Adaptive Logic Network Correlation Techniques for Optical Code Division Multiple Access Systems / 1994

M. J. Parham BEng.

Thesis submitted for the degree of Doctor of Philosophy in the Department of Electronic and Electrical Engineering at the University of Surrey.

April 1994

© M. J. Parham 1994
Abstract

Code Division Multiple Access (CDMA) techniques afford Local Area Networks (LANs) the support of concurrent, asynchronous communication between users without access delay. These properties are obtained by encoding users' data with high rate code sequences, so that data is spread over a much larger bandwidth than would usually be required for transmission. The necessary bandwidth is provided by using optical fibre both as the LAN medium and for incoherent optical signal processing.

Conventionally, extraction of a desired user’s signal is achieved by correlation using a single delay-line matched filter. Matched filters are optimal for the recovery of a known signal in the presence of additive noise. However, in a CDMA environment, their performance is limited by Multiple Access Interference (MAI), arising from the cross-correlation products of overlaid users, and degrades as the number of users increases.

Adaptive Logic Networks (ALNs), a form of Artificial Neural System (ANS), are applied to the extraction of a single user’s signal in a multi-user environment. In the approach taken, ALNs learn to incorporate the presence of interfering users’ signals, in deciding the actual data bit received. Computer simulation is used to compare the error rates obtained by ALNs and the previously proposed correlation receivers; the performance of the latter providing a benchmark. Simulations are conducted assuming chip synchronism between users and no external sources of noise, i.e. MAI is assumed dominant. Consideration is given to systems employing both sparse optical codes and Gold-like codes as spreading sequences.

In all the systems considered, ALNs are shown to enable significant reductions in error rate over the conventional correlation receivers. MAI effects, causing errors with the correlation receivers, are reduced by using additional temporal and intensity based information contained in the receiver input signal. This permits an ALN to extract details of the structure of interfering users’ signals, to provide a better context for the classification of the desired user’s signal. In the systems employing sparse codes, it is demonstrated that while a certain amount of MAI persists, it may be minimised by selection of the ALN input window, to provide the maximum possible information regarding the interfering users’ signals. In the systems using Gold-like codes, it is shown that ALNs can be used to completely eliminate the effects of MAI. This is significant since, although this form of code sequence is suited to coherent CDMA systems, the cross-correlation products arising in incoherent optical environments are normally considered to be unacceptably high.
Acknowledgements

This work was conducted with financial support from the Science and Engineering Research Council. I would like to thank my supervisors, Dr B. L. Weiss of the Department of Electronic and Electrical Engineering, University of Surrey, and Dr C. Smythe of the Department of Computer Science, University of Sheffield, for their advice, constructive criticism and encouragement throughout the duration of this project. I would also like to thank Dr A. D. Keedwell of the Department of Mathematical and Computing Sciences, University of Surrey, for assistance with the theory of optical orthogonal codes.
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACF</td>
<td>Auto-Correlation Function</td>
</tr>
<tr>
<td>ALN</td>
<td>Adaptive Logic Network</td>
</tr>
<tr>
<td>ANS</td>
<td>Artificial Neural System</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate</td>
</tr>
<tr>
<td>BPSK</td>
<td>Binary Phase Shift Keying</td>
</tr>
<tr>
<td>CCF</td>
<td>Cross-Correlation Function</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>CSMA / CD</td>
<td>Carrier Sense Multiple Access with Collision Detection</td>
</tr>
<tr>
<td>DS</td>
<td>Direct Sequence</td>
</tr>
<tr>
<td>FDDI</td>
<td>Fibre Distributed Data Interface</td>
</tr>
<tr>
<td>FDM</td>
<td>Frequency Division Multiplexing</td>
</tr>
<tr>
<td>FDMA</td>
<td>Frequency Division Multiple Access</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Correction</td>
</tr>
<tr>
<td>FH</td>
<td>Frequency Hopping</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GP</td>
<td>Genetic Programming</td>
</tr>
<tr>
<td>IOD</td>
<td>Index Of Discrimination</td>
</tr>
<tr>
<td>ISLN</td>
<td>Integrated Services Local Network</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MAI</td>
<td>Multiple Access Interference</td>
</tr>
<tr>
<td>MLE</td>
<td>Majority Logic Encoding</td>
</tr>
<tr>
<td>OOC</td>
<td>Optical Orthogonal Code</td>
</tr>
<tr>
<td>OOK</td>
<td>On-Off Keying</td>
</tr>
<tr>
<td>PN</td>
<td>Pseudo-Noise</td>
</tr>
<tr>
<td>PSK</td>
<td>Phase Shift Keying</td>
</tr>
<tr>
<td>RC-CDMA</td>
<td>Random Carrier – Code Division Multiple Access</td>
</tr>
<tr>
<td>SIK</td>
<td>Sequence Inverse Keying</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>SS</td>
<td>Spread Spectrum</td>
</tr>
<tr>
<td>SSMA</td>
<td>Spread Spectrum Multiple Access</td>
</tr>
<tr>
<td>TDM</td>
<td>Time Division Multiplexing</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>TH</td>
<td>Time Hopping</td>
</tr>
<tr>
<td>WDM</td>
<td>Wavelength Division Multiplexing</td>
</tr>
<tr>
<td>WDMA</td>
<td>Wavelength Division Multiple Access</td>
</tr>
</tbody>
</table>
Contents

Chapter 1 Introduction ........................................................................................................... 1
  1.1 Background ................................................................................................................ 1
  1.2 Statement of Intent .................................................................................................... 2
  1.3 Thesis Structure ........................................................................................................ 3
  1.4 Review of Previous Work ....................................................................................... 4

Chapter 2 Theory .................................................................................................................. 10
  2.1 Introduction ............................................................................................................... 10
  2.2 CDMA Principles .................................................................................................... 10
  2.3 CDMA Code Sequence Design ............................................................................. 12
    2.3.1 Conventional CDMA Sequences .................................................................. 13
    2.3.2 Sparse Codes for Optical CDMA Systems ............................................. 15
  2.4 Optical CDMA Networks ....................................................................................... 19
    2.4.1 System Description ..................................................................................... 19
    2.4.2 Simple Correlator ....................................................................................... 20
    2.4.3 Hard-limited Correlator ............................................................................. 22
  2.5 Adaptive Logic Networks ....................................................................................... 23
    2.5.1 Structure ........................................................................................................ 23
    2.5.2 The ALN Learning Algorithm .................................................................. 25
      Initialisation ........................................................................................................ 25
      The Concept of Responsibility ...................................................................... 25
      Adaptation ......................................................................................................... 27
  2.6 Adaptive Logic Networks as Optical CDMA Receivers ...................................... 27
  2.7 Summary .................................................................................................................. 30

Chapter 3 Simulation .......................................................................................................... 31
  3.1 Introduction ............................................................................................................... 31
  3.2 Environment ............................................................................................................. 31
    3.2.1 The Channel Model .................................................................................... 31
    3.2.2 Receiver Input Signal Generation ............................................................ 32
    3.2.3 Code Sequence Allocation ...................................................................... 35
  3.3 ALN Training ............................................................................................................ 36
    3.3.1 ALN Input Encoding ................................................................................... 36
      Explicit Encoding Techniques ....................................................................... 37
      Hard-limited Input ............................................................................................ 40
    3.3.2 Training Procedure ..................................................................................... 40
## Chapter 4  ALN Receivers in CDMA Systems employing Optical Orthogonal Codes

4.1 Introduction ................................................................. 45
4.2 The Simple Correlator .............................................. 45
   4.2.1 Configuration ................................................. 45
   4.2.2 Simulation Results ........................................... 46
   4.2.3 Error Sequence Analysis .................................... 47
   4.2.4 Extended Error Analysis .................................... 48
4.3 The Hard-limited Correlator ........................................ 50
   4.3.1 Configuration .................................................. 50
   4.3.2 Simulation Results ........................................... 50
   4.3.3 Error Sequence Analysis .................................... 51
   4.3.4 Extended Error Analysis .................................... 52
4.4 ALN Receivers ............................................................ 53
   4.4.1 ALN Configuration ........................................... 53
   4.4.2 ALN Simulation Results ..................................... 55
   4.4.3 ALN Training Parameters .................................. 55
   4.4.4 ALN Learning Behaviour .................................... 60
   4.4.5 Performance Assessment ................................. 67
4.5 Summary ................................................................. 69

## Chapter 5  Enhanced ALN Receivers in CDMA Systems employing Optical Orthogonal Codes via Code Shift Selection

5.1 Introduction ................................................................. 71
5.2 The Simple Correlator .............................................. 71
   5.2.1 Configuration .................................................. 71
   5.2.2 Simulation Results ........................................... 72
   5.2.3 Error Sequence Analysis .................................... 72
   5.2.4 Extended Error Analysis .................................... 72
5.3 The Hard-limited Correlator ........................................ 74
   5.3.1 Configuration .................................................. 74
   5.3.2 Simulation Results ........................................... 74
   5.3.3 Error Sequence Analysis .................................... 74
5.3.4 Extended Error Analysis ..................................................... 75

5.4 ALN Receivers ................................................................................. 76
  5.4.1 Configuration ................................................................... 76
  5.4.2 ALN Simulation Results ..................................................... 77
  5.4.3 ALN Training Parameters ................................................... 79
  5.4.4 ALN Learning Behaviour ................................................ 82
  5.4.5 Performance Assessment ................................................ 89

5.5 Summary ........................................................................................... 91

Chapter 6 ALN Receivers in Optical CDMA Systems employing
  Gold-like Codes ...................................................................................... 93
  6.1 Introduction .............................................................................. 93
  6.2 The Simple Correlator ............................................................. 94
    6.2.1 Configuration ................................................................. 94
    6.2.2 Simulation Results ......................................................... 94
    6.2.3 Error Sequence Analysis .............................................. 94
    6.2.4 Extended Error Analysis .............................................. 97
  6.3 The Hard-limited Correlator ..................................................... 98
    6.3.1 Configuration ................................................................. 98
    6.3.2 Simulation Results ......................................................... 98
    6.3.3 Error Sequence Analysis .............................................. 99
    6.3.4 Extended Error Analysis .............................................. 100
  6.4 ALN Receivers ............................................................................ 100
    6.4.1 Configuration ................................................................. 100
    6.4.2 ALN Simulation Results ................................................. 101
    6.4.3 ALN Training Parameters .............................................. 101
    6.4.4 ALN Learning Behaviour .............................................. 105
    6.4.5 Performance Assessment .............................................. 111
  6.5 Summary ...................................................................................... 112

Chapter 7 Conclusions ........................................................................ 114
  7.1 Introduction .............................................................................. 114
  7.2 Summary of Results .................................................................. 115
  7.3 Conclusions ............................................................................... 118
  7.4 Further Work ............................................................................ 122

References ......................................................................................... 124
Chapter 1   Introduction

1.1 Background

The use of optical fibre as the physical medium in a Local Area Network (LAN) has the potential to provide a communications bandwidth in the range of 1-100 THz. However, currently most network signal processing functions are performed electronically and therefore optical-to-electrical and electrical-to-optical conversions are required. This leads to the creation of processing bottlenecks since the speed of the system is limited by the hardware at these interfaces. As such, the amount of accessible bandwidth is limited to a fraction of that available; typically ~ 1GHz. In order to make better use of the fibre medium attention must be devoted to both the development of media access schemes better suited to the optical environment and to optical processing methods.

Media access schemes are concerned with allocating traffic demand efficiently over the available network bandwidth. In synchronous multiplexing techniques, such as Time-, Frequency- and Wavelength-Division Multiplexing (TDM, FDM, and WDM), domain slots are permanently allocated to connections. Such schemes are wasteful of resources whenever a user is idle and are thus unsuitable when the needs of users are unpredictable. Nevertheless, very high speed optical fibre networks have been proposed using these methods [Prucnal, 1986a][Acampora, 1989][Acampora, 1990][Champtac, 1990]. WDM schemes have received particular attention [Hill, 1990][Kaminow, 1989][Linke, 1989], since node hardware need only operate at electronic speeds and higher throughputs are achieved by concurrent transmission at different wavelengths.

Asynchronous multiplexing schemes allow users to share the communications bandwidth among themselves [Tobagi, 1980][Tanenbaum, 1988][Skov, 1989][Rubin, 1990]. For LANs, the methods used include contention based techniques such as Carrier Sense Multiple Access with Collision Detection (CSMA/CD)(IEEE 802.3) and token passing schemes such as the token ring (IEEE 802.5). CSMA/CD is efficient when only a few network users have data to transmit and traffic is bursty, i.e. transmissions are short and sporadic. However, performance degrades at higher loads, since as the number of users increases, so does both the frequency of collisions and the likelihood of the channel being busy, preventing transmission. The token ring is more efficient at high loads when many stations require the ability to transmit continuously for extended periods of time (stream traffic). However, both CSMA/CD and token ring suffer from cumulative delay as traffic increases since the entire bandwidth is dedicated to a single user at any one time.
The strong dependency of CSMA/CD performance on the ratio of packet transmission time to round trip propagation delay makes the transfer of this technique to optical systems inappropriate [Acampora, 1989]. At the high transmission rates being considered for optical networks the minimum packet length required becomes unacceptable. Token ring techniques have been transferred to optical fibre networks in the form of the Fibre Distributed Data Interface (FDDI). FDDI is designed to operate at high speed (100 Mb/s) and to support a mix of both stream and bursty traffic. In principle, signals may propagate concurrently around the ring since the token is released after packet insertion. However, at any time, only one station may hold the token that allows it to transmit. Therefore, capacity is limited to that of a single point-to-point link and so only a fraction of the inherent optical fibre bandwidth is used.

Code Division Multiple Access (CDMA) represents an alternative to the schemes outlined above. In contrast to other LAN Medium Access Control (MAC) protocols, it permits a destination station to capture a packet rather than the entire communications bandwidth. Indeed, it allows concurrent access to a number of network users without access delay and allocates bandwidth as and when it is needed. The multiple access capability is achieved via the assignment of code sequences with orthogonal-like properties to users of the system and this results in transmitted data being spread over a much larger bandwidth than normally required for transmission. As such, CDMA is particularly suited for use in optical fibre networks where bandwidth need no longer be a limited resource.

### 1.2 Statement of Intent

In order for CDMA to be viable for optical networks, the available bandwidth resource must be made accessible. This requires optical processing elements to prevent unnecessary signal conversions and maintain optical network rates, hence eliminating processing bottlenecks. To date, the type of optical signal processing techniques considered for optical CDMA systems have focused on the use of optical delay-lines as matched filters [Jackson, 1985]. A typical optical CDMA receiver employs a matched filter whose impulse response is the time-reversed complex conjugate of the code sequence to be received. Input of the desired signal to the receiver results in a correlation operation with a maximum output at the synchronisation time. This auto-correlation peak may then be compared to a threshold to determine whether the data sent was a ‘1’ or a ‘0’.

Matched filtering techniques are known to be optimal for the recovery of a signal in the presence of Additive White Gaussian Noise (AWGN) [Turin, 1976]. However, in a
CDMA system, a single matched filter is no longer optimal. Cross-correlation products between the overlaid (interfering) users' signals and the desired reference signal generate Multiple Access Interference (MAI). The effects of MAI may be minimised by designing the users' code sequences to have low cross-correlation values, thus reducing the likelihood of the decision threshold being falsely achieved. However, the matched filter still only considers the desired sequence in deciding the data sent and no account is taken of the presence of other users' sequences on the channel. As such, system performance degrades as the number of active users increases.

This project examines the application of a particular kind of Artificial Neural System (ANS), Adaptive Logic Networks (ALNs) [Armstrong, 1991a], to enhance the reception in an optical CDMA network. Neural network techniques make no assumptions regarding the underlying statistical nature of signals which suggests the possibility of improved performance when interference cannot easily be modelled. It is intended that the ALNs should learn to decode signals based not only on knowledge of the desired user's code sequence but also on that of the other users' sequences thereby using this information to reduce the number of errors occurring due to MAI.

The aim of the work is to improve the efficiency of incoherent optical CDMA schemes by reducing the bit error rate for a given number of users compared to that previously obtained via correlation receivers. Explicit knowledge of the code sequences used is not provided directly to the ALNs, rather an ALN is expected to learn details of the various code sequences present on an optical CDMA channel based on examples of network signals and the desired outputs in such cases.

1.3 Thesis Structure

Having introduced the background, motivation and aims of the project, the remainder of this chapter is concerned with a review of the previous work carried out in this and related areas. In chapter 2 the theory and concepts behind the work are presented. This includes an overview of the principles of CDMA with particular emphasis on the differences between the more common copper or radio based implementations and the relatively recent optical incarnations. The theory of ALNs is also examined and the chapter is unified by a more detailed exposition of why they should be used in such systems. Chapter 3 provides the transition from the theoretical to the practical details of the project. A methodology is developed for modelling an optical CDMA system in order to permit simulation of the newly proposed receiver architectures. Simulation of the previously considered receivers is also covered in order to establish a benchmark to evaluate performance. Chapter 4 presents the results obtained for the first configuration
of ALN receivers considered in the context of an incoherent optical system employing recently developed optical codes specially designed for such systems. Performance is compared to the former reception methods and the effect of ALN training parameters on receiver error rate is examined. Chapter 5 continues the thrust of chapter 4 and looks at methods to improve performance still further by particular consideration of the ALN receiver configuration with respect to the desired optical code sequence. Chapter 6 investigates the possibility of employing Pseudo-Noise (PN) sequences, traditionally used in conventional CDMA systems, in incoherent optical systems and reports results for ALN receivers in this environment. The results obtained using the two different coding schemes are then compared. Finally, chapter 7 presents a summary of the results, conclusions obtained from a discussion of these results and suggests areas for further investigation.

1.4 Review of Previous Work

The review of previous proposals for CDMA LANs will concentrate on optical fibre based systems, however efforts of researchers to apply these techniques to cable based systems are also acknowledged [Elhakeem, 1983][Smythe, 1985a][Simcock, 1987][Lee, 1987]. One of the first attempts to apply CDMA techniques to an optical fibre LAN was based on the use of a pseudorandom sequence generator to determine the position in a time frame of a pulse representing a user’s desired data ‘1’ [Shaar, 1983a]. In an attempt to reduce interference between users and thus improve error rate, prime sequences were developed with the aim of minimising coincidences between time mapped pulses [Shaar, 1983b]. This appears to be the first instance of codes being specifically designed for optical CDMA systems.

Many of the proposals for optical CDMA LANs involve the transfer of copper- or radio-based techniques, using conventional PN sequences, to the optical domain. Typically, processing in these systems is performed electronically with two main consequences. Firstly, channel sequences are constrained to be unipolar since optical to electronic conversion is carried out by square law devices and secondly, as discussed previously, accessible fibre bandwidth is limited. One such system has proposed the use of Gold codes to electronically encode user data which is then optically multiplexed on to a fibre-based network after conversion [Tamura, 1985]. A scheme based on an active star network topology has also been reported where input signals are added in equal proportion at the hub to form a multi-level composite sequence [Marshall, 1989]; the active star can be used to eliminate near-far effects where transmitters local to a receiver, swamp reception of transmissions from distant stations. The multi-level signal is then fed to hard-limiting correlation receivers. In order to obviate the need for bipolar signalling it
has been suggested that the receiver hard-limiter is removed to the hub, allowing transmitters to use binary on-off signalling [Marshall, 1990]. Majority Logic Encoding (MLE) is used to combine users’ codes via binary addition, the output to receivers being a binary vote between input streams.

Another system based on conventional codes and electronic processing has proposed the use of optical Sequence Inverse Keying (SIK) [O’Farrell, 1989a]. Depending on whether a data ‘0’ or ‘1’ is required, either a unipolar sequence or its complement is transmitted. After conversion, the unipolar sequence is correlated electronically with a bipolar reference at the receiver. Under these circumstances auto- and cross-correlation functions maintain the same relative values as those in bipolar baseband CDMA. Indeed, O’Farrell has shown that, given the dominance of MAI due to other users, the Signal-to-Noise Ratio (SNR) achievable in both optical and bipolar baseband CDMA systems is the same [O’Farrell, 1989a].

In view of the limitations on bandwidth availability imposed by electronic processing, research has also been carried out into the design of CDMA codes with very low cross-correlation values and giving rapid synchronisation times when used with optimal matched filters designed to extract a desired code from those of other users rather than AWGN [O’Farrell, 1989b][O’Farrell, 1991]. The aim is to provide an efficient CDMA channel which may be used in conjunction with other media access protocols such as CSMA or WDM to improve throughput or delay characteristics [O’Farrell, 1989a].

Other hybrid systems have also been reported in the literature, the intention being to overcome current limitations in optical technology by combining the properties of separate multiple access schemes to advantage. Coherent systems based on electro-optic modulation of a laser source suffer from laser frequency drift, difficulty in determining laser centre frequency at manufacture and electronic hardware speed limitations. This makes the realisation of pure Frequency-, Wavelength- or Time-Division Multiple Access (FDMA, WDMA or TDMA) systems difficult. To this end a Random Carrier - CDMA (RC-CDMA) system has been proposed [Foschini, 1988]. The aim is to spread signals so that the carriers can be randomly placed over the optical spectrum using the properties of CDMA to provide a degree of interference immunity. With the advent of more stable lasers possessing limited tunability, this overly pessimistic approach was refined to a hybrid FDMA-CDMA system [Vannucci, 1989a]. The ability to perform limited tuning allows the amount of spreading used to be reduced whilst maintaining some robustness to interference and laser frequency drift. A scheme has been suggested to allow users of two different bit rates to co-exist on the same network. A coherent system based on optical spreading of a laser by an electro-optic modulator and an electronically generated PN sequence was proposed for high bit rate users [Vannucci, 1989b], while direct laser
modulation and sparse encoding techniques were suggested for the support of low bit rate users [Vannuci, 1989a]. The use of Forward Error Correction (FEC) codes was advocated in both cases to afford greater interference protection than spectral spreading alone. As in the case of the other systems described, the use of electronic code generation limits the degree of spreading available. Furthermore, problems are anticipated in achieving synchronisation in the coherent high bit rate system.

In order to overcome the limitations of electronic processing, several systems have been proposed using all-optical encoding and decoding techniques. These methods can be classified according to whether the processing is of a coherent or incoherent nature. One coherent technique suggested is based on the encoding and decoding of femtosecond light pulses [Salehi, 1989a]. In this approach, these ultra-short light pulses are spatially decomposed using a diffraction grating and lens arrangement and then projected onto a pseudorandomly spatially patterned phase mask whose construction is based on PN sequences, such as m-sequences or Gold codes. The light is then recombined into a single beam via another lens and grating. The emerging light pulse is a longer duration, lower intensity pseudo-noise burst. Decoding is achieved at the receiver using a similar arrangement whose phase mask is the complex conjugate of that in the encoder. The light pulse is reconstructed and any interfering pulses remain spread. Similar techniques where all-optical processing is performed in the transform domain have also been reported [DeCusatis, 1990].

