125] Y. Xiangbin, X. Zhang and B. Guangguo, “Performance analysis of multicarrier CDMA system based on complex wavelet packet and space-time block codes in rayleigh fading channel,” *Journal of Circuits, Systems and Computers*, vol. 15, iss. 1, pp. 57-74, 2006.

[126] O. Kucur and G. Atkin, “Performance of Haar wavelet based scale-code division multiple access (HW/S-CDMA) using decorrelating multi-user detector,” *Computers and Electrical Engineering*, vol. 31, iss. 7,pp. 468-484, 2005.

[127] Y. Xiangbin, X. Dazhuan, X. Weiye and B. Guangguo, “Uplink Performance of complex wavelet packet based Multi-carrier CDMA system with space diversity combining technique,” *Proceedings, MAPE: IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, vol. 1, pp. 688-691, 2005.

[128] X. Yu, X. Zhang, D. Xu, and G. Bi, “Performance analysis of multiband complex wavelet based MC-CDMA system with space diversity combining in Rayleigh fading channel,” *Wireless Personal Communications*, vol. 41, iss. 2, pp.193-206, 2007.

[129] O. Kucur, E. Ozturk and G. Atkin, "Bit error rate performance of Haar wavelet based scale-code division multiple access (HW/S-CDMA) over the asynchronous AWGN channel," *International Journal of Communication Systems*, vol. 20, iss. 4, pp. 507-514, 2007.

[130] A. Jamin and P. Mahonen,"Wavelet packet modoulation for wireless communications," *Wireless Communications and Mobile Computing*, pp. 123-137, 2004.

[131] S. Lee and M. Kavehrad, "Airborne laser communications using wavelet packet modulation and its performance enhancement by equalization," *Proceedings of SPIE - The International Society for Optical Engineering, Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense V*, vol. 6201, p. 62011U, 2006.

[132] X. Tang, X. Huang and H. Yue, "A direct sequence spread wavelet modulation method," *IEEE 005 International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, vol. 2, 2005.

[133] R. Ren, S. Zhu and Y. Luo, "Multi-carrier modulation and multi-scaling distinguishable system based on adaptive orthogonal wavelet packet," *International Conference on Signal Processing and Communications, SPCOM*, pp. 244-249, 2004.

[134] H. Zhang, D. Yuan and F. Zhao, "Research of wavelet based multicarrier modulation system with near shannon limited codes," *Chinese Journal of Electronics*, vol. 14, iss. 3, pp. 430-433, 2005.

[135] A. Jamin and P. Mahonen, "Wavelet packet modulation for wireless communications," *Wireless Communications and Mobile Computing*, vol. 5, iss. 2, pp. 123-137, 2005.

[136] E. Okamoto, Y. Iwanami and T. Ikegami, "Application of wavelet packet modulation to mobile communication," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E87-A, iss. 10, pp. 2684-2691, 2004.

[137] D. Cariolaro and L. Favalli, "Recovery of ISI Channels using Multiresolution Wavelet Equalisation," *IEEE International Conference on Communications*, New York, USA, pp. 74-78, 2002.

[138] F. Liu, J. Cheng, J. Xu and X. Wang, "Wavelet Based Adaptive Equalisation Algorithm," *Global Telecommunications Conference*, Pheonix, Arizona, pp. 1230-1234, 1997.

[139] C. Lin, C. Shih and P. Chen, "Nonlinear time-varying channel equalizer using self-organizing wavelet neural networks," *IEEE International Joint Conference on Neural Networks – Proceedings*, vol. 3, pp. 2089-2094, 2004.

[140] A. Pradhan, S. Meher and A. Routray, "Communication channel equalization using wavelet network," *Digital Signal Processing*, vol. 16, iss. 4, pp. 445-452, 2006.

[150] M. Jiang, B. Deng and G. Gielen, "Wavelet neural networks for adaptive equalization by using the orthogonal least square algorithm," *Tsinghua Science and Technology*, vol. 9, iss. 1, pp. 24-29, 2004.

[151] J. Wang, W. Zhang and G. Song, "Nonlinear channel equalizer based on wavelet packet transform," *Shuju Caiji Yu Chuli/Journal of Data Acquisition and Processing*, vol. 21, iss. 1, pp. 34-36, 2006.

[152] S. Haykin, J. Nie and B. Currie, "Neural network-based receiver for wireless communications," *Electronics Letters*, vol. 35, no. 3, pp. 203-205, 1999

[153] J. Chapa and R. Rao, "Algorithms for designing wavelets to match a specified signal," *IEEE Transactions on Signal Processing*, vol.48, no. 12, pp. 3395-3406, 2000.

[154] www.wavelet.org. 10/05/07.

[155] www.sdltd.com. 10/05/07.

[156] www.bittware.com. 10/05/07.