Coherent systems have also been reported which use fibre-optic ladder networks to perform the encoding and decoding of data [Marhic 1989][Sampson, 1990a and b]. Ladder networks consist of fibre pairs coupled together at various points. Feeding a single laser pulse into the network generates a pulse train according to the number of couplings. Data may be impressed by On-Off Keying (OOK) of the laser, resulting in a coherent correlation at the receiver, or Phase Shift Keying (PSK) of half the generated pulses may be used to create anti-correlation nulls. At the receiver, the pulse train enters an identical ladder network via the adjacent port to that used for generation. Detection is also effected at the adjacent exit port via a thresholding operation. A coherent correlation is achieved since the series of pulses is phase coherent, all of them being generated by a single original pulse.

All-optical CDMA systems based on incoherent processing techniques have typically employed optical fibre delay-line matched filters to perform the correlation operation necessary for signal recovery. In incoherent systems, both transmitted sequences and delay-line taps are constrained to be of a unipolar nature. Conventional PN sequences, which rely on a bipolar representation to achieve their desirable correlation properties, do not perform well in this environment. It has been shown that the high number of ‘1’s in
this form of sequence gives rise to unacceptably high cross-correlation values for low numbers of users [Prucnal, 1986b]. This has led to the consideration of new data encoding methods. One proposal has been to assign patterns to users which are transmitted as pulses distributed over a number of fibres [Hui, 1985]. As such, correlator taps are taken from the relevant fibre positioned according to the pulses in the pattern.

Asynchronous CDMA LANs based on single optical fibre channels have led to the development of sparse coding techniques with correlation properties better suited to the unipolar environment. Most work in this area has concentrated on systems using either prime codes [Shaar, 1983b][Prucnal, 1986b] or Optical Orthogonal Codes (OOCs) [Chung, 1989][Salehi, 1987a][Salehi, 1989a], although other encoding techniques have also been reported [Mendez, 1989][Vethanayagam, 1991][Holmes, 1992]. Systems using both prime codes and OOCs are similar in that they are based on star topologies and use OOK, sending a sequence for a data ‘1’ and no sequence for a data ‘0’. Encoding may be performed electronically via direct laser modulation or optically by delay-lines. Detection is achieved by thresholding the output of the appropriate delay-line matched filter. It has been estimated that, for a given SNR, systems based on prime codes can support a comparable number of users to those employing Gold codes and conventional electronic processing [Prucnal, 1986b]. The advantage of such systems is that by using optical processing many more chips per bit may be used, allowing the support of more users.

Synchronous optical CDMA schemes based on prime codes have also been proposed. Such systems are able to use shifted versions of codewords and hence support more users, since although some auto-correlation and cross-correlation peaks will now be of equal magnitude, receivers are synchronised to the expected version of their own auto-correlation peak. Performance of synchronous systems have been shown to be better than asynchronous systems, in that more users may be supported for a given bit error rate [Kwong, 1991]. Hybrid schemes combining synchronous CDMA and TDMA on a single channel have also been investigated. Channels are allocated as time slots; high bit rate users use TDMA in these slots, whereas low bit rate users access the slots via CDMA techniques [Kwong, 1990]. This concept has been further developed to that of an Integrated-Services Local Network (ISLN) [Santoro, 1989]. Recently optical CDMA systems employing sparse coding have also been suggested for digital video multiplexing [Gagliardi, 1993].

The incoherent optical systems employing OOCs differ from those using prime codes in that they propose the use of non-linear optical processing elements to enable system performance to converge to that expected from calculations based on the code properties alone [Salehi, 1987b]. These devices are placed before the correlator where they perform a hard-limiting operation, clipping high optical intensities back to the level of a single
user. It has been demonstrated that this helps to exclude code interference patterns causing errors in a simple correlator by localised summation of optical intensity [Salehi, 1989a]. The structure of these receivers can be considered to be equivalent to an optical AND gate [Salehi, 1989a].

Neuromorphic (neural-like) networks based on a modification of the Hopfield model of associative memory [Hopfield, 1982] have also been proposed for all-optical signal processing applications in incoherent CDMA systems employing OOCs [Vecchi, 1988] [Salehi, 1989a]. The conventional Hopfield model stores bipolar memory vectors as sums of outer products in a connectivity matrix. Providing the number of stored memories is low, presentation to the matrix of an input vector results in convergence to the closest stored memory vector after a series of multiply and threshold cycles. In the neuromorphic scheme the bipolar memory vectors are replaced with the unipolar OOCs, which are again stored as sums of outer products in a clipped connectivity matrix. In this case convergence has been demonstrated after a single multiply and threshold cycle [Vecchi, 1988]. In the case of a CDMA receiver the neuromorphic network is required to act as a code filter and stores only the desired code sequence in the connectivity matrix. Input vectors are hard-limited before presentation to the connectivity matrix and the resulting output vector has its elements individually thresholded at the code weight. The desired vector is correctly recalled whenever a vector representing a data ‘1’ is presented. However, errors can occur when a data ‘0’ is presented if MAI mimics the data ‘1’ sequence.

Neural network receivers have also been investigated with a view to improving performance in CDMA systems employing Binary PSK (BPSK) with PN sequences (in bipolar format), such as would be used for radio-based links. The majority of research has focused on the less complicated case of synchronous CDMA where relative timeshifts between users are assumed to be zero and has involved the use of multi-layer perceptrons trained with either the standard backpropagation learning algorithm or various enhanced versions of it [Paris, 1988][Aazhang, 1990]. More recently, results have been reported that include extension to the asynchronous case; the relative delays between users being assumed unknown, but constant, during training [Aazhang, 1992].

The novel aspects of this work are in its application of a less well established neural network technique, ALNs, to enhance reception in an incoherent optical CDMA LAN using OOK. The previous application of neuromorphic networks to decoding in such systems is limited, since it is based only on the desired code sequence and takes no account of the presence of other users’ codes. Simulation results have shown that the error rates attained are equivalent to those achieved using the hard-limited correlator [Vecchi, 1988]. In the ALN approach adopted here, decisions as to whether the received
data bit is a ‘1’ or a ‘0’, are based not only on the desired code sequence structure but also on that of other users on the channel; the aim being to further reduce the number of errors arising due to MAI. In addition, whilst the neuromorphic systems were concerned solely with sparse encoding sequences, the use of Gold-like codes, a form of PN sequence, is also examined. Although, for reasons previously mentioned, such codes are not usually considered for use in unipolar optical systems, it is salient to study their performance in systems that are able to take into account the signals of interfering users. Finally, in contrast to the majority of the related work carried out using neural networks as receivers in non-optical BPSK CDMA systems, concentration is focused solely on the more complex case of asynchronous transmission by users.
Chapter 2  Theory

2.1  Introduction

This chapter provides the background to the concept of CDMA, including how such techniques may be achieved and the advantages such systems would offer optical fibre LANs. Particular emphasis is placed on the required properties of the code sequences employed in incoherent systems using all-optical processing and the differences in code design arising from the contrasting nature of signal processing available compared to that in the electrical domain. Consideration is then given to the components of an optical fibre CDMA LAN employing these “optical” codes, including a discussion of the types of receiver previously proposed for use in such systems. There follows an introduction to the theory of ALNs covering the configuration, properties and learning algorithms of these ANSs. Finally, ideas are presented as to how an ALN may be perceived as the basis of a generalised CDMA receiver.

2.2  CDMA Principles

CDMA refers to systems in which the multiple-access capability is achieved primarily by coding. In such systems the need for stringent coordination in time or frequency between stations, as in TDMA or FDMA, is removed by the mapping of data to a complex time-frequency representation. The most common form of CDMA is Spread Spectrum Multiple Access (SSMA), in which each user is assigned a particular code sequence which is modulated on the carrier with the data [Pursley, 1977]. Methods of spectral spreading by which SSMA may be achieved include Direct Sequence (DS) schemes, where data is multiplied by high rate code sequences, Frequency Hopping (FH), where data is transmitted over a cyclically changing pseudorandom set of frequencies according to a code sequence, chirped systems, where the data is transmitted on a carrier whose frequency is determined by a sweep generator, and Time Hopping (TH) where the data is transmitted at a cyclically changing pseudorandom instant in time. More recently, Spread Spectrum (SS) systems have been also been proposed based on the masking properties of chaotic signals [Oppenheim, 1992]. With the exception of the chirped and chaos-based systems, all the above methods employ specially designed codes to control their operation. For more details of these systems and an introductory tutorial to SS techniques see [Pickholtz, 1982][Dixon, 1984]. Although frequency hopped, chirped and time hopped schemes have been reported for use in optical CDMA systems, most research has been concentrated on DS-type schemes. This is because the necessary encoding and decoding processes may be readily
implemented using simple optical fibre delay-line structures. With this in mind, the term CDMA will be used to refer to this form of SS technique from now on, unless otherwise stated.

Shannon's channel capacity work explained that the transfer of data in noise could be improved by increasing either the SNR or the bandwidth across which the data is sent [Shannon, 1949]. The CDMA systems considered here take the latter approach and use the rapid modulation of single data bits by carefully selected high rate code sequences to spread the data over a much larger bandwidth than is necessary for transmission. The multiple access capability is achieved by designing the users' codes to have orthogonal-like properties in order to minimise mutual interference and thus permit extraction of a desired code in the presence of those of other users.

Data is recovered, or "despread", at the destination receiver via a correlation operation. In this way only code sequences destined for a particular receiver are despread, other interfering sequences appear as noise and remain spread. The improvement in SNR given by the bandwidth spreading and despreading operation is proportional to the ratio of the spread signal bandwidth (B_{ss}) to the baseband information rate (R_{info}) and is termed the processing gain (G_p):

\[ G_p = \frac{B_{ss}}{R_{info}} \]

Traditionally, SS techniques have been used in radio systems to combat either fading or interference, to enable a multiple access capability or to provide a degree of message privacy. The transfer of such techniques to LANs allows the same benefits and moreover, should give improved performance since noise and transmission characteristics are much more easily controlled in this environment [Marshall, 1990]. The features of CDMA that make it attractive as a LAN access scheme are:

- Contention free, unrestricted access – access to the channel is immediate, since there is no need to determine whether the network is in use before transmission, and unlimited in duration;
- Concurrent user support – a number of users may simultaneously transmit over the network;
- Asynchronous transmission by users – no centralised network control is required;
- The ability to add new stations easily, subject to code sequence considerations;
Noise immunity – SS techniques provide robustness to environmental noise and other sources of interference. This property allows the support of many different types of users, i.e. data rates, other signals appearing as noise subject to code sequence considerations. This permits node hardware to operate at rates suited to their own needs rather than at the network medium rate.

2.3 CDMA Code Sequence Design

Central to the design of any CDMA scheme is the choice of the high rate code sequences onto which data is mapped. In DS systems, a single data bit is divided into a sequence of 'F' chips. Two properties are required of the chip sequences used in CDMA systems:

- Any sequence in the code set is easily distinguished from a time shifted version of itself. To achieve this, the Auto-Correlation Function (ACF) of each chip sequence to be used in the system should have a maximum at the zero shift and minima at all other shifts to allow detection of the desired signal;

- Any sequence in the code set is easily distinguished from any other possibly time shifted sequence in the code set. To achieve this, the Cross-Correlation Function (CCF) between pairs of sequences in the code set used should be as small as possible for all shifts thereby minimising the interference of other users during extraction of a desired signal and facilitating a multiple access capability.

The choice of the code sequences and modulation techniques used in CDMA systems depends on system constraints; in particular the processing methods available as shown in Figure 2.1. Traditional radio- or copper-based DS systems take advantage of the ability to map binary (0, 1) sequences to bipolar (+1, -1) sequences in order to achieve the desired correlation properties. The code sequences are used to pseudorandomly shift the phase of a carrier which is also being phase modulated by the data sequence to be transmitted. Despreading is achieved via coherent correlation of a bipolar channel sequence with a bipolar reference sequence.

In optical systems employing optical fibre delay-line matched filters in an attempt to prevent an electronic processing bottleneck, both the channel sequences and correlator taps (i.e. reference sequence) are constrained to be unipolar since such optical processors combine signals by summation of optical intensity. The correlation operation performed is therefore incoherent and sub-optimal. Incoherent optical systems typically send a high rate code sequence for a data ‘1’ and no sequence for a data ‘0’ in an OOK style modulation format.
Figure 2.1. Correlator schematic highlighting the differences between incoherent optical and coherent electronic correlation. At the output the incoherent optical auto-correlation peak value is shown above the equivalent coherent correlation (in parentheses) for the sequence 1110010. Coefficients $a_j$ are (+1, -1) for the conventional correlator and (0, 1) for the optical correlator.

The contrast in processing abilities between the electronic and optical domains has led to the development of distinct coding sequences suited to the particular application environment:

- PN and related sequences for coherent (conventional) CDMA systems where bipolar signalling may be used;
- Sparse codes for optical CDMA systems using unipolar signalling and incoherent correlation.

Examples of the two code sequence formats are given in Figure 2.2 where their structural differences are clearly evident. Figures 2.2 (a) and (b) represent Gold codes, examples of a family of codes commonly employed in conventional bipolar CDMA systems, whereas Figures 2.2 (c) and (d) are examples of sparse OOCs developed for use in incoherent all-optical systems.

2.3.1 Conventional CDMA Sequences

The term conventional CDMA sequences is used here to refer to those PN (also called pseudorandom) and related sequences traditionally used in radio based SS systems, such as maximal-length shift-register sequences or “m–sequences”, Gold codes, etc. The
construction, properties and analysis of such sequences are well documented in the literature and are not discussed further here. More information may be found in an excellent review article by Sarwate and Pursley [Sarwate, 1980] and also in a recent discussion of wideband coding methods for CDMA systems [Mowbray, 1992].

The advantages of using PN sequences are that they are easily generated using shift registers and their correlation properties are well understood. These properties may be traded for others, such as larger code sets to support more users or to enable better synchronisation. The distribution of the cross-correlation interference arising with PN
sequences may also be modelled as Gaussian for large numbers of sufficiently long codes, thus simplifying performance analysis. However, although PN codes may be used in systems performing electronic correlation of bipolar reference sequences with unipolar channel sequences to achieve the same SNR as bipolar baseband CDMA systems, as mentioned in chapter 1, there is a need for the development of coherent optics if this level of performance is to be coupled with the advantages of high speed optical processing.

2.3.2 Sparse Codes for Optical CDMA Systems

Problems are experienced in transferring conventional PN sequences to CDMA systems employing optical processing elements since they rely on their bipolar representation to achieve the desired correlation properties. However, the incoherent correlation performed by optical processors can only sum intensities and cannot distinguish polarity, since intensity is proportional to amplitude squared. Therefore only unipolar representations are possible in these systems.

The effect of this limitation is demonstrated in Figure 2.3; part (a) shows the correlation functions of the two Gold codes shown in Figure 2.2 when used in the bipolar format employed in conventional CDMA systems, whereas part (b) shows the correlation functions obtained when the same codes are used in a unipolar format and processed by an incoherent optical correlator. Comparison of the two figures shows that the difference between the in-phase ACF peak and the peak of the out-of-phase ACF or CCF, referred to as the Index of Discrimination (IOD), is much smaller in the unipolar system than in the bipolar system. In mapping PN codes to unipolar sequences the sum to zero ability previously responsible for the desirable correlation properties is lost. As a result of this, the even balance of ‘1’s and ‘0’s now gives larger relative cross-correlation peaks resulting in high interference and unacceptable performance for a small number of users.

The key to the problem is the necessity to have many more ‘0’ than ‘1’ chips in order to avoid high cross-correlation values, hence the term sparse codes. Two main types of sparse codes have been developed for use in incoherent optical CDMA systems. The first, prime codes, are formed by time-mapping of one-coincidence sequences originally developed for use in a different form of CDMA system [Shaar, 1983b]. Appropriately transformed, these codes have cross-correlation functions with a maximum of one or two [Shaar, 1983b] and, as such, have been advocated for DS-type optical CDMA LAN applications [Prucnal, 1986b]. However, the prime codes can have high out of phase auto-correlation values and as such they are not directly suited to completely asynchronous systems. This problem does not occur with the second type, OOCs [Chung, 1989][Salehi, 1989b], and it is these codes that are considered in the context of this work. They are so named to emphasise their unipolar (optical) and orthogonal-like nature; hence their suitability to incoherent optical
Figure 2.3. Auto- and Cross-correlation functions for the 31 chip Gold codes of Figure 2.2: (a) in a bipolar CDMA system; (b) in an incoherent optical CDMA system.
correlation. Like the prime codes, they are only pseudo-orthogonal since cross-correlation values can never be less than one. Nevertheless, codes can be developed with desirable correlation properties as shown in Figure 2.4 for the OOCs of Figure 2.2.

An \((F, K, \lambda_a, \lambda_c)\) OOC is defined by:

- the length \(F\);
- the weight \(K\) – the number of ones in the code;
- the auto-correlation constraint \(\lambda_a\) – the maximum ACF value for all shifts other than the zero shift;
- the cross-correlation constraint \(\lambda_c\) – the maximum CCF value for all shifts.

When \(\lambda_a\) and \(\lambda_c\) are equal, the above definition is abbreviated to an \((F, K, \lambda)\) OOC. The minimum code overlap class exists when \(\lambda_a = \lambda_c = \lambda = 1\). Most work has been done on this class of codes. A comprehensive guide to the design, analysis and construction of OOCs is given in [Chung, 1989].
Advantages of the new codes include simple correlator design using fibre-optic delay-lines. Although the codes are not suitable for generation by shift registers they may also be produced by optical fibre delay-lines. The ability for both encoding and decoding to be done optically means that in theory more chips per bit can be used. This allows more users to be supported since performance has been shown to improve with an increasing number of chips [Prucnal, 1986b].

Although OOCs with correlation constraints of one minimise mutual interference in the incoherent optical system, these constraints impose a severe limitation on the number of code sequences in a set of given length. The upper bound on the number of OOCs in a set (N) has been shown to be [Chung, 1989]:

$$n \leq \left\lfloor \frac{(F - 1)}{K(K - 1)} \right\rfloor$$

where \(\lfloor x \rfloor\) denotes the integer part of a real value \(x\). A similar result has also been reported by [Mendez, 1990]. This represents the maximum number of stations or subscribers that a network could support if each is assigned a unique code as its address. However, the number of users that may simultaneously access the network is dictated by the required Bit Error Rate (BER) for the system and, as such, may be even lower. High weights are essential for an acceptable BER and the support of many concurrent users because the difference between the in-phase ACF peak and the out-of-phase ACF and CCF peaks of OOCs are relatively low compared to those of PN codes in bipolar systems. For example, a code weight of 8 has been proposed for a BER of 10\(^{-9}\) [Mendez, 1990]; a typical minimum acceptable value for an optical LAN. However, the expression for the number of subscribers in a code puts an upper limit on the weight and as such there is a trade-off between the number of users able to simultaneously access the network and error performance for a given length code. The choice of code length must therefore be based on the number of codes in a set of a given length, the code weight for a given length and whether a combination of the number of codes and weight can be found to support the required number of concurrent users at the target BER.

The constraints in choosing code length and the limited number of OOCs of given length compared to PN sequences, such as Gold codes, mean that longer codes are required to provide the same BER as a bipolar system. As such, incoherent optical CDMA is less efficient than bipolar baseband CDMA. In order to improve performance, recent studies have been conducted in relaxing OOC correlation constraints to two, i.e. \(\lambda = 2\), in the hope of supporting more users at comparable error rates. Results indicate that this may indeed be possible. For example, although it has been shown that \(\lambda = 1\) codes permit error rates several orders of magnitude lower than \(\lambda = 2\) codes, for the same number of users, weight
and length, choosing $\lambda = 2$ permits higher code weights than possible with $\lambda = 1$ and this may be used to compensate for the loss in performance [Azizoglu, 1990, 1992].

2.4 Optical CDMA Networks

2.4.1 System Description

The configuration of an optical CDMA system based on a star topology and consisting of $N$ transmitter-receiver pairs is shown in Figure 2.5. At a transmitter each low rate optical or electrical data bit is mapped to a high rate optical chip sequence via a delay-line encoder. The optical sequence selected by a transmitter usually corresponds to the address of the receiver to which it wishes to send data i.e. the code stored there, although other code assignment strategies have also been suggested [Smythe, 1985b]. The receiver compares the received signal with its stored sequence via a correlation operation, also effected by a delay-line, and compares the result to a threshold to decide the nature of the data bit sent. Extraction of a desired sequence in the presence of interfering users is achieved via the correlation properties of the codes which attempt to minimise post-correlation interference and thus the chance of the decision threshold being falsely attained.

Following the notation used in [Salehi, 1989c] the output of the $n$th transmitter’s optical encoder may be expressed as:

$$s_n(t) = s_n b_n(t) D P_n(t)$$  \hspace{1cm} (2.1)

where $s_n$ denotes the $n$th user's optical intensity, $b_n(t)$ the binary data bit and $D P_n(t)$ the selected OOC. Assuming that all transmitters send continuously to their respective receiver pairs the $n$th user’s binary data signal may be represented as:

$$b_n(t) = \sum_{\ell = -\infty}^{\infty} b^{(n)}_{\ell} P_T(t - \ell T)$$  \hspace{1cm} (2.2)

where $b^{(n)} = b^{(n)}_{\ell}$ is the data sequence of the $n$th transmitter which using an OOK modulation format takes on the value ‘0’ or ‘1’ with equal probability for each $\ell$ and $P_T$ is the rectangular pulse defined as above. The $n$th user’s OOC is characterised as:

$$D P_n(t) = \sum_{j = -\infty}^{\infty} A^{(n)}_j P_{Tc}(t - j T_c)$$  \hspace{1cm} (2.3)

$$P_{Tc}(t) = \begin{cases} 1 & 0 \leq t \leq T_c \\ 0 & \text{elsewhere} \end{cases}$$
\( p_{tc}(t) \) is the chipwaveform and \( A(n) = (A^{(n)}_j) \) is the \( n \)th periodic sequence of binary optical chip pulses \((0, 1)\) with period \( F = \frac{T}{T_c} \) and weight \( K \). Thus the bandwidth of \( s_n(t) \) is of the order of \( F \) times that of \( b_n(t) \).

The signal at the input to each receiver on the network may be represented as in (2.4) below, based on the assumptions of: (i) asynchronous transmission by users, i.e. no network time reference at either the bit or chip level, (ii) incoherent sources such that the optical intensity of overlapping users code sequences sums and (iii) interference other than the presence of multiple users on the channel is negligible i.e. MAI is dominant.

\[
r(t) = \sum_{n=1}^{N} b_n(t - \tau_n) D P_n(t - \tau_n)
\]  

(2.4)

where \( \tau_n \) is the relative delay associated with the \( n \)th signal and all signals are assumed to have the same bit rate, signal format and incident power (i.e. no near-far effect) at each receiver.

2.4.2 Simple Correlator

The output of a simple active correlator perfectly synchronised to the first users signal, i.e. with \( \tau_1 = 0 \), may be written:
\[ Z_1 = \frac{1}{T_c} \int_0^T r(t) D_P(t) \, dt \]

\[ = b_0^{(1)} K + I_1 \]  \hspace{1cm} (2.5)

Alternatively, if a passive tapped delay-line is used an equivalent result is achieved by convolving the input signal with a filter impulse response equal (matched) to the time reversed complex conjugate of the desired signal. The first term of (2.5), \( b_0^{(1)} K \), represents the desired signal component whereas the second term, \( I_1 \), is the interference due to cross-correlation products arising from the presence of the other \( N - 1 \) users’ signals. Errors only occur when the desired user sends a data ‘0’ and the interference term is such that the output threshold is achieved. For \( \lambda = 1 \) and a number of interfering users less than the weight of the code, i.e. \( N - 1 < K \), a threshold can be chosen so that the error due to MAI, neglecting other noise sources, is zero. When the number of other users is greater than or equal to the code weight, i.e. \( N - 1 \geq K \), errors occur due to MAI with non-zero probability.

Development of the BER of such a system requires the knowledge of the probability density function associated with the MAI signal \( I_1 \). To this end a technique based on the use of optical disk patterns has been used to investigate the interference properties of OOCs and then develop the probability density function associated with any two interfering OOCs [Salehi, 1989b]. This was extended to produce probability density functions for \( I_1 \) by convolution of the \( N - 1 \) probability density function pairs associated with the desired user’s OOC and each interfering user’s OOC [Salehi, 1989c]. The analysis considered the cases of chip synchronous interference, where users’ chips are assumed to overlap completely, and ideal chip asynchronous interference, where the time shifts between users can take on any value, but their OOCs are constrained to have no adjacent pulses. The probability density functions so obtained were then used to calculate upper and lower bounds respectively on the exact probability of error. The probability of error \( P_E \) derived for the simple correlator, operating under chip synchronous (worst case) interference conditions, is stated in equation (2.6) for future reference [Salehi, 1989c]:

\[ P_E = \frac{1}{2} \sum_{i = T_h}^{N - 1} \binom{N - 1}{i} \frac{K^2}{2^F} \left(1 - \frac{K^2}{2^F}\right)^{N - 1 - i} \]  \hspace{1cm} (2.6)

where \( N \) is the number of system users and \( T_h \) is the receiver decision threshold setting. The result is identical to that obtained by a different analysis in [Azizoglu, 1990, 1992].
2.4.3 Hard-limited Correlator

Use of a non-linear optical threshold element has been proposed to improve performance by eliminating the effects of localised optical intensity at the input of the simple correlator [Salehi, 1987b][Salehi, 1989c]. The device clips (normalised single-user) intensity values greater than or equal to ‘1’ back to ‘1’ and those less than ‘1’ to ‘0’; the resulting receiver structure is equivalent to an optical AND gate since if the output threshold is set at the code weight it is only reached when chips are present at all of the input taps [Salehi, 1989a]. As an example a simple correlation receiver configured to detect the OOC ‘1011000100000’ would have a threshold set to the auto-correlation peak of 4, the code weight, and an input of the form ‘3000000100000’ would cause this threshold to be falsely achieved due to the high optical intensity present at the first chip, despite the fact that the form of the desired pattern is not reproduced [Salehi, 1989a]. Hard-limiting the input pattern creates the sequence ‘1000000100000’, with a maximum correlation value of 2, which does not produce an error since the threshold is no longer achieved.

In this case the interference pattern at the output of the correlator may now be characterised as:

\[
I_{I}^{hl} = \frac{1}{T_c} \int_{0}^{T} g(I_1(t)) DP_1(t) \, dt \quad (2.7)
\]

where the unclipped interference signal is defined as:

\[
I_1(t) = \sum_{n=2}^{N} s_n(t - \tau_n) \quad (2.8)
\]

and the optical non-linear threshold element is represented by:

\[
g(x) = \begin{cases} 
1 & x \geq 1 \\
0 & 0 \leq x \leq 1 
\end{cases} \quad (2.9)
\]

again the value ‘1’ denotes the normalised value for a single-user’s intensity. As in the case of the simple correlator, upper and lower bounds on the probability of error for the hard-limited correlator system have been derived based on the assumptions of chip synchronous and ideal chip synchronous conditions respectively [Salehi, 1987b][Salehi, 1989c]. A more recent result for the probability of error of a hard-limiting correlator, operating under chip synchronous conditions, has been shown to agree closely with the previous result and is presented below in equation (2.10) [Azizoglu, 1990, 1992] for future reference:

\[
P_E = \frac{1}{2} \sum_{i=0}^{K} (-1)^i \left( \begin{array}{c} K \\ i \end{array} \right) \left( 1 - \frac{q}{K} \right)^{N-i}, \quad \text{where } q = \frac{K^2}{2F} \quad (2.10)
\]
2.5 Adaptive Logic Networks

2.5.1 Structure

ALNs are a type of ANS based on boolean logic [Armstrong, 1979][Armstrong, 1991a]. Specifically an ALN consists of processing elements of two-input logic gates arranged as the non-leaf nodes in a binary tree structure as shown in Figure 2.6. These nodes are assigned logical operators from the set of functions AND, OR, LEFT, RIGHT as shown in Table 2.1. The set consists precisely of the set of non-constant, increasing boolean functions of two variables [Armstrong, 1979] meaning that a change in one of the inputs from a logic '0' to a logic '1' will never cause a change at the output from a logic '1' to a logic '0'.

<table>
<thead>
<tr>
<th>Input (left, right)</th>
<th>AND</th>
<th>OR</th>
<th>LEFT</th>
<th>RIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2.1. Truth tables for the node functions of an adaptive logic network.

The motivation behind the use of a tree structure and the choice of node functions is that such structures have been shown to have insensitivity properties whereby the output of a binary tree function tends to remain constant in the event that small perturbations are applied to an input vector [Armstrong, 1979][Armstrong, 1991b]. As such, they have been proposed for pattern recognition applications where it is required that a trained tree classifies previously unseen patterns according to their similarity to the patterns on which it has been trained in a process termed "generalisation".

Input to the tree is presented in the form of a boolean vector whose elements are connected at random to the leaves of the tree. Inversions are included between some of these elements and the leaves to prevent any restriction in the function realisable by the tree arising from the non-constant and increasing nature of the nodes. Inversions need only be included at the tree inputs, since those required elsewhere in the tree may be moved by repeated application of DeMorgan’s theorem.

If the input is not in a form suitable for direct presentation to the tree, i.e. the input vector elements are real or integer values, then it must be mapped into a boolean vector first. Typically this involves quantisation of the range of values of each input vector element. Each quantisation level is encoded as a bitstring by a look-up table so that real or integer values are mapped to boolean vectors. Presentation of an unencoded input vector results in
Figure 2.6. Schematic diagram showing the essential features of an ALN. A balanced binary tree consists of $2^L - 1$ non-leaf nodes arranged in $L$ layers. As shown the configuration permits continuous valued inputs and a binary output decision. Applications requiring continuous outputs would use one tree per encoded output bit.

the generation of a bitstring for each of its elements. These bitstrings are then concatenated to form a vector for input to the ALN. The types of encoding methods considered are covered in more detail in chapter 3.
2.5.2 The ALN Learning Algorithm

Initialisation

Before training, the binary tree must first be initialised. This involves random assignment of functions from the set \((\text{AND}, \text{OR}, \text{LEFT}, \text{RIGHT})\) to the non-leaf nodes and random connection of the leaves of the tree to the elements and inversions of the input vector. Patterns from the training set are then presented in pseudorandom order as input vectors and the tree adapted by means of a suitable learning algorithm until either an acceptable level of performance is attained or for a given number of passes through the entire training set or "epochs".

The Concept of Responsibility

The first stage in deciding which nodes to adapt involves presentation of an input vector and evaluation of all node outputs from the leaves of the tree to the root node (output) in a forward pass. The increasing property of the node functions means that the positive correlation between node output and tree output can be used to guide adaptation by using the desired tree output as the desired node output. The desired tree output is used in conjunction with the actual output and inputs of the node currently under consideration to recursively propagate a "heuristic responsibility" signal in a backward pass from the root to the leaves. It is this heuristic responsibility signal which is used to identify whether the functions of the children of the node under consideration should be updated. This concept is described below and summarised in Figure 2.7.

In the Atree 2.0 Software Library [Dwelly, 1990][Armstrong, 1991b] used for this work, the heuristic responsibility employed consists of a combination of previously developed responsibility assignment schemes, comprising:

(a) Responsibility of the root node [Armstrong, 1979]. The root (output) node is always heuristically responsible (see Figure 2.7(a));

(b) Error Responsibility [Armstrong, 1979]. If a node is heuristically responsible and one of its input signals is not equal to the desired network output, that input signal is termed an “error”. If the signal on the left (right) is in error then the right (left) child is made heuristically responsible (see Figure 2.7(b)). This method of assignment is useful when a certain child cannot correct the error i.e. when it is a leaf;

(c) True Responsibility [Armstrong, 1979]. The right (left) child is made heuristically responsible if the node is heuristically responsible and a change in the value of the
right (left) input would change the node's output. For example, given heuristic responsibility, when the node is an AND and its left (right) input is a ‘1’ or when the node is an OR and its left (right) input is a ‘0’ (see Figure 2.7(c)). The version of true responsibility described in [Armstrong, 1979] also included cases when the node was assigned as a LEFT or RIGHT function, however these are now included within the extended definition of (d) below;
(d) Responsibility of LEFT / RIGHT nodes [Armstrong, 1991a and b]. If a node is heuristically responsible and its function is either LEFT or RIGHT then both its child nodes are deemed heuristically responsible (see Figure 2.7(d)).

Adaptation

Node function is determined by the values of two counters in each node which are activated whenever it is heuristically responsible [Armstrong 1991a and b]. The counters correspond to (0, 1) and (1, 0) input pairs at the node and are updated according to Table 2.2. During adaptation the desired output of the tree is used as the desired output of the node according to the positive correlation between node and tree functions caused by the node properties. Counters are not required for (0, 0) and (1, 1) inputs since the node functions all have the same outputs for each of these cases and thus adaptation is not required.

<table>
<thead>
<tr>
<th>Node Input (L, R)</th>
<th>Desired Output</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0</td>
<td>1</td>
<td>Increment (1, 0) Counter</td>
</tr>
<tr>
<td>1 0</td>
<td>0</td>
<td>Decrement (1, 0) Counter</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
<td>Increment (0, 1) Counter</td>
</tr>
<tr>
<td>0 1</td>
<td>0</td>
<td>Decrement (0, 1) Counter</td>
</tr>
</tbody>
</table>

Table 2.2. Counter update rules for heuristically responsible nodes.

In order to determine the node function assigned, the counters are compared to their start value or threshold. Functions are assigned following Table 2.3 where the counter initial value is assumed zero. After adaptation is complete, the nodes which have been assigned with LEFT or RIGHT functions may be pruned from the tree in a process termed "folding". These functions serve only to cut-off unnecessary sub-trees during training. The final folded tree consists solely of AND and OR gates.

<table>
<thead>
<tr>
<th>(1, 0) Counter Value</th>
<th>(0, 1) Counter Value</th>
<th>Function Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 0</td>
<td>&gt; 0</td>
<td>OR</td>
</tr>
<tr>
<td>&gt; 0</td>
<td>&lt; 0</td>
<td>LEFT</td>
</tr>
<tr>
<td>&lt; 0</td>
<td>&gt; 0</td>
<td>RIGHT</td>
</tr>
<tr>
<td>&lt; 0</td>
<td>&lt; 0</td>
<td>AND</td>
</tr>
</tbody>
</table>

Table 2.3. Node assignment rules for heuristically responsible nodes.

2.6 Adaptive Logic Networks as Optical CDMA Receivers

The tapped delay-line matched filter may be viewed from a connectionist point of view as either a single neuron of the McCulloch-Pitts type or alternatively as a feedforward neural
network with no hidden layer and a single output unit, as shown in Figure 2.8 (a). The tap coefficient values of the filter correspond directly to the weights in the input connections of the neuron (not shown in the figure) and the sum and threshold operation corresponds to the local non-linear processing operation performed by the output neuron where in this case the threshold is set to the code weight.

It has been suggested that for some applications a generalisation of this structure to include hidden units as in Figure 2.8(b), as well as the normal connections from the input directly to the output (not shown), may lead to improved performance for certain kinds of signal and interference [Hecht-Nielsen, 1990]. The concept is concerned with waveform recognition and the fact that incorporation of a hidden layer should allow decision making of a conditional nature that is not possible using a straightforward weighted sum approach. In the event that no improvement is possible over the conventional approach, the weights associated with the hidden layer should tend to zero during training, effectively removing it and resulting in the standard filter structure.

In the context of a CDMA receiver application it is conceivable that this approach could lead to each neuron in the hidden layer becoming associated with a particular user’s waveform, thereby forming a series of individual matched filters or so-called “grandmother cells” (again see Figure 2.8(b)). The hidden to output layer weights could then be adapted to make a decision as to the output based on the relevant information provided by the hidden layer outputs i.e. data value, signal level, relative time shifts of encoding sequence etc. for each user. In this way, in contrast to the single matched filter approach, the presence of other users on the same channel may be taken into account. This scenario is somewhat idealised in that the hidden layer representation is likely to be more distributed, unless it is specifically trained as such, but it serves to illustrate the basic motivation behind the use of neural networks as receivers in CDMA systems.

The generalisation from the matched filter / single neuron equivalence to the feedforward neural network described above can also be applied to neural networks consisting of logical elements. The hard-limited correlator receiver has been shown to be equivalent to an optical AND gate, as depicted in Figure 2.8(c)[Salehi, 1989a]. In a system using OOCs of weight 2, the hard-limited correlator would be equivalent to a 2-input AND gate which in turn could be considered to be a special case of an ALN, as prescribed in section 2.5. This can then be extended to the case of a code of weight K by arranging a series of 2-input logic elements to form an equivalent K-input AND gate. These units form the basis for generalisation to the ALN CDMA receiver. By analogy to the analogue neuron implementation described above, the ALN CDMA receiver may be considered to consist of a series of such logic sub-networks, each associated with a particular user’s signal, plus an additional logic stage to make a decision based on the output states of these sub-networks. In this approach the
Figure 2.8. Neural networks as CDMA receivers: (a) Correlator and single neuron equivalence. (b) Generalisation of the correlation receiver to a feedforward neural network receiver shown here with a single hidden layer. (c) Equivalence of the hard-limited correlator to an optical AND gate. Generalisation of the single AND gate again leads to a feedforward neural network structure but in this case the processing elements are based on logic gates.
conditional nature of the output stage in signal recognition is more apparent. Again an actual implementation is unlikely to be as strictly defined as this since the sharing of gates or inclusion of redundancy are probable, however the potential for ALN CDMA receivers is suitably illustrated. Furthermore, given suitable encoding methods, the ALN receivers would be able to consider intensity information denied to the hard-limited receivers.

2.7 Summary

This chapter first introduced the basic principles of CDMA and outlined the features that make it suitable for application as a LAN medium access scheme. Attention was then focused on optical CDMA systems and ALNs were proposed as novel receivers for such environments. Particular emphasis has been placed on the following areas:

• The desired correlation properties of CDMA code sequences;

• The differences between conventional electronic and incoherent optical signal processing systems;

• The consequent differences in CDMA code sequence design arising from the contrasting nature of signal processing techniques, i.e. PN sequences for systems using electronic processing and sparse code sequences for incoherent optical systems;

• An optical CDMA network based on a star topology and the previous approaches to decoding in such systems, i.e. the simple and hard-limiting correlators based on optical fibre tapped delay-line matched filters;

• The concept of ALNs and their application to decoding in optical CDMA networks; the aim being to improve performance by considering the presence of both the desired and interfering users' signals on the network.
Chapter 3  Simulation

3.1  Introduction

Software simulation was used to investigate the performance and behaviour of ALN receivers in an incoherent optical CDMA system. The simulation can be divided into three separate stages and these form the main sections of this chapter. The first stage, described in the next section, is concerned with the specification of an environment to permit the generation of signals of the type found on the communications channel at the input to receivers in such a system. The second stage, covered in the following section, is devoted to the training of ALN receivers using these signals and the selection of the parameters associated with this process. The final stage, presented in the last section, describes the testing of the ALN receivers and the methods used to evaluate their performance.

3.2  Environment

The simulation of network data is based on a model of an incoherent optical CDMA channel. This is used to develop data generators to produce input signals for the receivers under consideration based on the specification of the code sequences to be used in a particular system. The system configurations examined in the course of this work are presented in the final part of this section.

3.2.1  The Channel Model

The purpose of the channel model is to completely define an idealised optical CDMA channel. The constraints and assumptions contained in this definition can then be used as the basis to generate input for the receivers under consideration and to provide a context for simulation results. The features of the modelled optical CDMA channel are similar to those used by [Salehi, 1989c] and are described below:

- Each user is assigned a unique code sequence of the same length, ‘F’ chips, so that user data bit and chip rates are matched;
- An OOK modulation scheme is employed. A user transmits data bits with equal probability sending a code sequence for a data ‘1’ and no sequence for a data ‘0’;
- All transmitters send continuously to their respective receiver pairs:
• All users are assumed to have the same effective optical power at any given user's receiver;

• The system is based on incoherent light sources so that the optical intensities of overlaid multiple users sum together i.e. the channel is additive;

• A receiver is assumed to be perfectly synchronised to the transmitter whose code sequence it intends to receive;

• The transmission of data on the network is not under centralised control, thus, although it is required for communication between pairs of users to establish synchronism, in general signals interfering with a particular desired signal will be asynchronous;

• The only system performance degradation is due to MAI caused by the simultaneous presence of other users' signals on the channel. MAI is assumed to be dominant and all other forms of noise are neglected since optical fibre is immune to electromagnetic noise.

3.2.2 Receiver Input Signal Generation

Data generators were configured to allow the study of a particular receiver in the presence of a number of interfering users and they were based on the features stated in the previous section. The data generators produce composite sequences consisting of the sum of the desired and interfering users' signals on the channel at the input to a receiver in the form given in equation (2.4) of chapter 2. Since the receiver under investigation is assumed synchronised to the desired user's transmitter, its associated delay, \( \tau_1 \) (assuming \( n = 1 \) for the desired user), is zero and thus its contribution to the composite signal is either its code sequence or no sequence. However, the signals of interfering users are not in general synchronised to this receiver, i.e. \( \tau_n \neq 0 \) given \( n \neq 1 \), thus, within the same time window, the contribution of each to the composite signal generally consists of two parts, the end of the previous code sequence multiplied by the previous data value and the start of the next code sequence multiplied by its data value. The process of composite sequence generation is shown graphically in Figure 3.1.

In the channel model of section 3.2 it was stated that transmission of data onto the network was not under centralised control and, as such, the transmissions between all users were completely asynchronous. The data generation methods considered impose one further constraint on the system model. Although the relative transmissions between users are assumed asynchronous at the data bit level, they are assumed to be synchronous at the chip level. In a real system this is unlikely to be the case. Nevertheless, it provides
Figure 3.1. Formation of a composite signal demonstrating the OOK modulation scheme, the asynchronous nature of interfering users and the summing of optical intensities. (a) Desired user’s sequence ‘1100000000000000’ (synchronised to receiver), data value = ‘1’. (b) Interfering user #1, sequence ‘1010000000000000’, “left” data value = ‘1’, “right” data value = ‘0’. (c) Interfering user #2, sequence ‘1001000000000000’, “left” data value = ‘0’, “right” data value = ‘1’. (d) Resultant composite sequence at receiver input.

for simulation of worst case conditions since interfering chips overlap completely with a desired chip whereas in a real system desired chips are only likely to be overlapped by a fraction of an interfering chip. Furthermore, this simplifies simulation since the channel sequence need only be sampled once per chip before presentation to the discrete inputs of the receivers. The method used could be extended to various degrees of chip asynchronism by refining the representation of a chip to effectively permit sampling more than once per chip.

Two basic techniques were considered for generating data:

- Method ‘A’ – based directly on simulation of the channel model;
- Method ‘B’ – based on generation of all possible composite sequences for a given number of users.
In approach ‘A’ each transmitter in a given system configuration is allocated both a unique code sequence and an independent discrete uniformly distributed random number generator, producing a zero or a one, to represent the data source. Multiplication of the code sequence and the data value is used to simulate the OOK modulation scheme. The chip values of the composite signal are formed by summing the chip contributions of each user on the channel. In the case of the desired user’s signal, where the transmitter is assumed synchronised to the receiver, its contribution to the composite signal is either its code sequence or nothing depending on the output of the random number generator deciding the data bit. However, in determining the interfering users’ contribution to the composite signal, the asynchronous nature of the system has to be taken into account. Previous and next sequences are generated for each interfering user and multiplied by their respective data values. Each pair of sequences are then concatenated and shifted by a random number of chip periods to simulate the relative delays between users. Again, independent random number generators are used for each interfering user. The first ‘F’ chips of each interferer’s shifted sequence are then added to the desired user’s contribution to create the final composite sequence which is then written to a file. The corresponding target receiver output value i.e. the desired user’s data value is also saved to the file. The process is repeated until the required number of input-target pairs has been generated.

The second method, ‘B’, generates an exhaustive set of composite sequences based on all possible combinations of user contributions. For the desired user, the set of contributions contains the assigned code sequence and an all-zero sequence. For each interfering user the set contains the first ‘F’ chips of all possible shifts of all possible combinations of two independently modulated concatenated sequences. Composite sequences are formed by producing all the possible combinations of selecting a sequence in order from each set and adding them. The composite sequences and associated target values corresponding to the desired user's data value are then stored in a file. Although the composite sequences are formed in an inherent order in this file, this does not affect the training of the ALNs since sequences are automatically presented in a pseudorandom order by the training procedure. The method described above generates the complete set of composite sequences in that each possible data and shift value (for interfering users) are produced exactly once. If large numbers of sequences are produced using method ‘A’, the distribution of individual composite sequences in the set would be expected to approach that obtained using method ‘B’.
Figure 3.2. The three (16, 2, 1) sequences used in the CDMA system simulations employing optical orthogonal codes. (a) Desired user’s code sequence. (b) and (c) Interfering users’ code sequences.