[157] www.mathworks.co.uk/access/ helpdesk/help/toolbox/nnet/nnet.shtml

[158] S. Haykin, "Neural Networks, A Comprehensive Foundation," Prentice Hall, U.S.A., 1999, ISBN 0-13-908385-5.

[159] Sung-Hyun Heo, Sung-Kwon Park, Sang-Won Nam, "Channel Equalization for Severe Intersymbol Interference and Nonlinearity with a Radial Basis Function Neural Network," *IEEE Proceedings International Joint Conference On Neural Networks*, Piscataway USA 1999 Vol 6 pp 3992-3995

[160] M. Powell, "Restart procedures for the conjugate gradient method," *Mathematical Programming*, vol. 12, pp. 241-254, 1977.

[161] www.mathworks.com/access/helpdesk/help/toolbox/signal. 15/06/2007

[162] B. Sheu, D. Chen, "1D Compact Neural Networks for Wireless Communication and Mobile Computing," *IEEE Proceedings of the International Symposium on Circuits and Systems*, Atlanta USA, Vol 3, pp 551-554, May 1996

[163] L. Chengliang, and Z. Weiyu, "Subspace separator and functional link neural network based receiver for DS-CDMA signal," *Proceedings, 8th International Conference Advanced Communication Technology*, Vol. 1, pp. 782-785, 2006

[164] M. Ibnkahla and J. Yuan , "A neural network MLSE receiver based on natural gradient descent: Application to satellite communications," *Eurasip Journal on Applied Signal Processing, Nonlinear Signal and Image Processing-Part II*, Vol. 2004, Iss.16, pp. 2580-259, 2004.

[165] E. Soujeri and H. Bilgekul, "Multiuser detection of synchronous MC-CDMA in multipath fading channels using hopfield neural networks," *Neural Processing Letters*, Vol. 18, Iss. 1, pp. 49-63, 2003

[166] Y. Sha, B. Wang and G. Wang, “Simulation on performance of rake receiver with neural network-based combiner,” *Dongbei Daxue Xuebao/Journal of Northeastern University*, Vol. 28, Iss. 2, pp. 197-200, 2007

[167] K. Raghavendra and A. Tripathy “An efficient channel equalizer using artificial neural networks,” *Neural Network World*, vol. 16, iss. 4, pp. 357-368 2006.

[168] J. Choi, M. Bouchard and T. Yeap, “Recurrent neural equalization for communication channels in impulsive noise environments,” *Proceedings of the International Joint Conference on Neural Networks*, vol. 5, pp. 3232-3237, 2005.

[169] C. Lee, J. Go and B. Baek, “Feature extraction for neural network equalizers trained with multi-gradient,” *IEEE International Conference on Neural Networks - Conference Proceedings*, vol. 2, pp. 1575-1578, 2004.

[170] P. Heng, W. Cao and P. Qiu, “Decision feedback equalizers based on two weighted neural network,” *Proceedings of International Conference on Machine Learning and Cybernetics*, vol.5, pp. 3125-3156, 2004.

[171] K. Samaras, A. M. Street, D. C. O'Brien and D. J. Edwards, “Methods for evaluating error rates in infrared wireless links,” *Electronics Letters*, vol. 33, no. 20, pp. 1720-1721, September 1997.

[172] A. J. C. Moreira, A. M. R. Tavares, R. T. Valadas and A. M. de Oliveira Duarte, “Modulation methods for wireless infrared transmission systems: performance under ambient light noise and interference,” in *Proceedings of SPIE Conference on Wireless Data Transmission*, Philadelphia, U.S.A., vol. 2601, pp. 226-237, October 1995

[173] F. Stremler, *Communication Systems*, 3rd Edition, Addison-Wesley, U.S.A., 1990.

[174] The Open University, *Digital Telecommunications*, Block 6, Coding and Modulation. Open University Press, U.K., 1990.

[175] S. Karni, *Applied Circuit Analysis*. John Wiley & Sons, U.S.A., 1988.

[176] I Otung, *Communication Engineering Principles*, Palgrave, U.K., 2001.

[177] A. Burr, *Modulation and Coding for Wireless Communications*, Prentice-Hall, U.K., 2001.

[178] J. Proakis and M. Salehi, *Contemporary Communication Systems Using Matlab*, PWS Publishing Company U.S.A., 1998.

[179] Koorosh Akhavan, Mohsen Kavehrad and Svetla Jivkova, “High-Speed Power-Efficient Indoor Wireless Infrared Communications Using Code Combining – Part I,” *IEEE Transactions On Communications*, vol. 50, no. 7, July 2002.

[180] K. Akhavan, M. Kavehrad and S. Jivkova, "High-Speed Power-Efficient Indoor Wireless Infrared Communications Using Code Combining – Part II," *IEEE Transactions On Communications*, vol. 50, no. 9, September 2002.

[181] R. Stewart, *DSPededia Part 1*, Entegra Ltd, U.K., 1999