3.2.3 Code Sequence Allocation

The study considered the performance of ALN receivers in incoherent optical CDMA systems of three users; one desired and two interfering users. Three different configurations were investigated; the first two employed OOCs specially developed for use in such systems while the third used conventional PN-type sequences, Gold-like codes, more commonly associated with radio- and copper-based CDMA systems. The actual OOCs used are shown in Figure 3.2 and are of length (F) 16, weight (K) 2 and correlation constraint (X) 1.

The Gold-like codes used in the final system simulation are shown in Figure 3.3 and are of length 15. Gold-like codes were chosen since they exist at a code length similar to that of the OOCs used, allowing performance comparison, and because the true Gold codes more commonly used in conventional CDMA systems do not exist for that length. The number of users and lengths of codes employed here were chosen to permit investigation of ALN receivers with manageable data set sizes and to allow simulations within a reasonable computation time. The error rates achievable with these codes do not reflect practical values for real optical LANs, which typically require rates of less than $10^{-9}$ and should be able to support more users by employing longer code sequences and, in the case of OOCs, higher weights. However, both sets of codes used are shorter versions of
the types of codes that could actually be used in real systems and are constructed via the same principles.

### 3.3 ALN Training

The ALN receiver simulations were carried out using C language programs based on a library of ALN routines [Dwelly, 1990][Armstrong, 1991b] which include facilities for initialising, training and testing ALN trees. A typical training program first reads in a set of input-target pairs created by the data generators described previously. Following this the input vectors are mapped to a boolean form suitable for input to an ALN. The ALN may then be trained on the encoded data vectors using the learning algorithm described in chapter 2 so as to minimise the number of disagreements between the actual and desired ALN outputs.

#### 3.3.1 ALN Input Encoding

The composite sequences produced by the data generators are of a multi-level nature and must therefore be converted into bit vectors compatible with the boolean nature of ALN inputs before training (or testing) may be carried out. This may be achieved in two distinct ways: either by explicit encoding of the input chip intensity range or by direct hard-limiting of composite chip values.
Explicit Encoding Techniques

In this approach the magnitude of composite sequence chip values is quantised and each level is then assigned a unique bit string according to a set encoding scheme. The operation is performed for each chip in the composite sequence and the resulting bit strings concatenated to form a boolean vector for input to the ALN. In this way both intensity and temporal information are preserved; both being mapped into a spatial representation across the ALN inputs.

In the 3-user systems, described in section 3.2.3, the maximum possible composite chip value is three when all users contribute overlapping chips and the minimum is zero when no users contribute chips. Following the chip synchronous assumption, four quantisation levels are required. The particular encoding schemes employed in the simulations are shown in Table 3.1. Their operation is illustrated in Figure 3.4, where encoded examples of the input sequence given in Figure 3.4(a), are shown in Figures 3.4(c)-(g).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1000</td>
<td>000</td>
<td>00</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>1</td>
<td>0100</td>
<td>100</td>
<td>01</td>
<td>100</td>
<td>101</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>110</td>
<td>10</td>
<td>101</td>
<td>000</td>
</tr>
<tr>
<td>3</td>
<td>0001</td>
<td>111</td>
<td>11</td>
<td>001</td>
<td>011</td>
</tr>
</tbody>
</table>

Table 3.1. ALN input encoding schemes.

The methods chosen allow investigation of the effects of encoding length, code weight and hamming distance on ALN learning. The lengths of the bar, temperature and binary codes were chosen so as to support the desired number of users with the minimum length for their particular type. The scatter code lengths were chosen as the minimum required to give similar hamming properties as the other codes. The minimum length approach was adopted for reasons of efficiency and manageability since the encoding length multiplies the composite sequence length to give the actual ALN input vector length. Of the above, the bar code has the longest encoding length (4) and the binary code the shortest (2).

The term weight is used here to refer to the number of ‘1’s in each distinct code sequence of a particular encoding scheme. The bar code has a constant code weight of one for each code in the set, whereas the weight of the temperature code increases linearly with the quantisation level. The other encoding methods used have no obvious structure to their weight distribution.
Figure 3.4. Encoding of a composite sequence as a boolean vector suitable for input to an ALN.
(a) Raw composite sequence '2120010000000000'.
(b) Hard-limited encoding.
(c) Bar encoding.
(d) Temperature encoding.
(e) Binary encoding.
(f) Scatter encoding (d = 1).
(g) Scatter encoding (d = 2).
Y-axis:
(a) Normalised optical intensity
(b)-(g) Boolean
The hamming distance \( (d) \) between two binary numbers is the number of places in which they differ and, as such, it can be used as a measure of how similar (or different) binary vectors are for the purposes of pattern classification. The hamming distances between the quantisation levels shown in Table 3.1 are given in Figure 3.5. Both the bar code (Figure 3.5(a)) and scatter code 2 \( (d = 2) \) (Figure 3.5(e)) are identical from a hamming distance point of view, having a constant value of 2 between all levels. The temperature code (Figure 3.5(b)) and scatter code 1 \( (d = 1) \) (Figure 3.5(d)) are also identical in this respect having constant hamming distances of 1 between adjacent levels and proportionally graded values for distances between other levels. Despite these similarities, it should be mentioned that scatter codes have been proposed as a more efficient encoding technique than the bar and temperature codes [Smith, 1990]. Whereas bar and temperature codes preserve both the closeness and ordering of linear space in the hamming space, the scatter codes preserve closeness only. This permits the use of more points in a given space and thus more distinct codes for a given length.
The binary code (Figure 3.5(c)) does not possess a regular hamming structure. Indeed, for this reason it is not usually considered suitable for ALN systems since closeness is not preserved in hamming space [Armstrong, 1991b]. For example a hamming distance of one could equally represent either a large or small difference between unencoded values depending on whether it is derived from the most or least significant bit positions respectively. Nevertheless, it is included for completeness since it is the most compact encoding available and the input range here is limited to values of the same order of magnitude.

**Hard-limited Input**

An alternative approach to the explicit encoding techniques described above is to perform a hard-limiting operation on the chip values of the composite sequences. As in the case of the hard-limited correlation receiver, chip intensity values less than one (normalised single-user intensity) are clipped back to zero and those greater than or equal to one are set to one. This approach, shown in Figure 3.4(b), allows an ALN to consider only temporal information, which from the synchronised ALN point of view again becomes spatial across its inputs. Nevertheless the approach is useful since the results obtained permit comparison with the hard-limited correlator to study the effect of increasing the number of taps alone and with the encoded input ALN receivers to see if the retained intensity information is used or necessary.

### 3.3.2 Training Procedure

Apart from the choice of ALN input encoding scheme, the other parameter to be considered before training was the initial ALN tree size. Trees sizes of 12, 13, 14 and 15 layers, corresponding to 4096, 8192, 16384 and 32768 leaf nodes respectively, were chosen based partly on the heuristic that 10-15 layers were adequate for most problems [Armstrong, 1992]. Fifty individual trees were trained for each combination of encoding method and tree size in order to allow the effects of different random initial node assignments, input connections and pattern presentation order to be taken into consideration. Trees were trained for a maximum of fifty epochs since preliminary trials suggested most adaptation was performed within this period for the tree sizes used. The training set used was the exhaustive set generated using generation method ‘B’. The motivation for this was that the number of errors made by the correlation receivers was likely to be small compared to the size of the training set due to the CDMA sequence design and, furthermore, the number of these that would be correctable using adaptive techniques is likely to be even smaller. For the numbers of users and code lengths considered, the set used is of manageable size and is also guaranteed to contain examples of all possible error sequences.
3.4 ALN Testing

Before testing, the trees obtained from training were folded to remove the superfluous LEFT and RIGHT nodes. Evaluation of the trees was then carried out using test signals created by the data generators. Composite sequences were encoded using the same encoding scheme with which the tree under test was trained and then presented to the ALN input. Testing was used to compare the error rates obtained with ALNs and those obtained with the previously proposed receivers, to examine the effect of training parameters on performance and to study the behaviour of trained ALNs. The results taken, and the test set used, depended on the particular aspect of the ALN receiver under examination.

3.4.1 ALN Receiver Performance Considerations

ALN receivers were assessed primarily on the error rate achieved under test with the complete set of input signals used to train the ALN. In neural network research it is not generally considered good practice to test performance using the training set. Testing is usually done with a separate data set taken from the problem space and independent of the training set. In this way the “generalisation” ability of a network to classify previously unseen patterns, according to their similarity to those that it has learnt may be examined, rather than its memorisation ability alone. However, in certain situations such as the XOR and parity problems, there is a need to classify very similar input patterns separately. This is the case here since the modulation format employed, and in the case of the systems using OOCs, the sparse nature of the CDMA code sequences, mean that composite sequences corresponding to a desired data ‘1’ may be very similar to those corresponding to a data ‘0’, as shown in Figure 3.6. Given the assumption that the only performance degradation is due to MAI, generalisation is therefore likely to be counter-productive in such cases, since performance may only be improved by classifying similar sequences as corresponding to different output values. Generalisation would be a relevant issue for ALNs trained on purely MAI dominated composite sequences and then tested using composite sequences corrupted with additive noise. However, since interference other than MAI has been neglected, this issue is not considered further.

In order to put the results into perspective they were compared with those obtained by simulating both simple and hard-limited correlators and testing them on the same data set. Simulation of the neuromorphic network receiver based on OOCs was not considered since it has been shown to give the same results as the hard-limited correlator [Vecchi, 1988]. ALN receivers may be further assessed by analysis of the set of input composite sequences supplied to them, since this allows comparison of the results obtained with the
Figure 3.6. An example of very similar composite signals with different desired outputs. In (a) the desired output is a ‘1’ while in (b) the desired output is a ‘0’. The correlation receivers will classify both signals as a data ‘1’ since they only consider the first 2 tap positions and sequence (b) mimics the desired ‘1’ sequence in this respect. For the case shown, the sequences may be distinguished if other tap positions are considered as proposed using ALNs. However, the sequences only differ at one chip position and therefore generalisation would tend to classify them as corresponding to the same targets.

fundamental limits imposed on performance by the amount of available information present in the data.

Although the final ALN error rate was used as the main performance criterion, consideration was also given to the size of trained trees and the training time required. However, these were deemed to be of secondary importance. The ALN training algorithm
used makes no attempt to minimise tree size and in any case it is likely that a trained tree
could be produced in an integrated form by synthesis on a programmable gate array
[Armstrong, 1991b]. If adaptation is done off-line then long training times also need not
necessarily present a problem. Furthermore, the possibility for parsimonious evaluation
of simulated ALNs means that training times are reduced compared to those of other
neural network techniques [Armstrong, 1991a].

3.4.2 Evaluation of Training Parameters

The effects of input encoding and initial tree size on ALN error rates were investigated in
order to establish rules for their selection in efficient ALN training. To this end, the
average error rate was calculated for each set of fifty trees trained with the same
combination of these training parameters. This allowed for the differences in initial tree
configuration caused by the random assignment of node functions and input connections
and differing pseudorandom orders of pattern presentation during training. The effect of
both input encoding and initial tree size was evaluated by comparing the average error
rate to the best and worst case possible error rates. The former was derived by analysis of
the data set, while the latter was defined by the worst performance achieved by any
simulated receiver for the particular system configuration under consideration. Their
effect on final tree size and training times was also investigated.

3.4.3 Investigation of Trained ALNs

A trained and folded ALN results in a tree of AND and OR gates and thus, in theory, rules
could be extracted from this structure to explain its operation. However, the final tree is
likely to be large since no attempt is made to minimise its size during training.
Furthermore, different trees may achieve the same function in different ways depending
on their initial configurations and training orders, and any rules obtained would have to
be interpreted in terms of the input encoding mechanism. Therefore, in practice,
elicitation of decision rules is not viable. Nevertheless, by treating an ALN as a “black
box” it is still possible to gain an insight into its operation by examination of the signals
present at its inputs and outputs. Since it is now only necessary to consider the effect of
particular types of input sequences, this was done with a test set consisting of the unique
sequences contained in the complete test set. Although the users’ data values and the
timeshifts of interfering users are modelled as equally likely, the OOK modulation format
and sparse nature of OOCs mean that the distribution of unique sequences in the
complete test set is highly non-uniform. Wide variations occur in the distribution of
unique sequences in the set due to different combinations of sequences forming the same
composite and the high prevalence of zero valued chips. As such, the number of unique
sequences contained in a data set may be much smaller than the set size.
The ALN and correlation receivers were tested on this unique test set and the particular types of sequences causing errors at the output of each receiver were noted. Comparison of these sequences allows general deductions to be made about how ALNs make decisions based on their structure and target values since the configuration of the each type of receiver is different. Utilisation of temporal information may be investigated since the correlation receivers only use the two taps corresponding to their desired sequence, whereas an ALN may examine all the chips of a composite sequence. More specifically, it may be analysed in isolation by comparing hard-limiting receivers. Similar studies may be carried out for intensity information; comparison of hard-limiting and non-hard-limiting receivers can be used to show whether this information is necessary and to what extent it is used when retained.

3.5 Summary

In this chapter a methodology has been developed to permit the simulation and evaluation of ALN and previously proposed CDMA receivers. As such, it provides the context for the results presented in chapters 4, 5 and 6. The following key areas have been discussed:

- The specification of a model of an optical CDMA channel;

- The development of data generation methods, based on the assumptions of the channel model, to permit simulation of the type of signals present at the input of receivers in an optical CDMA network;

- The configuration of the systems to be examined by simulation, including the numbers of users and the types of code sequences allocated to them;

- The ALN training procedure; encompassing the selection of ALN input encoding schemes, initial tree sizes, training times and data set size and composition;

- ALN testing; comprising the specification of performance criteria, the evaluation of training parameters and methods by which the operation of ALNs may be investigated.
Chapter 4 ALN Receivers in CDMA Systems employing Optical Orthogonal Codes

4.1 Introduction

The results of computer simulations are presented to enable a comparison of performance between the proposed receivers employing ALNs and the simple and hard-limited correlation receivers previously considered for use in incoherent optical CDMA systems. This chapter is concerned with systems employing the OOCs described in chapter 3 and attention is focused on receivers synchronised to the ‘1100000000000000’ phase of the desired user’s code. The first two sections cover the simulation results obtained for the simple and hard-limited correlation receivers respectively. Each section contains a description of the receiver configuration, the error rates obtained and a discussion of their performance. These results allow comparison not only of theory and previous results with simulation but also provide a means of examining the behaviour of receivers in the context of the information available. Furthermore, they provide a reference for the ALN receiver simulations described in the next section. A number of possibilities are investigated with a view to improving performance including manipulation of intensity information and the use of additional taps. Details of configuration, error rates and a discussion of results are provided for each case. Results are given in the context of the limitations and assumptions covered in chapter 3 and in the case of the ALNs are necessarily empirical since, in general, there exists no easy way of extracting from the ALN exactly what it is doing. Indeed, as will be demonstrated, different ALN configurations may exist that would achieve the same results. However, a general appreciation of ALN behaviour may be obtained by examining the structure and distribution of sequences causing errors in the respective receivers.

4.2 The Simple Correlator

4.2.1 Configuration

The configuration of the simple correlator is shown in Figure 4.1(a). The fibre-optic delay-line is tapped according to the code to be received; in this case ‘1100000000000000’ and therefore having 2 taps. The outputs of the taps are summed and then presented to a threshold at the synchronisation time. If this threshold, shown in Figure 4.1(b), is matched or exceeded then the receiver decides that the transmitter has
sent a data ‘1’, whereas if the summed output of the delay-line falls below this threshold the receiver decides that the transmitter has sent a data ‘0’.

4.2.2 Simulation Results

The simulation results for the simple correlator are presented in Table 4.1. Testing was carried out using the complete test set described in chapter 3. The error rate obtained agrees exactly with that expected from equation (2.6) for a simple correlator operating under chip synchronous conditions (N = 3, Th = K = 2 and F = 16).

<table>
<thead>
<tr>
<th>Number of Errors</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Set Size</td>
<td>8192</td>
</tr>
<tr>
<td>Error Rate</td>
<td>7.81 x 10^{-3}</td>
</tr>
</tbody>
</table>

Table 4.1. Test results for the simple correlator simulation.
4.2.3 Error Sequence Analysis

The types of sequences causing errors at the output of the simple correlator are shown in Table 4.2. To determine the circumstances under which errors occur it is only necessary to examine the first 2 chip positions of each composite sequence, since at the synchronisation time the simple correlation receiver considers only these chips in deciding whether the received sequence constitutes a data ‘1’ or ‘0’; the other chip values are shown thus ‘*’, indicating “don’t care” states. From the table it is seen that errors only occur when the desired output is a ‘0’ and the actual output is a ‘1’, thus interfering users are contributing chip values when the desired user is sending a data ‘0’, i.e. no sequence, in ways that cause the correlator output threshold to be exceeded.

<table>
<thead>
<tr>
<th>Seq. Type</th>
<th>Generic Composite Causing Error</th>
<th>Actual Output</th>
<th>Target Output</th>
<th>Freq. with Target = 0</th>
<th>Freq. with Target = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>20***************</td>
<td>1</td>
<td>0</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>02**</td>
<td>1</td>
<td>0</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>11***************</td>
<td>1</td>
<td>0</td>
<td>32</td>
<td>3136</td>
</tr>
</tbody>
</table>

Table 4.2. The unique families of composite sequences causing errors at the output of the simple correlator. Reference sequence: ‘110000000000000000’.

Errors arise in two distinct ways. In the first, sequences of type A and type B cause errors when the interfering users’ chips occupy the same tap position such that their combined intensity achieves the output threshold. In the second, sequences of type C cause errors when the interfering users contribute to separate tap positions effectively mimicking the desired data ‘1’ code sequence, again causing the threshold to be attained.

Further information concerning simple correlator performance may be obtained by examining the types of input composite sequence causing the receiver output threshold to be exceeded and their distribution in the data set when paired with their associated target output values. The frequencies of the input sequences paired with targets of ‘0’ are shown in column 5 of Table 4.2 recalling the fact that errors occur only when the actual output is a ‘1’ and the target is a ‘0’. Column 6 of Table 4.2 lists the frequencies of the same composite sequences, however, in this case the values are associated with target pairings of ‘1’. Comparison of the corresponding values in the two columns indicates that there are conflicts in the data set that require the same composite sequence to be classified as both a ‘1’ and a ‘0’ for sequences of type C. This means that any receiver will make errors given the data set used and the restrictions described in chapter 3. However, in this case the sequence type is correctly classified so as to minimise error by assigning it to correspond to the input-target pair of the highest frequency. Nevertheless, the simple
The correlator can be considered suboptimal since there exist cases when there is no conflict in the data set yet the composite sequence is wrongly classified, despite the fact that it is of a unique type. This implies that there is information present that could be used to perform a correct classification and so reduce the error rate. Examples of such cases are the sequences of types A and B in Table 4.2 where no pairings with the target value of 1 exist.

### 4.2.4 Extended Error Analysis

The analysis of the sequences causing errors at the output of the simple correlator is now extended to cover the entire composite length of 16 chips, as shown in Table 4.3. As a result, the generic sequences of Table 4.2 become unique in type since the data set includes all possible composite sequences. Although the simple correlator is unable to use this extra information, it is pertinent to ascertain whether consideration of further taps provides potential for a reduction in error rate via another reception technique, namely by use of an ALN. Furthermore, by examining the data set for sequences with conflicting targets which give rise to errors in any receiver a fundamental limit on performance may be established.

Table 4.3 indicates that, from the data set point of view of 16 taps, sequences other than of the type '20***************' and '02***************' (patterns 1 to 18) may be correctly classified. Composite sequences in the data set that mimic the desired user’s data sequence in the first 2 chip positions become distinguishable via the consideration of additional chips. Specifically, sequences 19 to 24 are uniquely associated with a target of ‘0’ and thus may now be considered correctable. The remainder of the sequences occur with conflicting target values and will always give rise to errors. The best possible error rate that may achieved, considering the data set over the full 16 taps, is limited by the 9 conflicts of sequences 25 to 33 and their classification as corresponding to the higher frequency target so as to minimise this rate. This means classification of sequence 25 with a target value of ‘0’, sequence 26 with either target value and the remaining conflicting sequences, i.e. 27 to 33, with a target value of ‘1’. This results in the occurrence of 15 actual errors using the test set or a best possible error rate of 1.83 x 10⁻³.
Table 4.3. The unique composite sequences causing errors at the output of the simple correlator synchronised to the '1100000000000000' phase of the desired OOC and considered over 16 chips.
4.3 The Hard-limited Correlator

4.3.1 Configuration

The configuration of the hard-limited correlator is shown in Figure 4.2(a). The structure consists of a non-linear optical processing element, characterised by $f(x)$, followed by the simple correlator arrangement previously described in section 4.2.1. The non-linear element clips back (normalised) chip intensity values $\geq 1$ to one and those $< 1$ are set to zero. The transfer characteristic of the input threshold element is shown in Figure 4.2(b).

4.3.2 Simulation Results

The hard-limited correlation receiver simulator was also tested with the complete data set generated using method 'B', described in chapter 3. The results are shown in Table 4.4.
and indicate that the performance is improved by a factor of 2 with respect to the simple correlator for the code weight used. The simulation error rate is in agreement with that derived by [Azizoglu, 1992] for a hard-limited correlator operating under chip synchronous conditions i.e. equation (2.10) with $K = 2$, $N = 3$ and $F = 16$.

<table>
<thead>
<tr>
<th>Number of Errors</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Set Size</td>
<td>8192</td>
</tr>
<tr>
<td>Error Rate</td>
<td>$3.91 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

Table 4.4. Test results for the hard-limited correlator simulation.

4.3.3 Error Sequence Analysis

Table 4.5 shows the single type of sequence causing errors at the output of the hard-limited correlator. The distributions shown are those in the transformed training set, i.e. after hard-limiting. Again, as was the case with the simple correlator, only the first 2 chips need to be considered and errors still only occur when the desired output is a '0' and the actual output is a '1'.

<table>
<thead>
<tr>
<th>Seq. Type</th>
<th>Generic Composite Causing Error</th>
<th>Actual Output</th>
<th>Target Output</th>
<th>Freq. with Target = 0</th>
<th>Freq. with Target = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>11**************************</td>
<td>1</td>
<td>0</td>
<td>32</td>
<td>4096</td>
</tr>
</tbody>
</table>

Table 4.5. The type of sequence causing errors at the output of the hard-limited correlator synchronised to the '1100000000000000*' phase of the desired code. Frequencies given after hard-limiting.

It is apparent that errors are characterised by sequences of type C from Table 4.2, with the increase in the frequency of composites of target value '1' in Table 4.4 being caused by the mapping of multi-level sequences of that target value to binary sequences of the type C. Errors now only occur when the chips of interfering users overlap on separate taps to mimic the desired data '1' and no longer because intensity is allowed to sum on a single tap. The hard-limiting operation ensures that the effective interference is reduced to that of a single user per tap (for $\lambda = 1$). This means that, in the case of sequences of types A and B:

(A) $20************** \rightarrow 10**************,$

(B) $02************** \rightarrow 01**************,$

and the output threshold is no longer achieved. This behaviour is in accordance with that expected and the receiver can be considered optimal, given the number of chips being
examined and the transformation imposed on the input data. Errors are anticipated due to the existence of conflicts in the data set, however the rate is minimised by assigning the input sequence to the target with the corresponding higher frequency.

4.3.4 Extended Error Analysis

As with the simple correlator, the analysis of the composite sequences causing errors will now be extended to encompass the full 16 chips of the desired OOC with a view to establishing the performance bounds of an ALN examining this type of input. However, in this case the effect of the hard-limiting operation must be taken into account since it is the transformed data set that is presented to the correlator section, and therefore ultimately to an ALN, when considering the purely temporal information provided by this method of input encoding.

<table>
<thead>
<tr>
<th>Seq. Type</th>
<th>Unique Composite Causing Error</th>
<th>Actual Output</th>
<th>Target Output</th>
<th>Freq. with Target = 0</th>
<th>Freq. with Target = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1101000000000000</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>130</td>
</tr>
<tr>
<td>2</td>
<td>110100000000100</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>110100000000001</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>1100100000000000</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>1100100000000010</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>1110000000000010</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>1110100000000000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>72</td>
</tr>
<tr>
<td>8</td>
<td>1110000000000000</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>96</td>
</tr>
<tr>
<td>9</td>
<td>110000000000101</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>72</td>
</tr>
<tr>
<td>10</td>
<td>1100000000000001</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>96</td>
</tr>
<tr>
<td>11</td>
<td>1100000000000010</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>130</td>
</tr>
<tr>
<td>12</td>
<td>1100000000000100</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>13</td>
<td>1100000000000000</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>992</td>
</tr>
</tbody>
</table>

Table 4.6. Sequences causing errors after hard-limiting considered over 16 chips. Again the desired OOC is ‘1100000000000000’.

It is clear from Table 4.6 that examining the composite sequences over 16 chips does not permit further improvement in error performance over that possible using 2 taps. This is because all the sequences listed above have conflicting target values and the best that can be done is to assign the sequences to the higher frequency target value of ‘1’ which would result in a performance identical to that of the hard-limited correlator. In order to understand what is happening it is necessary to consider the non-linear transformation performed on the data set by the hard-limiting operation. As in the case of the hard-
limited correlator, the '20************' and '02************' sequences are mapped to new sequences but these are still paired only with targets of '0' and thus remain correctable. The operation also causes a reduction in the number of unique sequences in the transformed data set by mapping the multi-level composite sequences to boolean vectors. Therefore, as far as the correlator or ALN stage of the receiver is concerned, certain sequences cease to exist. For example, sequences 19 and 26 in Table 4.3 are mapped to sequences 1 and 11 in Table 4.6. Furthermore, this process also creates input-target pairings that previously did not exist. This is important since unique sequences in the data set with targets of '1' and no corresponding target of '0' are mapped to sequences which previously only had target values of '0', e.g. sequences 19 to 24 in Table 4.3 now have target '1' pairs as shown by the sequences 1 to 5 in Table 4.6. Taking the sequence '310100000000000001' as an example, it is obvious that it could only be paired with a target of '1' in the data set since 2 interfering users of equal power cannot produce a chip intensity of 3. However, clipping creates '110100000000000000' (sequence 1 in Table 4.6) which previously only had a target value of '0'. This is the mechanism by which conflicts are created; there are only nine unique sequences in Table 4.3 with conflicting targets, hard-limiting increases this to the thirteen shown in Table 4.6.

Considering only 2 taps, it has been shown that the hard-limiting correlator offers improved performance over the simple correlator. This is because the number of errors corrected by clipping outweigh the number of data set conflicts generated by this operation. It is interesting to note that a 2-tap receiver able to make use of intensity data could also only achieve the same performance as a hard-limited correlator since it would still be limited by the sequences of type C in Table 4.2. However, when considered over the full composite length, hard-limiting the input data constrains the potential performance to a lower level than that possible by consideration of the raw data set, the reason being that useful information is being discarded before it can be used. From Table 4.6 the best error rate possible for the set under hard-limiting can be calculated to be 3.91 x 10^{-3}.

4.4 ALN Receivers

4.4.1 ALN Configuration

The simple and hard-limiting correlators are limited to a number of tap positions equivalent to the weight of the code to be received by design. Decisions on how to classify an input sequence are based on knowledge of the structure of the code to be received alone and no account is taken of the codes of the other system users. The output
of the correlation stage is compared to the decision threshold after the last non-zero chip position in the desired code sequence and interference minimisation is achieved through the correlation properties of the codes.

The aim of using ALN enhanced receivers is that they should learn details of both the desired and interfering codes, based on intensity and temporal data contained within the input composite sequences presented, and use this structural information to make a better decision and hence improve performance. In order to achieve this, it is clear that the ALN must be able to examine more taps than previously considered, since the analysis of the
data set over just 2 taps has shown that any improvement is constrained to be equivalent to that of the hard-limited correlator, even if intensity information is used; although some knowledge of other users is provided via intensity it is not sufficient to lower the error rate further. Therefore, the ALNs were allowed to examine the input sequence over the full length of an OOC. In this way the performance of ALN receivers may be assessed using the analysis of the composite sequences in the data set causing errors in the correlator receivers, as documented in sections 4.2.4 and 4.3.4. This analysis showed that the error rate need only be limited by the conflicts in the data set and that performance improvements were potentially possible over those attained using the correlation receivers, though how this could be achieved was not specified.

The ALN configuration is shown in Figure 4.3. Taps were placed at each chip interval regardless of whether the desired sequence occupied these chip positions or not. As in the case of the correlation receivers, the ALN was assumed synchronised to the ‘1100000000000000’ phase of the desired OOC and thus the ALN can be thought of as using the same two taps as the correlation receivers plus 14 other succeeding chips or “future” values of the composite sequence. The taps feed the input intensity encoding mechanism which creates a boolean vector suitable for input to the ALN by concatenating the individual encoder outputs representing each composite chip.

4.4.2 ALN Simulation Results

The range of results obtained for the ALNs, trained using the combinations of input encoding schemes and initial tree sizes described in Chapter 3, are presented in Table 4.7 in terms of the best case, worst case and average error rate for each set of fifty trees. Testing was performed with the complete test set used in evaluation of the correlation receivers. Comparison of the worst case error rates with the results obtained using a hard-limited correlator (Table 4.3) shows that the performance is always at least as good as the previous best case receiver. Moreover, the figures for best case error rates, given in Table 4.7, show that with the exception of ALNs using hard-limited input, significant performance improvements can be made over the previously proposed receivers.

4.4.3 ALN Training Parameters

The figures for best and worst case error rates by themselves cannot be used to provide guidelines on how to achieve ALNs with low error rates efficiently because they give no indication how frequently each is achieved for a given set of training parameters. Therefore, in order to identify the effect of tree size and encoding scheme on performance the average error rate was used, since it reflects whether a given set of training parameters produces a predominance of trees attaining either worst case, best case or intermediate
<table>
<thead>
<tr>
<th>Input Encoding Scheme</th>
<th>Initial Tree Size (Leaves)</th>
<th>Average Final Tree Size (Leaves)</th>
<th>Best Case Error Rate (Single Tree)</th>
<th>Worst Case Error Rate (Single Tree)</th>
<th>Average Error Rate (50 Trees)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bar Code</strong></td>
<td>4096</td>
<td>174</td>
<td>$2.93 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.67 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>8192</td>
<td>317</td>
<td>$2.44 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.33 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>16384</td>
<td>507</td>
<td>$2.20 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.25 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>32768</td>
<td>852</td>
<td>$1.95 \times 10^{-3}$</td>
<td>$3.42 \times 10^{-3}$</td>
<td>$2.82 \times 10^{-3}$</td>
</tr>
<tr>
<td><strong>Tempre. Code</strong></td>
<td>4096</td>
<td>154</td>
<td>$2.44 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.61 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>8192</td>
<td>261</td>
<td>$2.44 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.55 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>16384</td>
<td>450</td>
<td>$2.44 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.35 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>32768</td>
<td>667</td>
<td>$1.95 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.03 \times 10^{-3}$</td>
</tr>
<tr>
<td><strong>Binary Code</strong></td>
<td>4096</td>
<td>244</td>
<td>$2.44 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.28 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>8192</td>
<td>340</td>
<td>$1.83 \times 10^{-3}$</td>
<td>$3.42 \times 10^{-3}$</td>
<td>$3.03 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>16384</td>
<td>549</td>
<td>$1.95 \times 10^{-3}$</td>
<td>$3.42 \times 10^{-3}$</td>
<td>$2.96 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>32768</td>
<td>1024</td>
<td>$1.95 \times 10^{-3}$</td>
<td>$3.42 \times 10^{-3}$</td>
<td>$2.58 \times 10^{-3}$</td>
</tr>
<tr>
<td><strong>Scatter Code 1 (d = 1)</strong></td>
<td>4096</td>
<td>158</td>
<td>$2.44 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.63 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>8192</td>
<td>281</td>
<td>$1.95 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.39 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>16384</td>
<td>450</td>
<td>$2.44 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.32 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>32768</td>
<td>767</td>
<td>$1.95 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.07 \times 10^{-3}$</td>
</tr>
<tr>
<td><strong>Scatter Code 2 (d = 2)</strong></td>
<td>4096</td>
<td>301</td>
<td>$2.44 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.29 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>8192</td>
<td>408</td>
<td>$1.95 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$2.93 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>16384</td>
<td>673</td>
<td>$1.95 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$2.70 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>32768</td>
<td>1141</td>
<td>$1.83 \times 10^{-3}$</td>
<td>$3.42 \times 10^{-3}$</td>
<td>$2.27 \times 10^{-3}$</td>
</tr>
<tr>
<td><strong>Hard-limited Input</strong></td>
<td>4096</td>
<td>134</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>8192</td>
<td>227</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>16384</td>
<td>283</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>32768</td>
<td>632</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
<td>$3.91 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

Table 4.7. Performance of ALNs trained using a variety of initial tree sizes and encoding schemes.

The average error rates given in Table 4.7 are shown plotted in Figure 4.4 with the lowest possible error rate achievable following the analysis of section 4.2.4 provided as a reference.

The results obtained using hard-limiting of composite sequences represent the best possible error rates that can be expected, based on the analysis of the data set under transformation given in section 4.3.4. Performance is constant over the range of initial
Figure 4.4. Dependence of performance on training parameters.
tree sizes considered, being improved beyond that of the simple correlator to a level identical to that of the hard-limited correlator. This confirms, that provision of extra temporal data only, is not sufficient to reduce the error rate further; information that could be used to attain a better error rate is discarded before the ALN can use it. The average error rate for the hard-limited ALN shown in Figure 4.4 can also be used as a reference, since it represents the previous best case receiver performance achieved by the hard-limited correlator and the worst case error rate achieved by any ALN.

Examination of the structure of individual ALNs folded after training revealed that the most efficient tree produced using hard-limited input was a single AND gate connected to the first 2 chip positions of the clipped input vector. This corresponds to the optical AND gate model of the hard-limited correlator described in chapter 2. However, as can be seen from the figures for average final tree size in Table 4.7, the trees produced were usually more complex. This is a result of the large initial tree sizes used and reflects a more general trend for ALNs of larger initial size to form larger final folded trees. This demonstrates that an ALN may realise a desired function in a number of ways and reflects the fact that the learning algorithm makes no attempt to minimise the tree structure produced i.e. perform boolean minimisation.

Figure 4.4 indicates that the remaining encoding schemes, which preserve intensity information, have the potential to improve error performance beyond that previously achieved for all tree sizes. In general the improvement can be seen to increase with initial tree size; the degree of change depending on the particular input encoding used. This is not surprising since larger trees should be capable of implementing more complex functions better suited for adaptation to a task, and once the necessary degree of complexity is reached, additional increases in tree size provide more ways of realising a desired function with the result that the proportion of trees achieving lower error rates should increase for a given number of epochs. Furthermore, the likelihood of a tree failing to realise a desired function due to unsuitable connections to the input variables is reduced with increasing tree size.

The plots obtained using the temperature code and the two scatter codes allow direct comparison of the effect of hamming distance on ALN performance since they are all of the same length. Any given tree size therefore has the same number of connections to the input vector elements and, as such, each has the potential to realise an equally complex function. Both the temperature code and scatter code 1 have a hamming distance of 1 between adjacent encodings, whereas scatter code 2 has a constant value of 2 between all encodings. ALNs employing scatter code 2 are seen to consistently attain a higher level of performance than those using either the temperature code or scatter code 1. From Table 4.7 it can also be seen that, despite the same encoding length and initial tree size,
the average folded tree size produced using scatter code 2 is always much larger than that obtained using the temperature code or scatter code 1. This suggests that the larger distance enables creation of a more complex function from the same starting conditions to produce a lower error rate by affecting more nodes during adaptation. It should also be noted that scatter code 1 and the temperature code exhibit very similar results in terms of both average error rates and final folded tree sizes. Since the hamming distances between all members of their respective distinct codes are the same in both cases (see Figure 3.5 (b), (d)), it appears that the additional increasing weight property of the temperature code has little effect. Since the sparse nature of the OOCs means that the higher composite chip values, which would be encoded using higher weight codes, are less likely to occur, it would appear that there are not enough high chip values in the test set for this property to become pronounced.

The bar code and scatter code 2 can be used to study the effect of input encoding length since both have constant hamming distances of 2 between all input encodings. Figure 4.4 shows that ALNs using scatter code 2 of length 3 consistently outperform those using the bar code of length 4. Table 4.7 shows that this is achieved using a more complex function of fewer input variables since the final folded tree sizes produced using scatter code 2 are greater than those obtained using the bar code. In fact, the results achieved using the bar code are comparable to those attained using the temperature code and scatter code 1, which have a smaller distance property but shorter lengths, for all but the largest initial tree size.

The binary code and scatter code 2 also achieve comparable performance for small tree sizes where it appears that the higher absolute hamming distance of scatter code 2 is offset by the shorter length of the binary code. However, as initial tree size increases, then ALNs using scatter code 2 improve more rapidly than those using binary encoding, despite maintaining similar final tree sizes. This seems to indicate that hamming distance becomes dominant once a certain complexity of function has been achieved and that merely adding more connections per input vector bit to an encoding scheme of lower distance is less effective.

ALNs using binary input encoding also outperform those using the bar code. However, this is due to the binary code of length 2 having twice as many connections per input vector than the length 4 bar code. If the encoding length is neglected and performance considered purely on a connections per encoded input vector bit basis, bar encoding with 32768 leaves now outperforms binary encoding with 16384 leaves. This again suggests the dominance of hamming distance, once the number of connections per input bit reaches a particular level.
To summarise encoding effects, the best case error rate trees were more frequently achieved using a high hamming distance for a given encoding length i.e. the most changes possible per encoding length given the number of users to be supported. The larger hamming values put more distance between encoded vectors at the ALN input. The greater the change at the input, the more nodes in a given tree affected during adaptation and thus the more opportunity for nodes to change to implement a function suitable for distinguishing similar unencoded vectors, as required in this case. The use of a low encoding length means that each bit of the encoded input vector is connected to more nodes in an ALN of a given size compared to an ALN employing a higher length input encoding. Therefore, a change in a single bit of the input vector has the opportunity to affect more nodes; it effectively allows the implementation of a more complex function for the same size tree, since it has to consider fewer input variables. This is reinforced by the fact that the more successful encoding schemes result in larger folded trees for the same initial tree size.

4.4.4 ALN Learning Behaviour

Examination of the final states of individual trees after training permits a more detailed study of ALN learning behaviour. The proportion of trees in each set making a given number of actual errors is shown in Figures 4.5 to 4.9 for each intensity preserving encoding scheme and over the range of tree sizes examined. Achievement of 15 actual errors corresponds to the best possible performance given the conflicts in the data set and of 32 errors a level equivalent to that of the hard-limited correlator. The general capability of increasing initial tree size to produce more trees making fewer actual errors is observed, as reflected by the average error rates of Figure 4.4, with the degree of shift to trees of lower error rates depending on the efficacy of the encoding scheme.

A discrete distribution in error rates from best to worst case is expected due to the discrete frequencies of sequences in the test set. Nevertheless, intermediate numbers of errors between those achieved would be expected to occur, if sequences were equally likely to be corrected in any particular combination. However, this is not observed. From Figures 4.5 to 4.9 it is seen that occurrences of error rates corresponding to between 20–28 and 28–32 actual errors are extremely rare. The same phenomenon is also observed, though to a lesser extent, due to the low numbers of near best case trees produced, between 16 and 20 actual errors as shown in Figure 4.9(d). Therefore, the correction of composite sequences seems to occur in distinct groups. Furthermore, correction of these groups to achieve lower error rates is seen to become progressively more difficult. This is particularly pronounced using the bar code, temperature code and scatter code 1, Figures 4.5, 4.6 and 4.8, where achieving a level of 28 errors is relatively easy, but groups corresponding to 20 errors always have a lower relative frequency while those associated
Figure 4.5. Performance distribution of trained trees for the bar code.
Figure 4.6. Performance distribution of trained trees for the temperature code.
Figure 4.7. Performance distribution of trained trees for the binary code.

(a) 4096 Initial leaves

(b) 8192 Initial leaves

(c) 16384 Initial leaves

(d) 32768 Initial leaves
Figure 4.8. Performance distribution of trained trees for scatter code 1 ($d = 1$).
Figure 4.9. Performance distribution of trained trees for scatter code 2 (d = 2).
with 16 errors are lower still for all tree sizes. The lowest possible number of errors was never achieved using these encoding schemes. Trees using binary input encoding, Figure 4.7, show a more even shift with increasing tree size up to the level of 20 actual errors and a similar difficulty in achieving lower error rates. Only trees using scatter code 2, shown in Figure 4.9, demonstrate a smooth transition between groupings to achieve a small number of examples of the lowest possible error rate trees for initial sizes of 32768 leaves.

A study of the sequences corrected with respect to the simple correlator to obtain the actual number of discrete errors described above shows that there was indeed a tendency for sequences to be learnt as groups, as shown in Table 4.8. This appears to be based on the similarity of the input sequences corrected in each group, as demonstrated by groups 3 and 4 in the table. Reasons for the progressive difficulty of achieving groups of lower rates may be found by examination of the sequences corrected and their frequencies in the test set. In all cases, the ALN corrects sequences causing errors with correlation receivers by re-classifying them as corresponding to the desired output of ‘0’.

<table>
<thead>
<tr>
<th>Group</th>
<th>Error Rate</th>
<th>Actual Errors</th>
<th>Sequences Corrected w.r.t. Simple Correlator</th>
<th>Freq. with Target = 0</th>
<th>Freq. with Target = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.91 x 10^{-3}</td>
<td>32</td>
<td>20************** 02**************</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3.42 x 10^{-3}</td>
<td>28</td>
<td>110200000000000000</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2.44 x 10^{-3}</td>
<td>20</td>
<td>110100000000000000 1101000000000100 1101000000000001</td>
<td>4 2 2</td>
<td>0 0 0</td>
</tr>
<tr>
<td>4</td>
<td>1.95 x 10^{-3}</td>
<td>16</td>
<td>110010000000000010 1100100000000000 1110000000000010</td>
<td>2 2 2</td>
<td>0 0 0</td>
</tr>
<tr>
<td>5</td>
<td>1.83 x 10^{-3}</td>
<td>15</td>
<td>111000000000000010</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.8. The predominant error rates achieved by individual trees. Reference sequence: ‘1100000000000000’.

The sequences corrected to achieve the performance of group 1 correspond to those also corrected by the hard-limited correlator, although in this case the input encoding does not create conflicts in the data and therefore further learning is not precluded. The relative ease of learning these sequences may be explained by the fact that in all cases sequences of the type ‘20…’ or ‘02…’ are uniquely paired with targets of ‘0’ and are either nearer, using the bar code, binary code and scatter code 2, to the encoded all-zero vector or equidistant, using the temperature code and scatter code 1, from the encoded all-zero and
data '1' vectors in hamming space. Therefore, a classification based on hamming distance is likely to be sufficient to permit their correction. Furthermore, any insensitivity property of the tree acting on the encoded bits of the first two chip positions is likely to help by placing them into either a '20**************' or '02**************' grouping.

Looking at the sequences pertaining to groups 2 to 5 as a whole, it is seen that they all have the characteristic '11...' signature at the first two chip positions, mimicking the desired data '1'. As such, for all encoding schemes the sequence to be corrected is always closer in hamming space to the desired data '1' sequence than the data '0' sequence; differences at chip positions other than the first two do not matter since both data '1' and data '0' are of the form '***00000000000000' and these will add the same constant to each absolute distance. This is analogous to the hard-learning problems characterised by the XOR and parity functions mentioned in chapter 3. Moreover, in this case, any insensitivity properties acting on the first two chips will group them with targets of '1' since sequences beginning '11...' are statistically more likely to correspond to a data '1'. This explains why correction beyond group 1 is difficult. Furthermore, evidence from Table 4.8 suggests that the low and decreasing frequencies of sequences, over groups 2-5, may account for the increasing difficulty of sequence correction.

The predominance of trees achieving a performance level corresponding to group 2 using the temperature code and scatter code 1, Figures 4.6 and 4.8, may be attributed to the relatively higher distance between the '0' and '2' states in the encoding hamming maps given in chapter 3, Figure 3.5(b) and (d). Under these conditions, the '1102...' sequence is corrected in preference to those in groups 3 to 5 which must be differentiated by the lower distance between the '0' and '1' states. The particular difficulty in attaining the lowest error rate of band 5 is to be expected since the sequence has conflicting targets whose frequencies only differ by one occurrence.

4.4.5 Performance Assessment

For an application it is obvious that the ALN providing the lowest error rate should be selected. The study of the effect of the training parameters and learning behaviour has shown that only the larger trees are capable of forming the functions required in order to learn the sequences characterised by '11*************' and requiring an output of '0' as "exceptions" to the general rule that this signifies a desired data '1'. This is achieved, based on information contained in the structure of the input composite sequences, by considering additional intensity and temporal data to provide a context for classification. This is made difficult by the sparse nature of the OOCs. For example looking at table 4.8, it can be seen that composites requiring classification as data '0', differ from the desired data '1' sequence at a maximum of 2 chip positions (groups 2-5). However, one of these
chips always derives from a succeeding sequence. As such, whilst it may provide positional information about the preceding sequence it does not say anything about its data value, since consecutive data bits are independent. Therefore, decisions as to the nature of the received bit may have to be made on the basis of as little as one additional chip position. The most successful trees use an efficient high distance per length encoding scheme which makes learning easier by affecting more nodes for a given initial size.

A comparison of the best case error rate performance under worst case interference conditions for all receivers is given in Table 4.9. The best case ALN performance reduces the error rate by a factor of 4, when compared to the simple correlator receiver and by a factor of 2, with respect to the hard-limited correlator — the previous lowest error rate receiver. The best case ALN error rate also represents the lowest possible error rate in the context of the available data; composites being correctly classified with their target output values in the absence of conflicts and with the higher frequency target pairing in the case of conflicting target values.

<table>
<thead>
<tr>
<th>Receiver</th>
<th>Simple Correlator</th>
<th>Hard-limited Correlator</th>
<th>Best Case ALN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Rate</td>
<td>7.81 x 10^{-3}</td>
<td>3.91 x 10^{-3}</td>
<td>1.83 x 10^{-3}</td>
</tr>
</tbody>
</table>

Table 4.9. Performance comparison of best case ALN with the previously proposed receivers.

It is important to note that low error rates are not precluded by small initial tree size, as shown by the best case entries in Table 4.7, it is just that they are rare. For example, using binary input encoding with an initial tree size of 8192 leaves, produced one tree out of the set of fifty capable of a best possible classification. This, as previously mentioned, is a result of the ALN learning algorithm making no attempt to minimise the size of a trained tree and the greater probability of a smaller tree’s performance being limited by its initial configuration of connections to the input. If efficiency of implementation is also to be considered as a performance criterion, then the observed possibility of producing small, low error rate trees provides scope for further investigation. The most obvious way would be either to modify the learning algorithm or to develop a new one. However, given the algorithm used there are still a number of options. A study of the behaviour of trees with small initial size found that the relatively high error rates achieved, predominantly corresponding to groups 1 and 2 in Table 4.8, were typically reached very early in the training period and that performance remains fixed for the rest of the training time. It may be that the groups of Table 4.8 (except group 5) correspond to local minima in which the ALN gets trapped due to its connection pattern and that either different or additional connections are required to enable the error rate to be reduced further. In the former case
more trials could be made in the hope of finding a connection pattern allowing further improvements. In the latter case there may be no alternative but to increase the tree size. Alternatively, the groups of Table 4.8 may correspond to plateaux in performance and further improvements are very slow due to the nature of the training set in that the number of unique, correctable errors is small and they are only sparsely distributed in the data set. In this case either different connection patterns could be investigated to see if any might lead to improvement within the training time given or the number of epochs allowed may simply be increased. Therefore, in selecting trees there is a trade-off between initial and hence final size and training time.

4.5 Summary

The error rates obtained by simulation of the simple and hard-limited correlators were shown to agree with those expected from theory and were used as benchmarks to assess ALN performance. Analysis of the types of sequences giving rise to errors in these receivers was used to look at the limits set on performance. Results of this analysis showed that:

- The lowest possible error rate achievable, considering only the taps associated with the desired user’s sequence, is identical to that obtained with the hard-limited correlator, regardless of whether intensity information is used;

- Errors arising through MAI effects at the input of correlation receivers may be eliminated by taking into account the presence of other users’ signals. As such, lower error rates than achieved with the previously proposed receivers are indeed possible;

- Provision of additional temporal data alone limits performance to that achieved by the hard-limited correlator. The clipping operation discards useful information before it can be used and creates conflicts in the data which preclude further reduction in error rate.

The results obtained by simulation of ALNs showed that:

- ALNs are capable of realising the performance limits set by the analysis of the input data;

- If the presence of other users’ signals is considered and intensity information is retained and used, ALNs are capable of correctly classifying inputs causing errors with the correlation receivers and significant reductions in error rate may be
achieved. However, complete elimination of MAI effects is not possible using the ALN configuration considered;

- For a given input encoding scheme, average error rate decreased with increasing initial tree size;

- For a given initial tree size, the lowest error rates were achieved using intensity-retaining input encoding schemes with a high hamming distance to length ratio;

- For a given initial tree size, lower error rates were associated with larger final tree sizes, reflecting the incorporation of greater knowledge regarding interfering users;

- There was a tendency for input sequences causing errors with the correlation receivers to be corrected in groups based on their similarity;

- The difficulty in attaining best case error rates was caused by the low frequency of inputs causing errors with the correlation receivers, the sparse nature of the codes and the choice of ALN input window.
Chapter 5 Enhanced ALN Receivers in CDMA Systems employing Optical Orthogonal Codes via Code Shift Selection

5.1 Introduction

In the previous chapter, consideration was given to the enhancement of reception in an optical CDMA environment via the use of ALNs. More specifically, the methods investigated involved the retention of chip intensity data and the supply of additional “future” temporal data by extending the number of taps considered with respect to the correlator structures previously proposed. Given both types of information, it was shown that an ALN could indeed learn details of the structure of interfering users’ code sequences and hence distinguish cases where combinations of such sequences had formerly given rise to errors. As such, it was shown that ALNs could be used to improve performance in an optical CDMA environment. In this chapter further attention is focused on the choice of taps considered by an ALN receiver and the phase of the desired sequence to which it is synchronised. In particular, receivers are considered to be synchronised to the ‘0000001100000000’ phase of the desired sequence. Again results are presented within the context of the limitations and assumptions given in chapter 3. Once more the results obtained for the ALNs are necessarily empirical.

The structure of the chapter also follows that of chapter 4, the first two sections being concerned with the simple and hard-limited correlators. Simulation results are reported for each and the sequences causing errors are identified. The analysis of these sequences is then extended to cover the entire composite sequence length using the method developed in the previous chapter. This provides the basis for both bounding and examining the performance of ALN receivers exposed to this form of input. In the following section simulation results and discussion are presented for ALN receivers using the same input encoding schemes and initial tree sizes as in chapter 4.

5.2 The Simple Correlator

5.2.1 Configuration

The simple correlator considers only the taps associated with the desired code sequence and therefore the configuration is as shown in Figure 4.1.
5.2.2 Simulation Results

Since the composite sequences are generated using the same interfering code sequences with only the desired code sequence contribution being shifted, the error rate achieved by simulation of this correlator configuration is expected to be identical to that obtained in chapter 4. The results presented in Table 5.1 confirm this and again agree with that expected from equation (2.6) with N = 3 and Th = K = 2.

<table>
<thead>
<tr>
<th>Errors</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. Patterns in Set</td>
<td>8192</td>
</tr>
<tr>
<td>Error Rate</td>
<td>7.81 x 10^{-3}</td>
</tr>
</tbody>
</table>

Table 5.1. Simulation error rate obtained for the simple correlator.

5.2.3 Error Sequence Analysis

Table 5.2 shows that errors occur for exactly the same reason as in chapter 4, i.e. whenever the output threshold is exceeded either via a summation of optical intensity on one of the correlator taps or via separate contributions to both taps. Changing the phase of the desired code to which the receiver is synchronised makes no difference as would be expected since the receiver is still only considering two taps. Once again the receiver may be considered suboptimal, since the ‘*******20*******’ and ‘*******02*******’ cases are uniquely associated with targets of zero and there exists a method of correcting these cases.

<table>
<thead>
<tr>
<th>Seq. Type</th>
<th>Generic Composite Causing Error</th>
<th>Actual Output</th>
<th>Target Output</th>
<th>Freq. with Target = 0</th>
<th>Freq. with Target = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td><em><strong><strong><strong>20</strong></strong></strong></em></td>
<td>1</td>
<td>0</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td><em><strong><strong><strong>02</strong></strong></strong></em></td>
<td>1</td>
<td>0</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td><em><strong><strong><strong>11</strong></strong></strong></em></td>
<td>1</td>
<td>0</td>
<td>32</td>
<td>3136</td>
</tr>
</tbody>
</table>

Table 5.2. The types of sequences giving rise to errors in the simple correlator simulation. Reference sequence: '0000000110000000'.

5.2.4 Extended Error Analysis

If the analysis is extended to consider taps for the whole sequence length, the picture is somewhat different to that obtained in chapter 4. Table 5.3 shows the sequences causing errors considered over the entire 16 chip length of the OOC; again the sequences may be considered unique since the data set is exhaustive. From Table 5.3 it is clear that sequences of the form ‘*******20*******’ and ‘*******02*******’ remain potentially correctable.
as was the case with the simple correlator synchronised to either of the code phases considered so far. Furthermore, as in the analysis of the sequences causing errors in the simple correlator considered over the full 16 chips given in Table 4.3, there now exists the possibility to correct interference of the form ‘******11******’ that mimics the desired code sequence, since there is unique information available to permit correct classification. Although conflicts remain in the data set, comparison of Table 5.3 with Table 4.3 shows that there are now only 2 cases (sequences 15 and 16, Table 5.3) where this occurs using the ‘0000000110000000’ OOC phase as opposed to the 9 arising through the use of the ‘1100000000000000’ phase. This means that there is the potential for further improvement over that achieved using the ALNs in chapter 4; the best possible performance considering the full 16 taps and retaining intensity information is now equivalent to an error rate of 9.77 x 10^-4. However, as in chapter 4, how this may be achieved is not specified by this analysis.

<table>
<thead>
<tr>
<th>Seq. Type</th>
<th>Unique Composite Causing Error</th>
<th>Actual Output</th>
<th>Target Output</th>
<th>Freq. with Target = 0</th>
<th>Freq. with Target = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0000110200000000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0000010200100000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0000100201000000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0000000201100000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0000000020110000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0000010020100000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0000001020010000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0000011020000000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0000000110200000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0000020110000000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0000010110010000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0000100110100000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0000001111010000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0000010111000000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0000000111101000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>58</td>
</tr>
<tr>
<td>16</td>
<td>0000101110000000</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>58</td>
</tr>
</tbody>
</table>

Table 5.3. Sequences causing errors with the simple correlator considered over the full 16 taps.
Reference sequence: ‘0000000110000000’.

Further examination of Table 5.3 shows that the composite sequences occur in mirror image pairs; reflection of a given sequence in Table 5.3, either about its end or its midpoint, also results in a sequence in Table 5.3, e.g. sequences 1 and 5, 15 and 16 etc. This should
not be surprising given the symmetry imposed by the selection of the code phase. Moreover, the possibility for further improvement in performance is seen to derive from this. Being sparse, the OOCs contributing to the composite sequences have little structure and any mechanism to maximise the improvement in error rate must be able to use this effectively. Considering the composite sequence over the full 16 taps and synchronisation to the '0000000110000000' phase means that the ALN receiver's field of view will now always include the other positive chips of the interfering OOCs whenever one of them occupies a desired user's chip position. This is in contrast to the analysis of the previous chapter where the choice of code phase meant that the full structure of the interfering codes was not always "visible" at the ALN input.

5.3 The Hard-limited Correlator

5.3.1 Configuration

As with the simple correlator the configuration for the hard-limited correlator is the same as in the previous chapter, see Figure 4.2.

5.3.2 Simulation Results

No difference in error rate is anticipated to that obtained for the hard-limited correlator in section 4.3, since again only 2 taps are being considered and the interfering users are uniformly distributed. This is verified by the results presented in Table 5.4 which are again in line with that predicted by theory using equation (2.10).

| Errors | 32 |
| No. Patterns in Set | 8192 |
| Error Rate | $3.91 \times 10^{-3}$ |

Table 5.4. Simulation error rate obtained for the hard-limiting correlator.

5.3.3 Error Sequence Analysis

Table 5.5 shows that the types of composite sequences causing errors in the hard-limited correlator simulation correspond to those of type C in Table 5.2. Errors no longer occur due to those of types A and B for the same reasons as given in chapter 4, i.e. cases where summation of optical intensity on one tap previously caused the output threshold to be exceeded, are now eliminated. Again errors now only occur when interfering users contribute to different tap positions of the correlator corresponding to the occupied chips of the desired user's sequence. The phase to which the correlator is synchronised makes no difference, since the correlator only considers the positive chips associated with the desired
Table 5.5. Input sequences causing errors in the hard-limited correlator simulation. Reference sequence: ‘0000001100000000’.

<table>
<thead>
<tr>
<th>Seq. Type</th>
<th>Generic Sequence Causing Error</th>
<th>Actual Output</th>
<th>Target Output</th>
<th>Freq. with Target = 0</th>
<th>Freq. with Target = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>1111111111111111</td>
<td>1</td>
<td>0</td>
<td>32</td>
<td>4096</td>
</tr>
</tbody>
</table>

This type of receiver is again achieving the best possible performance given the data set and the number of taps considered, since it is only limited by the conflicts in the data set under the non-linear transformation and the classification is performed so as to minimise the error rate.

5.3.4 Extended Error Analysis

Examination of the sequences giving rise to errors in the simple correlator has demonstrated that increasing the number of taps considered offers the potential for a reduction in error rate. Moreover, synchronisation to the ‘00000001100000000’ phase of the desired user’s OOC was seen to offer a greater degree of improvement than that possible with the ‘1100000000000000’ phase. The sequences causing errors with the hard-limiting correlator are shown in Table 5.6 and are now examined over the full 16 chip period to see if a change in code phase offers any further improvement to ALN receivers using this form of input.

From Table 5.6 it can be seen that, as in the previous chapter, multi-level composite sequences in the data set previously only associated with targets of ‘1’ are mapped to sequences previously only paired with those of target ‘0’, thus creating conflicts in the sequences presented to any receiver architecture. For example, the sequence ‘00001002201000000’ previously uniquely paired with a target of ‘1’ is mapped to ‘00001001101000000’ (sequence 2 Table 5.6), a sequence in the data set corresponding only to a target of ‘0’ and unaffected by the hard-limiting operation.

Conflicts are also generated where unique sequences in the data set with different targets are mapped to a single sequence that did not exist in the raw data set, for example sequences 7 and 8 in Table 5.6. Analysis of the table as a whole shows, that despite considering the clipped composite sequences over 16 chips, no improvement in performance is possible, since it is fundamentally limited by the conflicts in the transformed data set. The best that can be done is to minimise the error rate by classifying each conflicting sequence with its higher frequency target pair. In each case the pairing would be with a target of ‘1’ and the resulting error rate of $3.91 \times 10^{-3}$ would correspond to that of...
the hard-limited correlator with only 2 taps, as was the case with the examination of the expanded data set for the '1100000000000000' phase. Performance is therefore not improved by increasing the number of taps and changing the code phase. Once again it is has been demonstrated that hard-limiting discards information that has the potential to be used beneficially in order to reduce the error rate.

5.4 ALN Receivers

5.4.1 Configuration

Analysis of the simple and hard-limited correlators in the previous two sections has shown that performance is not improved by selecting a different phase of the OOC to which the desired user is synchronised. This was to be expected since these receivers only consider the 2 taps associated with the desired user and once again make no use of the structure of other interfering codes in their decision as to the actual code sent. However, the extended analysis of the composite sequences causing errors in these receivers over the full 16 chips of the OOC has indicated that further performance improvements are possible over those obtained with the ALNs in chapter 4, provided intensity information is retained. As in chapter 4, the analysis of Table 5.2 has shown that using only two taps the best possible performance is equivalent to that of the hard-limited correlator. Therefore ALNs will be considered that examine the composite sequences over the full 16 chip period, synchronised to the '0000001100000000' phase and with the intensity data suitably encoded. The configuration of the ALN receiver is shown overleaf in Figure 5.1. In this case the ALN can be considered as examining the same 2 taps as the correlator plus 7 "past" and 7 "future" values of the composite sequences. The possibility of examining such
Additional "future" taps

Input

Additional "past" taps

Encoded Input Vector

Complements

Taps as per correlators

Figure 5.1. Schematic showing configuration for an ALN synchronised to the '000000110000000' OOC phase. The receiver can be considered to examine both "future" and "past" taps with respect to the simple correlation receiver, which are then passed on to the encoding stage.

inputs is realised by placing the actual taps corresponding to those of the desired OOC at the centre of the delay-line.

5.4.2 ALN Simulation Results

The range of results obtained for this configuration of ALN using the complete data set and the initial tree sizes and encoding schemes of chapter 3 are presented in Table 5.7. The figures for worst case error rates indicate that again performance is always at least as good
<table>
<thead>
<tr>
<th><strong>Input Encoding Scheme</strong></th>
<th><strong>Initial Tree Size (Leaves)</strong></th>
<th><strong>Average Final Tree Size (Leaves)</strong></th>
<th><strong>Best Case Error Rate (Single Tree)</strong></th>
<th><strong>Worst Case Error Rate (Single Tree)</strong></th>
<th><strong>Average Error Rate (50 Trees)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bar Code</strong></td>
<td>4096</td>
<td>218</td>
<td>1.95 x 10^{-3}</td>
<td>3.91 x 10^{-3}</td>
<td>3.34 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>8192</td>
<td>418</td>
<td>9.77 x 10^{-4}</td>
<td>3.91 x 10^{-3}</td>
<td>2.74 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>16384</td>
<td>662</td>
<td>9.77 x 10^{-4}</td>
<td>3.91 x 10^{-3}</td>
<td>2.20 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>32768</td>
<td>1126</td>
<td>9.77 x 10^{-4}</td>
<td>2.93 x 10^{-3}</td>
<td>1.39 x 10^{-3}</td>
</tr>
<tr>
<td><strong>Tempâ€™e Code</strong></td>
<td>4096</td>
<td>210</td>
<td>1.95 x 10^{-3}</td>
<td>3.91 x 10^{-3}</td>
<td>3.31 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>8192</td>
<td>345</td>
<td>9.77 x 10^{-4}</td>
<td>3.91 x 10^{-3}</td>
<td>2.89 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>16384</td>
<td>624</td>
<td>9.77 x 10^{-4}</td>
<td>3.91 x 10^{-3}</td>
<td>2.27 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>32768</td>
<td>1042</td>
<td>9.77 x 10^{-4}</td>
<td>3.42 x 10^{-3}</td>
<td>1.69 x 10^{-3}</td>
</tr>
<tr>
<td><strong>Binary Code</strong></td>
<td>4096</td>
<td>330</td>
<td>9.77 x 10^{-4}</td>
<td>3.91 x 10^{-3}</td>
<td>2.49 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>8192</td>
<td>461</td>
<td>9.77 x 10^{-4}</td>
<td>3.42 x 10^{-3}</td>
<td>2.12 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>16384</td>
<td>744</td>
<td>9.77 x 10^{-4}</td>
<td>2.93 x 10^{-3}</td>
<td>1.65 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>32768</td>
<td>1121</td>
<td>9.77 x 10^{-4}</td>
<td>2.93 x 10^{-3}</td>
<td>1.16 x 10^{-3}</td>
</tr>
<tr>
<td><strong>Scatter Code 1</strong></td>
<td>4096</td>
<td>202</td>
<td>1.95 x 10^{-3}</td>
<td>3.91 x 10^{-3}</td>
<td>3.34 x 10^{-3}</td>
</tr>
<tr>
<td>(d = 1)</td>
<td>8192</td>
<td>370</td>
<td>9.77 x 10^{-4}</td>
<td>3.91 x 10^{-3}</td>
<td>2.87 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>16384</td>
<td>620</td>
<td>9.77 x 10^{-4}</td>
<td>3.91 x 10^{-3}</td>
<td>2.42 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>32768</td>
<td>957</td>
<td>9.77 x 10^{-4}</td>
<td>3.42 x 10^{-3}</td>
<td>2.01 x 10^{-3}</td>
</tr>
<tr>
<td><strong>Scatter Code 2</strong></td>
<td>4096</td>
<td>306</td>
<td>9.77 x 10^{-4}</td>
<td>3.91 x 10^{-3}</td>
<td>2.39 x 10^{-3}</td>
</tr>
<tr>
<td>(d = 2)</td>
<td>8192</td>
<td>562</td>
<td>9.77 x 10^{-4}</td>
<td>2.93 x 10^{-3}</td>
<td>1.45 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>16384</td>
<td>773</td>
<td>9.77 x 10^{-4}</td>
<td>2.93 x 10^{-3}</td>
<td>1.37 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>32768</td>
<td>1289</td>
<td>9.77 x 10^{-4}</td>
<td>2.93 x 10^{-3}</td>
<td>1.04 x 10^{-3}</td>
</tr>
<tr>
<td><strong>Hard-limited Input</strong></td>
<td>4096</td>
<td>129</td>
<td>3.91 x 10^{-3}</td>
<td>3.91 x 10^{-3}</td>
<td>3.91 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>8192</td>
<td>283</td>
<td>3.91 x 10^{-3}</td>
<td>3.91 x 10^{-3}</td>
<td>3.91 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>16384</td>
<td>311</td>
<td>3.91 x 10^{-3}</td>
<td>3.91 x 10^{-3}</td>
<td>3.91 x 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>32768</td>
<td>510</td>
<td>3.91 x 10^{-3}</td>
<td>3.91 x 10^{-3}</td>
<td>3.91 x 10^{-3}</td>
</tr>
</tbody>
</table>

Table 5.7. The range of results obtained for the ALN receivers trained using the encoding schemes and initial tree sizes described in chapter 3.

as the previous best case receiver, i.e. the hard-limited correlator. Moreover, comparison with Table 4.7 shows that when the worst case error rate is reduced below this level, the improvement is greater than that obtained with the ALNs using the configuration of chapter 4.

The best case error rates show that again, with the exception of ALNs using hard-limited encoding, significant improvements in performance are attained with respect to the
previously proposed receivers. Furthermore, as might be anticipated from the analysis of
the data set in section 5.2.3, the best case error rates of Table 5.7 are seen to be lower than
those of Table 4.7. It should also be noted that these rates are obtained for smaller initial
tree sizes and over the range of intensity preserving encoding schemes.

5.4.3 ALN Training Parameters

A general examination of the effect of training parameters on performance is now presented
using the average error rate obtained for each set of fifty trees trained with the same input
encoding schemes and initial tree sizes. This is a prelude to a more detailed study of
individual ALN learning behaviour presented in the next section. The average error rates of
Table 5.7 are shown plotted in Figure 5.2. Also presented in Figure 5.2 are the
performance bounds corresponding to the previous best case error rate and the best
possible error rate given the analysis of the data set presented in section 5.2.3.

As in chapter 4, the simulation results for ALNs using hard-limited encoding are seen to
consistently achieve the limited error rate predicted by the analysis from the data set under
transformation given in section 5.3.3, i.e. that of the hard-limited correlator. This again
demonstrates that intensity information is discarded before it can be used and the necessity
to preserve it if any further reduction in error rate is to be achieved. Study of the structure
of individual trees again revealed cases where the most efficient receiver structure was
found to be a single AND gate, again corresponding to Salehi’s optical AND gate model
[Salehi, 1989a] described in chapter 2. However the figures for average tree size again
demonstrate that this is not the norm and that the learning algorithm does not consider the
efficiency of implementation in terms of gates required in pursuit of its goal.

The plots in Figure 5.2 indicate that all intensity retaining encoding schemes have the
capability to improve performance over that of the previous best case. As in chapter 4,
performance is seen to increase with initial tree size for a given input encoding scheme.
Again this should be expected since larger trees are capable of implementing more complex
functions and are less likely to be limited by the configuration of input vector to ALN leaf
connections. However, the average error rates of Figure 5.2 are consistently lower than
those shown in Figure 4.4 for all encoding schemes. Moreover, the degree of change is
much more pronounced in Figure 5.2 than in Figure 4.4. Indeed, for initial tree sizes of
32768 leaves, all intensity preserving encoding schemes exhibit a tendency to achieve a
performance closer to the best possible error rate given the data set used.

The plots of Figure 5.2 reinforce the conclusions drawn in chapter 4, regarding the
properties of the encoding schemes used. Looking at the results for the temperature code
and two scatter codes again shows the ability of the higher distance code, scatter code 2 (d
Figure 5.2. Effect of training parameters on error rate averaged over fifty trees.
= 2), to achieve a higher level of performance for the same encoding length. Once more this is achieved using a larger final folded tree than obtained with either of the other two encoding methods, again suggesting that the higher distance property allows more nodes to be affected during training, to create a more complex tree function better suited to the task from similar initial conditions. ALNs using the temperature code and scatter code 1 \((d = 1)\) again give rise to very similar average error rates and final tree sizes consistent with their identical hamming maps (see Figure 3.5(b) and (d)) and demonstrate the apparent redundancy of the order property of the temperature code. Interestingly, comparison of the average final tree sizes in Table 5.7 with those in Table 4.7 shows that, with the exception of ALNs using scatter code 2 and 4096 initial leaves, final tree sizes are greater for the ALN configuration of this chapter. Taken in conjunction with the lower average error rates obtained, which imply the correction of more errors, this can be seen as reflecting the ability of the ALN to incorporate the greater knowledge of interfering users available to it as a result of using this configuration.

The plots obtained from ALNs trained using the bar code and scatter code 2 once more allow comparison of input encoding length since the hamming maps of both codes are identical (see Figures 3.5 (a) and (e)). As in chapter 4, scatter code 2 is seen to permit a lower error rate than the bar code for all the initial tree sizes considered. The superior performance of the lower encoding length may be attributed to the fact that, using scatter code 2, more ALN leaves are connected to each bit of the encoded input vector. Therefore, any change in the input has the opportunity to affect more ALN nodes during adaptation. This, coupled with the fact that the tree sizes obtained using scatter code 2 are also larger than those using the bar code, as shown in Table 5.7, suggests that the improvement is derived by forming a more complex function of fewer input variables.

Given the effects noted above, the similarity in average error rates obtained using the bar code, temperature code, and scatter code 1, for all but the largest initial tree size, is not unexpected. It would seem that the larger distance property of the bar code is offset by the shorter encoding lengths of both the temperature code and scatter code 1. The divergence of ALNs using the bar code for trees of 32768 leaves also noted in chapter 4 again suggests the dominance of hamming distance, once a sufficiently large tree is available, and that this is more effective than increasing the tree size alone for a lower distance encoding scheme. This is also demonstrated by the plots for the binary code and scatter code 2 since, although the former permits more connections per encoded input vector, the trees produced using the latter, with a higher absolute hamming distance, result in the lower average error rate. The same is true of the bar and binary codes; although as shown the binary code permits the lower error rate, if taken purely on a connections per encoded input vector element basis, the bar code outperforms the binary code for initial tree sizes greater than 16384 leaves.
5.4.4 ALN Learning Behaviour

The frequencies with which given numbers of actual errors were made by ALNs in each set trained using the same parameters, are shown in Figures 5.3 to 5.7, to enable further insight into learning and a comparison with the behaviour observed in chapter 4. The best possible performance for this configuration, based on the analysis of section 5.2.3, corresponds to 8 actual errors and that of the hard-limited correlator to 32 actual errors. The tendency for the proportion of trees making fewer actual errors to increase with tree size is observed, as demonstrated by the average error rates of Figure 5.2. However, the transition towards trees making progressively fewer actual numbers of errors, as demonstrated by looking from parts (a) to (d) for each of Figures 5.3 to 5.7, is much more smooth and rapid than that associated with Figures 4.5 to 4.9. Indeed, although the level of performance is still dependent on the particular input encoding scheme used, the degree of change is much more marked for all encoding schemes. For example, using 8192 initial leaves, all encoding techniques produce examples of best case performance. Furthermore, for trees of this size employing scatter code 2, ALNs achieving the lowest possible error rate dominate the set of all trees trained using this combination of parameters, as shown in Figure 5.7 (b). Figures 5.3 (d) and 5.5 (d) also show that, using 32768 leaves, best case performance also accounts for the majority of trees trained using the bar and binary codes. Moreover, even the temperature code and scatter code 1, nominally producing the least improvement in performance, manage to achieve significant numbers of best case error rate trees for this size and configuration, Figures 5.4 (d) and 5.6 (d), whereas using that of chapter 4 they did not produce any trees achieving the lowest error rate set by the analysis of the data set.

The distributions of the number of actual errors made by the sets of trees in Figures 5.3 to 5.7 show no tendency for the sequences causing errors in the correlation receivers to be corrected in groups of more than one, as found in chapter 4. The distribution is discrete and the actual number of errors varies by units of 4, which corresponds to correction of one further sequence of the form ‘*******11*******’ for each grouping between 32 and 8 actual errors. This is consistent with the composite sequence frequencies associated with targets of ‘0’ in Table 5.3 and suggests arbitrary correction of sequences once a performance level equivalent to that of the hard-limited correlator has been reached.

The results of an investigation of the sequences corrected to attain the levels of performance associated with the actual numbers of errors shown in Figures 5.3 to 5.7 are displayed in Table 5.8. Achievement of 32 actual errors was found to correspond to the correction of all sequences of the form ‘*******02*******’ and ‘*******20*******’, as shown in group 1 of Table 5.8. These are the same sequences causing errors in the simple correlator that are eliminated by hard-limiting the input. However, in this case, the retention of
Figure 5.3. Performance distribution of trained trees for the bar code.
Figure 5.4. Performance distribution of trained trees for the temperature code.
Figure 5.5. Performance distribution of trained trees for the binary code.
Figure 5.6. Performance distribution of trained trees for scatter code 1 \((d = 1)\).
Figure 5.7. Performance distribution of trained trees for scatter code 2 (d = 2).
<table>
<thead>
<tr>
<th>Group</th>
<th>Error Rate</th>
<th>Actual Errors</th>
<th>Sequences Corrected w.r.t. Simple Correlator</th>
<th>Freq. with Target = 0</th>
<th>Freq. with Target = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.91 x 10^{-3}</td>
<td>32</td>
<td><em><strong><strong><strong>02</strong></strong></strong></em></td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><em><strong><strong><strong>20</strong></strong></strong></em></td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3.42 x 10^{-3}</td>
<td>28</td>
<td>as above + 0000020110000000 or 0000000110200000</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2.93 x 10^{-3}</td>
<td>24</td>
<td>as group 1 + 0000020110000000 0000000110200000</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2.44 x 10^{-3}</td>
<td>20</td>
<td>as group 3 + 1 from group 7 see below</td>
<td>see below</td>
<td>see below</td>
</tr>
<tr>
<td>5</td>
<td>1.95 x 10^{-3}</td>
<td>16</td>
<td>as group 3 + 2 from group 7</td>
<td>“”</td>
<td>“”</td>
</tr>
<tr>
<td>6</td>
<td>1.46 x 10^{-3}</td>
<td>12</td>
<td>as group 3 + 3 from group 7</td>
<td>“”</td>
<td>“”</td>
</tr>
<tr>
<td>7</td>
<td>9.77 x 10^{-4}</td>
<td>8</td>
<td>as group 3 + 0000010110010000 0000100110100000 0000001110100000 000001011101000000</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.8. The sequences corrected for the actual numbers of errors shown in Figures 5.3 to 5.7. Reference sequence: ‘0000000110000000’.  

Intensity information by all the other encoding schemes does not rule out the possibility for further correction of sequences. The relative ease of correction of these types of sequences with respect to those of the form ‘*******11*******’ is not unexpected. These types of sequences do not mimic the desired data ‘1’ OOC at its positive chip positions. Furthermore, the hamming distance between these sequences and the desired ‘1’ and all-zero sequences is such that, for the bar code, binary code and scatter code 2, they are closer to the all-zero vector, the target output, and for the temperature code and scatter code 1 they are equidistant between the two. Any insensitivity properties of the tree acting on the ‘...02...’ or ‘...20...’ elements of the input should therefore be able to group them with the desired target more easily than the ‘...11...’ sequences of Table 5.3 which, by their nature, will always be closer to the desired data ‘1’, since they mimic its positive chip.
Attainment of 28 actual errors, group 2 Table 5.8, was found to correspond to the correction of either the sequence ‘00000001102000000’ or ‘0000020110000000’ and of 24 actual errors, group 3 Table 5.8, to the correction of both. Whilst in the case of the temperature code and scatter code 1, this might be explained based on the larger distance between levels ‘2’ and ‘0’, and levels ‘1’ and ‘0’, which distinguish the remaining sequences from the desired data sequence (see Figures 3.5 (b) and (d)), this does not explain why this should be so for the other input encodings.

The remaining ALN error figures from 20 to 8 actual errors, groups 4 to 7, were indeed found to correspond to correction of one further sequence in addition to those corrected for 24 errors. No particular combinations of sequences were found to dominate these rates. Table 5.8 shows that the additional sequences corrected for groups 4 to 7 are all distinguished from the desired data ‘1’ at 2 composite chip positions. Additionally, the frequencies associated with the desired output target value of ‘0’ are constant for each sequence. This is in contrast to the similar sequences corrected by the previous ALN configuration, listed in groups 3 to 5 of Table 4.8, which in some cases only differ from the desired sequence at one chip position and have non-constant target ‘0’ frequencies which are typically less than those of Table 5.8. This explains the superior performance results of the present configuration, since more information is available for discrimination both within the input composite sequences and in the number of times they occur. These factors also explain why there appears to be no preference for the correction of particular groups of sequences for a given error rate and the more even and pronounced transition to trees of lower error rates with tree size. As a caveat, it should be added that the more even distribution of trees over the performance range means there are fewer examples of the types of sequences corrected for a given rate than found in chapter 4. This may prevent the discernment of any trend for grouping by similarity as previously found.

5.4.5 Performance Assessment

The study of the training parameters and their effect on learning ability has provided guidelines for producing ALNs capable of significantly lower error rates than the previously proposed receivers. The trend for larger trees to give better performance, as in chapter 4, is again noted. However, using this configuration of ALN, it has been shown that for a given tree size, the average error rates obtained are consistently lower than those using that of chapter 4. Indeed, using a high hamming distance per length input encoding scheme, scatter code 2 (d = 2), it was observed that best case error rate trees predominated the set trained for initial sizes of 8192 leaves. Again it is postulated that this is a result of the input being able to affect more nodes during adaptation. Achievement of lower error
rates has also been shown to be easier than using the previous configuration with all encoding schemes producing higher numbers of best case trees given sufficiently large tree sizes.

The improved level of performance is again attributable to the utilisation of retained intensity data and the consideration of additional temporal data with respect to the correlation receivers. An ALN using 2 taps, retaining suitably encoded intensity data, would enable the correction of sequences of the type causing errors in the simple correlator receiver to give a performance equivalent to that of the hard-limited correlator. However, by considering the input over the full length of an OOC, retention of intensity data allows the correction of these types of sequences without generating the conflicts associated with hard-limiting that constrain further improvement. This then allows the additional temporal data of the input to be used to form a better classification based on the structure of the interfering users. The reason for the enhanced performance over the previous ALN configuration is that when two interfering users' codes sum to mimic the desired OOC their other positive chips are always available to the ALN to provide a context for classification. Despite the sparse nature of the OOCs, the ALN is therefore able to consider the full structure of the interference. This is not the case using the ALN configuration of chapter 4.

Table 5.8 provides a comparison of best case error rates under the worse case conditions of chip synchronous interference. For the ALN configuration synchronised to the '0000000110000000' phase of the desired user's code, the error rate is reduced by a factor of 8 with respect to the simple correlator, by a factor of 4 compared to the hard-limited correlator and by nearly a factor of 2 over the previous ALN configuration. As in chapter 4, the best case ALN error rate represents the best possible performance given the data set; sequences with unique targets are correctly classified and those with conflicting targets are paired with the higher frequency target so as to minimise the error rate.

<table>
<thead>
<tr>
<th>Receiver</th>
<th>Simple Correlator *</th>
<th>Hard-limited Correlator *</th>
<th>Best Case ALN †</th>
<th>Best Case ALN ‡</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Rate</td>
<td>7.81 x 10^{-3}</td>
<td>3.91 x 10^{-3}</td>
<td>1.83 x 10^{-3}</td>
<td>0.98 x 10^{-3}</td>
</tr>
</tbody>
</table>

Table 5.9. Receiver performance comparison under worst case interference conditions. Reference sequence: † '1100000000000000', ‡ '0000000110000000', * either sequence.

Similar comments to those made in chapter 4 apply regarding efficiency of implementation in terms of tree size. Table 5.7 again shows the tendency for larger initial tree sizes to produce lower error rates and larger final folded trees. Once more this is a result of the learning algorithm making no attempt to minimise tree size and the lower probability of the performance of larger trees to be impeded by connection constraints. The potential for more
efficient implementations is highlighted by the achievement of best case error rates for initial trees of 4096 leaves using scatter code 2 shown in Table 5.7 and Figure 5.7 (a). Given the ALN learning algorithm used, there are again two options for producing more compact trees. If the groups of Table 5.8 (except group 7) correspond to local minima, then one could search the connection space of smaller sized trees for a configuration giving improved performance. By contrast, if such groups represent plateaux of performance one could train smaller trees for a longer period of time. Both of these methods again imply a trade-off between training time and tree size.

5.5 Summary

The results obtained by simulation of the simple and hard-limited correlators, synchronised to the ‘00000011000000’ phase of the desired OOC, were found to be identical to those achieved using the ‘11000000000000’ code phase in chapter 4, as expected. The analysis of input sequences causing errors with the correlation receivers, developed in chapter 4, was repeated showing that:

- The best possible error rate achievable, using only the taps associated with the desired user’s sequence, is once more identical to that of the hard-limited correlator, even if intensity information could be used;
- There exists a potential for lower error rates, than achieved with either the correlation receivers or the previous ALN configuration, providing the relevant information regarding interfering users is utilised. However, complete elimination of MAI is still not possible;
- The provision of additional temporal data alone, again limits performance to that of the hard-limited correlator. Once again hard-limiting of an extended input sequence discards useful information, preventing further reduction in error rate.

Results provided by the simulation of ALNs showed that:

- The potential for significant reductions in error rate over the correlation receivers and the previous ALN configuration, by consideration of additional temporal and intensity data, was again realisable;
- For a given input encoding scheme, the average ALN error rate again decreased with increasing initial tree size. Furthermore, using the same initial tree sizes, lower error rates were obtained than when using the previous ALN configuration;
- For a given initial tree size, the lowest error rates were, once more, most easily
achieved with high hamming distance, low length encoding schemes. Moreover, all intensity retaining input encoding schemes produced lower error rates at smaller initial tree sizes than the previous ALN configuration;

- The enhanced ALN performance is attributable to the fact that the full structure of the interfering users' code sequences is made accessible, under circumstances that would cause errors with the correlation receivers. The incorporation of this additional information is reflected by the achievement of lower error rates and larger final tree sizes than produced using the same initial tree sizes and the configuration of chapter 4.
Chapter 6 ALN Receivers in Optical CDMA Systems employing Gold-like Codes

6.1 Introduction

Chapters 4 and 5 were concerned with using ALN receivers to enhance performance in optical CDMA systems employing OOCs. OOCs possess correlation properties that were developed specifically for use with incoherent optical correlators to avoid the high cross-correlation peaks associated with using the more conventional PN type of spreading sequences in such systems. However, the sparse nature of OOCs and the OOK modulation scheme used mean that the additional information made available for signal classification by consideration of interfering users' sequences is limited. Indeed, it has been shown in chapter 5 that the best results are obtained when an ALN is configured so that it has access to the maximum possible detail of the structure of the interfering code sequences.

This chapter considers the performance of ALN receivers in an incoherent optical CDMA system employing Gold-like codes; spreading sequences usually associated with conventional bipolar CDMA systems. The codes used are those described in chapter 3; the desired sequence, '111100010011010', being shown in Figure 3.3(a) with Figures 3.3(b) and 3.3(c) corresponding to the interfering users' code sequences. The correlation functions of these codes are not constrained as with the OOCs and therefore higher error rates would be anticipated using a simple correlation receiver. However, it is interesting to consider the performance of receivers able to take into account the sequences of other interfering users since the Gold-like codes contain more structure than the sparse OOCs.

The format of this chapter follows that of chapters 4 and 5. The performance of the simple and hard-limited correlation receivers are investigated and the input composite sequences causing errors to arise in each are examined. Each analysis is then extended to consider whether such errors are correctable via the consideration of further taps, i.e. the provision of more information regarding the interfering users, in order to establish limits on ALN performance using this form of input. The actual performance of ALN receivers is then studied to allow comparison with these limits and with the systems using OOCs described in the previous chapters.
6.2 The Simple Correlator

6.2.1 Configuration

The configuration of the simple correlator is similar to that shown in Figure 4.1(a). However, the receiver now has 8 taps positioned according to the positive chip locations of the desired sequence '111100010011010'. Correspondingly, the output threshold Th, shown in Figure 4.1(b), is now set to a value of 8, the weight of the desired code, in order to minimise the effects of interference due to other users.

6.2.2 Simulation Results

The simulation results obtained for the simple correlator with the input data set created using method ‘B’ of chapter 3 are shown in Table 6.1. The error rate achieved using Gold-like codes is approximately 2.5 times higher than that of the systems using OOCs of a similar length, as can be seen by comparison with Tables 4.1 and 5.1. The decrease in performance is expected since the more even balance of ‘1’ and ‘0’ in the Gold-like codes gives rise to higher cross-correlation products than the sparse OOCs with the result that the receiver output threshold is exceeded in error more frequently.

<table>
<thead>
<tr>
<th>Number of Errors</th>
<th>144</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Set Size</td>
<td>7200</td>
</tr>
<tr>
<td>Error Rate</td>
<td>2.00 x 10^{-2}</td>
</tr>
</tbody>
</table>

Table 6.1. Simulation results for the simple correlator.

6.2.3 Error Sequence Analysis

The composite sequences causing errors at the output of the simple correlator are shown in Table 6.2. Only the chip values relevant to the correlator tap positions are displayed; the other values being marked ‘***’, indicating don’t care states that play no part in deciding the data bit being received. As with the systems using OOCs, errors are seen to arise when the desired output of the correlator is a ‘0’ and the actual output is a ‘1’. This occurs when the contributions of interfering users’ chip values sum at the correlator taps so as to reach the output threshold, despite the fact that the actual desired user is transmitting nothing.

There are two mechanisms by which the threshold may be erroneously crossed. The first is characterised by the presence of composite sequences of type 1 from Table 6.2 at the correlator input. In this case interfering users’ code sequences overlap such that the input mimics exactly the desired user’s data ‘1’ sequence with the resultant composite sequence contributing exactly one positive chip to each of the correlator taps. The second occurs...
when the input is of a form consistent with sequences 2 to 76 in Table 6.2. In such cases the composite chip sequence need not contribute to every tap of the correlator but the threshold may still be crossed due to localised summation of optical intensity. However, in contrast to the similar mechanisms described in sections 4.2.3 and 5.2.3 for OOCs, this may be derived either from summations of optical intensity at more than one tap or by a combination of single and multiple contributions at separate taps. This occurs because the Gold-like codes, having higher weights, are not constrained to affect only one tap per user’s code sequence i.e. if it is defined for the Gold-like codes, the correlation constraint $\lambda \neq 1$ whereas for the OOCs $\lambda = 1$. The higher code weights also mean there are more ways for sequences to interfere to achieve the threshold and this accounts for the greater numbers of sequences in Table 6.2 vis-à-vis Tables 4.2 and 5.2.

Looking at the relative frequencies of the targets associated with the sequences of Table 6.2 enables further comment on the operation of the simple correlator. Sequences of type 1 in Table 6.2 are associated with conflicting target values and thus errors are always expected for inputs of this form. However, in this case this type of sequence is classified with a target of ‘1’, the higher frequency input-target pairing, so as to minimise the error. The remaining composites in Table 6.2, 2-76, are uniquely paired with targets of ‘0’, so that the errors caused by these inputs are capable of being corrected based on information contained within the sequences, though not using the sum and threshold approach of the simple correlator.

<table>
<thead>
<tr>
<th>Seq. Type</th>
<th>Generic Composite Causing Error</th>
<th>Actual Output</th>
<th>Target Output</th>
<th>Freq. with Target = 0</th>
<th>Freq. with Target = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1111**<em><em>1**11</em>1</em></td>
<td>1</td>
<td>0</td>
<td>12</td>
<td>484</td>
</tr>
<tr>
<td>2</td>
<td>0112**<em><em>0**02</em>2</em></td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0212**<em><em>1**11</em>1</em></td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1211**<em><em>0**02</em>1</em></td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1202**<em><em>0**11</em>1</em></td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1111**<em><em>1**01</em>2</em></td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1101**<em><em>1**02</em>2</em></td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0102**<em><em>0**12</em>2</em></td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0202**<em><em>1**02</em>1</em></td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>2020**<em><em>1**21</em>0</em></td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1021**<em><em>0**12</em>1</em></td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1121**<em><em>0**21</em>0</em></td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>2120**<em><em>0**12</em>0</em></td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.2. Types of input causing errors at the output of the simple correlator. Reference sequence: ‘111100010011010’. Continued overleaf.
<table>
<thead>
<tr>
<th>Seq. Ref.</th>
<th>Generic Composite Causing Error</th>
<th>Actual Output</th>
<th>Target Output</th>
<th>Freq. with Target = 0</th>
<th>Freq. with Target = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>2111**<em><em>0**21</em>0</em></td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>2020<em><strong>1</strong>1</em>1*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>2010**<em><em>1**12</em>1</em></td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>1011<em><strong>0</strong>22</em>1*</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>1111<em><strong>1</strong>12</em>0*</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>1111<em><strong>1</strong>21</em>0*</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0112<em><strong>0</strong>12</em>1*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0212<em><strong>0</strong>21</em>0*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>1211<em><strong>0</strong>12</em>0*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>1202<em><strong>0</strong>21</em>0*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>1101<em><strong>1</strong>12</em>1*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>0102<em><strong>0</strong>22</em>1*</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>0202<em><strong>1</strong>12</em>0*</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>2020<em><strong>1</strong>20</em>1*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>1021<em><strong>0</strong>11</em>2*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>1121<em><strong>0</strong>20</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>2120<em><strong>0</strong>11</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>31</td>
<td>2111<em><strong>0</strong>20</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>32</td>
<td>2020<em><strong>1</strong>10</em>2*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>33</td>
<td>2010<em><strong>1</strong>11</em>2*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>34</td>
<td>1011<em><strong>0</strong>21</em>2*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>35</td>
<td>2010<em><strong>2</strong>11</em>1*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>36</td>
<td>1011<em><strong>1</strong>02</em>2*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>2110<em><strong>1</strong>02</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>38</td>
<td>2101<em><strong>1</strong>11</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>39</td>
<td>2010<em><strong>2</strong>01</em>2*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>2000<em><strong>2</strong>02</em>2*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>41</td>
<td>1001<em><strong>1</strong>12</em>2*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>42</td>
<td>1101<em><strong>2</strong>02</em>1*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>43</td>
<td>1021<em><strong>2</strong>11</em>0*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>44</td>
<td>0022<em><strong>1</strong>02</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>0122<em><strong>1</strong>11</em>0*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>46</td>
<td>1121<em><strong>1</strong>02</em>0*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>47</td>
<td>1112<em><strong>1</strong>11</em>0*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>48</td>
<td>1021<em><strong>2</strong>01</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.2 continued.
<table>
<thead>
<tr>
<th>Seq. Type</th>
<th>Generic Composite Causing Error</th>
<th>Actual Output</th>
<th>Target Output</th>
<th>Freq. with Target = 0</th>
<th>Freq. with Target = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>1011<em><strong>2</strong>02</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>0012<em><strong>1</strong>12</em>1*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>51</td>
<td>0112<em><strong>2</strong>02</em>0*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>52</td>
<td>1121<em><strong>1</strong>20</em>0*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>53</td>
<td>0122<em><strong>0</strong>11</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>54</td>
<td>0222<em><strong>0</strong>20</em>0*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>55</td>
<td>1121<em><strong>0</strong>11</em>0*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>56</td>
<td>1212<em><strong>0</strong>20</em>0*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>57</td>
<td>1121<em><strong>1</strong>10</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>58</td>
<td>0112<em><strong>0</strong>21</em>1*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>59</td>
<td>0212<em><strong>1</strong>11</em>0*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>60</td>
<td>2120<em><strong>2</strong>10</em>0*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>61</td>
<td>1121<em><strong>1</strong>01</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>62</td>
<td>1221<em><strong>1</strong>10</em>0*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>63</td>
<td>2220<em><strong>1</strong>01</em>0*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>64</td>
<td>2211<em><strong>1</strong>10</em>0*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>65</td>
<td>2120<em><strong>2</strong>00</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>66</td>
<td>2110<em><strong>2</strong>01</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>67</td>
<td>1211<em><strong>2</strong>01</em>0*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>68</td>
<td>2011<em><strong>2</strong>10</em>1*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>69</td>
<td>1012<em><strong>1</strong>01</em>2*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>70</td>
<td>1112<em><strong>1</strong>10</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>71</td>
<td>2111<em><strong>1</strong>01</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>72</td>
<td>2102<em><strong>1</strong>10</em>1*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>73</td>
<td>2011<em><strong>2</strong>00</em>2*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>74</td>
<td>2001<em><strong>2</strong>01</em>2*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>75</td>
<td>1002<em><strong>1</strong>11</em>2*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>76</td>
<td>1102<em><strong>2</strong>01</em>1*</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.2 continued. Types of input sequences causing errors at the output of the simple correlator. Reference sequence: '11100010011010'.

6.2.4 Extended Error Analysis

The analysis of the sequences causing errors with the simple correlator is now extended to cover the full 15 taps of the generated composite sequence. Receivers using this form of input would therefore have access to information regarding the structure of interfering
users’ sequences as well as the desired sequence. The aim is to see whether, by consideration of these additional taps, there is the potential for further improvement in error rate by putting conditions previously causing errors into context.

From Table 6.2 it is clear that sequences of the types 2-76 remain potentially correctable when considered over 15 chips, since they are uniquely distinguishable based on only 8 chips. Therefore attention is focused on sequences of type 1 in Table 6.2 which can correspond either to receipt of a desired data ‘1’ or to the mimicking of a data ‘1’ by interfering users in the presence of an actual data ‘0’. The four unique kinds of sequences corresponding to sequence 1, and paired with a target of ‘0’ in Table 6.2, are shown in Table 6.3. It can be seen that, by examining the additional chip positions, all the sequences are now uniquely identified with a target of ‘0’ and the previous conflict in target values arising via consideration over only 8 taps is now resolved. This means that, from the data set point of view, all sequences may be correctly classified by a suitable receiver and that, in this case, performance need not be limited by MAI.

<table>
<thead>
<tr>
<th>Seq. Type</th>
<th>Unique Sequence Causing Error</th>
<th>Actual Output</th>
<th>Target Output</th>
<th>Freq. with Target = 0</th>
<th>Freq. with Target = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1111011110111110</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1111000102110122</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>111100012211010</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>111102010211010</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1111000102110111</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.3. Sequences of type 1 in Table 6.2 considered over the full 15 chips of the code length. Reference sequence: ‘111100010011010’.

6.3 The Hard-limited Correlator

6.3.1 Configuration

The configuration of the hard-limited correlator is similar to that shown in Figure 4.2(a) in section 4.3.1. However, as stated in section 6.2.1, the correlator stage now has 8 taps corresponding to the positive chips of the desired Gold-like sequence and the output threshold ‘Th’ is again set at 8. The input thresholding remains the same as in Figure 4.2(b).

6.3.2 Simulation Results

The simulation results for the hard-limited correlator are presented in Table 6.4. As in
section 6.2.2 testing was performed with the complete set generated using method ‘B’ described in chapter 3. The error rate obtained is lower than that of both the simple and hard-limited correlators in systems employing OOCs, as can be seen by comparison with Tables 4.1, 4.4, 5.1 and 5.4. The reasons for this will become clear in the following section.

| Number of Errors | 12 |
| Test Set Size | 7200 |
| Error Rate | $1.67 \times 10^{-3}$ |

Table 6.4. Simulation results for the hard-limited correlator.

6.3.3 Error Sequence Analysis

The types of sequences causing errors after hard-limiting are shown in Table 6.5. The inputs are of the form of sequence 1 in Table 6.2, mimicking the desired data ‘1’ sequence at all tap positions. The increase in the frequency associated with targets of ‘1’ is caused by the mapping of all multi-level composite sequences containing the desired sequence to this binary form. The reduction in error rate is caused by clipping sequences of the form 2-76 in Table 6.2. In this way, cases causing errors by localised summation of intensity at single taps are eliminated; the threshold no longer being reached since the transformed input sequences no longer contribute to all correlator taps. The behaviour of the hard-limited correlator is again analogous to an optical AND gate. Given the number of taps considered and the transformation imposed on the input, the hard-limited correlator performs as well as it can, since the conflicting sequences causing errors are classified so as to minimise the error rate as shown in Table 6.5.

The reason for a lower error rate than both correlation receivers using OOCs again stems from the structure of the Gold-like codes. The higher weight of these codes means that there are relatively fewer cases, where interfering users overlap so as to duplicate the desired code sequence, whereas cases causing the threshold to be reached by summation of optical intensity at taps are more likely since there are more ways of this occurring and it may now occur at more than 1 tap given $\lambda \neq 1$. Therefore, elimination of the latter form of errors by hard-limiting results in a lower error rate.

<table>
<thead>
<tr>
<th>Seq. Type</th>
<th>Generic Composite Causing Error</th>
<th>Actual Output</th>
<th>Target Output</th>
<th>Freq. with Target = 0</th>
<th>Freq. with Target = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1111<em><strong>1</strong>11</em></td>
<td>1</td>
<td>0</td>
<td>12</td>
<td>3600</td>
</tr>
</tbody>
</table>

Table 6.5. Sequences causing errors with the hard-limited correlator. Composite sequence and frequencies shown are those after transformation. Reference sequence: ‘111100010011010’.
6.3.4 Extended Error Analysis

The inputs in Table 6.5 causing errors with the hard-limited correlator, i.e. those associated with targets of ‘0’ giving actual outputs of ‘1’, are shown in Table 6.6 over 15 chips. It is clear that, despite the provision of additional data, no further improvement in performance is possible since these sequences are also associated with targets of ‘1’. This is as a result of the non-linear hard-limiting operation which removes intensity based information previously allowing correct classification. For example, although sequences of type 1 in Table 6.3, formerly uniquely paired with a target of ‘0’, are unaffected by the transformation, clipping of certain multi-level composites containing the desired data ‘1’ generates identical sequences with targets of ‘1’, as shown by the corresponding non-zero frequency entry for sequence 1 in Table 6.6. Sequences of type 2-4 do not exist for either target value in the raw data set and the conflicts are generated by the mapping of multi-level sequences with unique target values to identical sequences in the transformed input space. Given the conflict in target values, the best that can be done is to classify the input with the higher frequency target pairing, so as to minimise the error rate. This limits the best possible error rate to be the same as that of the hard-limited correlator $1.67 \times 10^{-3}$.

<table>
<thead>
<tr>
<th>Seq. Type</th>
<th>Unique Composite Causing Error</th>
<th>Actual Output</th>
<th>Target Output</th>
<th>Freq. with Target = 0</th>
<th>Freq. with Target = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11110111101110</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>11110001011101</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>90</td>
</tr>
<tr>
<td>3</td>
<td>11110001111100</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>141</td>
</tr>
<tr>
<td>4</td>
<td>1111010101111010</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>117</td>
</tr>
</tbody>
</table>

Table 6.6. Inputs of Table 6.5 with a target of ‘0’ shown over 15 chips. Reference sequence: ‘111100010011010’.

6.4 ALN Receivers

6.4.1 Configuration

The analysis of the inputs causing errors with the simple correlator given in section 6.2.2 has shown that improvement in performance to a level equivalent to that of the hard-limited correlator is possible by utilisation of intensity data and the 8 taps corresponding to the desired user’s sequence. Moreover, it has been shown from the extended analyses given in section 6.2.3 and 6.3.3 that, provided intensity based information is retained, consideration of additional taps has the potential to permit even further reductions in error rate. Indeed, it was demonstrated in section 6.2.3 that the information regarding the interfering users supplied by the extra taps is such that performance need not be limited by MAI at all.
As in chapters 4 and 5, the premise for using ALNs is that their ability to form more complex decision functions than the simple sum and threshold type associated with the correlation receivers, should enable enhanced performance, given the findings summarised above. As such, the investigation of ALN receivers is concentrated on those permitted to examine input composite sequences over the full 15 taps consistent with synchronisation to the desired user’s sequence of ‘11100010011010’. The best results are anticipated using input encoding schemes that preserve intensity-based information. However, ALNs were also trained using hard-limited input to compare simulation results with the analysis of section 6.3.4. The basic configuration of the ALN receivers is similar to that shown in Figures 4.1 and 5.1 with the input window reduced to 15 taps accordingly.

6.4.2 ALN Simulation Results

The range of results obtained for ALNs trained using the initial tree sizes and input encoding schemes covered in chapter 3 are shown in Table 6.7. The figures for worse case error rate indicate that, once more the performance of ALN receivers is always at least as good as that of the hard-limited correlator. However, the tendency for worst case rates to remain close to this level regardless of tree size, as noted in chapters 4 and 5, is no longer apparent. The best case error rates show that the capacity for significant improvement with respect to the correlation receivers, suggested by the extended analysis of input data, is indeed realisable and that the performance of ALNs need only be limited by the information present in this data. ALNs using hard-limited input achieve the best possible error rate, given the restrictions imposed by this encoding method, as discussed in section 6.3.4. Similarly, in line with the findings of section 6.2.4, ALNs retaining intensity-based information need not be limited by MAI whatsoever. The table also shows that this is possible for all such encoding schemes for initial tree sizes of 4096 initial leaves. Therefore, better performance is attainable than in the systems using OOCs covered in chapters 4 and 5 and for smaller initial tree sizes. The parity of best and worst case error rates noted for instances of larger tree sizes, dependent on the particular encoding scheme used, also demonstrates a greater ease in obtaining this level of performance than previously found.

6.4.3 ALN Training Parameters

The effect of the training parameters will now be studied using the values for average error rate obtained for each group of trees trained using the same combination of initial tree size and encoding method. These values are shown in Table 6.7 and have been plotted in Figure 6.1. Those obtained for hard-limited ALNs are not shown in Figure 6.1 in order to allow clearer distinction of the plots of the higher performance intensity preserving encoding schemes. As mentioned previously, the values obtained for ALNs using this encoding
Table 6.7. Range of performance results for the ALN receivers using a variety of initial tree sizes and input encoding schemes. It should be noted that the achievement of an error rate of zero, marked thus †, signifies that receiver performance is not limited by MAI.

The plots for the temperature code and the two scatter codes again allow comparison of the effect of hamming distance on learning, since all have the same encoding length. As noted in chapters 4 and 5, the higher distance property of scatter code 2 (d = 2) gives rise to the
Figure 6.1. Average error rates for the sets of 50 trees trained with the same initial sizes and encoding schemes.

Bar Code  Triangle Code  Scatter Code 2 (d = 2)
Temperature Code  Scatter Code 1 (d = 1)  Best Case (X-axis)
better average error rate. Indeed, using 8192 or more leaves, all of the trees in the set are seen to learn the training set perfectly. From Table 6.7 it is clear that once more this is associated with a larger average final tree size. This again suggests that the higher performance is attributable to the higher distance encoding allowing more nodes to be affected during training for the same number of input connections to leaf nodes. However, in Figure 5.1 the performance of both the temperature code and scatter code 1 is seen to converge to that of scatter code 2 with increasing tree size. This is in contrast to the equivalent plots obtained using OOCs shown in Figures 4.2 and 5.2, where scatter code 2 consistently outperformed the other codes over the same range of initial tree sizes.

Comparing the plots for the distance 2 codes, i.e. the bar code and scatter code 2, again demonstrates the relative superiority of the lower encoding length of scatter code 2, once more accompanied by higher average final folded tree sizes for the same initial sizes as shown in Table 6.7. Again this seems to indicate that this is achieved by forming a more complex function of fewer input variables. In this case, it is the greater number of node connections which dominates, hamming distances being the same. However, again in contrast to Figures 4.2 and 5.2, this is only apparent for the smallest tree size of 4096 leaves. For tree sizes of 8192 leaves, the average error rate for the bar encoded ALNs improves dramatically, again converging to best possible performance with further increases in tree size.

The similarity of the plots obtained using the bar code, temperature code and scatter code 1, and also the binary code and scatter code 2 for tree sizes of above 8192 leaves, are in line with the trade-off between input encoding length and hamming distance noted above. However, the tendency for the hamming distance of the encoding scheme to be the more dominant factor with increasing tree size noted in chapters 4 and 5 is not as pronounced. Although there is some evidence for this provided by differences between the binary code and scatter code 2 at 4096 leaves and between the bar code and both the temperature code and scatter code 1 at 8192 leaves, beyond these points convergence occurs for all encoding schemes, unlike in Figures 4.2 and 5.2. This difference may be explained by the structure of the spreading sequences assigned to the system users in each case. The Gold-like codes have an inherent distance advantage, built in before input encoding, due to their higher weights. Although all the ALN receivers considered in this study examine input over a similar numbers of taps, in the OOC systems each interfering user is limited to a contribution of 2 positive chips to each composite input sequence, whereas each interfering user in the Gold-like systems may contribute 6 positive chips (compare Figures 3.2 and 3.3).

As a final point, it should be mentioned that the final tree sizes shown in Table 6.7 are greater than those in Table 5.7. The differences in tree sizes noted between Table 5.7 and
4.7 were attributed to the incorporation of greater knowledge regarding the problem domain. The latest observations are consistent with this view; using the Gold-like codes there are more types of sequences causing errors with the simple correlator, which may be corrected using ALNs, compared to the systems employing OOCs.

6.4.4 ALN Learning Behaviour

The distributions of actual numbers of errors made by ALNs for each particular combination of training parameters are shown in Figures 6.2-6.6. The distributions underlie the average rates shown in Figure 6.1 and provide a more convenient means of contrasting behaviour with that obtained in the systems employing OOCs shown in Figures 4.5-4.9 and 5.3-5.7. Attainment of zero errors obviously reflects the best possible performance, though it should be remembered that this corresponds to the excision of MAI only as described in section 6.2.4. The worst case achieved by any ALN corresponds to 12 actual errors; a performance equivalent to that of the hard-limited correlator.

Figures 6.2-6.6 illustrate the relative ease with which performance may be enhanced with respect to the correlation receivers when Gold-like codes are employed in place of OOCs. There is no evidence to suggest that learning becomes progressively more difficult once a performance equivalent to that of the hard-limited correlator has been reached, as observed in chapter 4. Furthermore, the shift to lower error rates is observed at smaller tree sizes than in chapter 5; a predominance of best case ALNs occurs at 4096 leaves for those using either the binary code or scatter code 2, Figures 6.4(a) and 6.6(a), and at 8192 leaves for those using the other encoding schemes, as shown in Figures 6.2(b), 6.3(b) and 6.5(b). The contrast for the temperature code and scatter code 1 is particularly pronounced since, although instances of best case performance were observed in the configuration of chapter 5 using 32768 leaves (see Figures 5.4 (d) and 5.6(d)), they did not account for the majority of trees in the set trained as here.

The distributions also demonstrate another aspect of learning behaviour which is also visible in chapter 5. Comparing the bar code, temperature code and scatter code 1 at 4096 leaves, Figures 6.2(a), 6.3(a) and 6.5(a), with the same codes at 8096 leaves, Figures 6.2(b), 6.3(b) and 6.5(b), show that an increase in tree size can have a dramatic effect on learning ability. At 4096 leaves, the error rates achieved for each code are distributed among several values with no one dominating performance. At 8192 there is a sudden shift to a predominance of best case rate trees. This is not observed with either the binary code or scatter code 2; they have already reached this stage for the lowest tree size studied. Using the configuration of chapter 5, the same effect may be observed for the bar and binary codes between sizes of 16384 and 32768 initial leaves, Figures 5.3 and 5.5 (c) and (d), and for scatter code 2 between 4096 and 8192 initial leaves, Figure 5.7(a) and (b).
Figure 6.2. Performance distribution of trained trees for the bar code.
Figure 6.3. Performance distribution of trained trees for the temperature code.
Figure 6.4. Performance distribution of trained trees for the binary code.
Figure 6.5. Performance distribution of trained trees using scatter code 1 (\(d = 1\)).
Figure 6.6. Performance distribution for trained trees using scatter code 2 (d = 2).
The remaining encoding schemes did not appear to have reached a suitable tree size to achieve this level of performance consistently using the configuration of chapter 5. The same was found for all encoding schemes when using the configuration of ALN studied in chapter 4. No predominance of best possible rate trees was observed for any tree size and, using all encoding schemes, the performance of most trees was seen to be concentrated into discrete bands intermediate between the best possible rate derived from the analysis of the input data and the previous best case performance of the hard-limited correlator. Moreover, the change observed with tree size in chapter 4 was found to be much less sensitive to initial tree size than the cases described above. Rather the prevalence of trees in higher error rate bands gradually gave way to concentrations in lower error rate bands as tree size was increased over the range studied. Furthermore, as mentioned in chapter 4, there appeared to be progressive difficulty in achieving lower error rate bands at all tree sizes.

The sequences corrected by ALNs with respect to the simple correlator to give a level of performance equivalent to 12 actual errors, were found to correspond exactly to those eliminated by the hard-limited correlator (sequences 2-76 Table 6.2). This is not surprising since with any of the encoding schemes used they will be further from the desired data ‘1’ sequence than the other error causing sequences of type 1 in Table 6.2. However, without examination of tree structures, it not clear whether the classification is made based on the retention or ignorance of intensity data. The achievement of error rates equivalent to actual numbers of errors between 12 and 0 corresponds to correction of combinations of sequences of type 1 in Table 6.2, using the information supplied by considering additional taps, shown in Table 6.3. In this case ALNs must be considering intensity information otherwise the performance limit set by hard-limiting could not be exceeded (see Tables 6.4 and 6.5). No preference for the correction of particular groups of errors was noted, despite the higher frequency of sequence 1 of Table 6.3. It is possible that the similarity of sequences 2-5 in the same table offsets this; their summed ‘0’ target frequencies being the same. Achievement of zero errors corresponds to the correction of all inputs of the type previously causing errors in the correlation receivers i.e. all sequences in Table 6.2.

6.4.5 Performance Assessment

The study of ALN performance over a range of initial tree sizes and input encoding schemes has shown that the ability to enhance performance beyond that of the previously proposed receivers, suggested by analysis of information present at the receiver input, is indeed viable. The trend for larger trees to produce lower error rates was again noted, though it was found that better performance is possible at lower initial tree sizes than using the configurations of chapters 4 and 5. Observation of the effects of input encoding on ALN error rates support the conclusions that higher distance codes with low lengths aid
adaptation by affecting more nodes during training. However, the differences observed between the different encoding schemes were much less marked than in the systems using OOCs and this has been ascribed to the higher weights of the Gold-like spreading sequences.

The mechanism by which performance is improved again depends on the retention of intensity-based information and the supply of additional information regarding the structure of interference patterns causing errors in the correlation receivers. This is achieved by encoding the intensity of the composite input sequence and providing extra taps at the receiver, so that it may consider not only the structure of the desired sequence, but also that of the sequences of other system users, in deciding the nature of the received signal. It has been shown that, using the configuration of this chapter, ALNs can be trained to completely overcome the effects of MAI. This has been attributed to the higher weights of Gold-codes which provide more information regarding the interfering users than the OOCs and thus a better context for classification. A summary of the comparative performance of the receivers considered in this chapter is given in Table 6.8

<table>
<thead>
<tr>
<th>Receiver</th>
<th>Simple Correlator</th>
<th>Hard-limited Correlator</th>
<th>Best Case ALN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Rate</td>
<td>2.00 x 10^{-2}</td>
<td>1.67 x 10^{-3}</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 6.8. Performance comparison under worst case interference conditions for an optical CDMA system employing Gold-like codes.

The same comments regarding efficiency of implementation in terms of tree size as made at the end of chapters 4 and 5 again apply, notwithstanding the fact that, using Gold-like codes, lower error rate trees were easier to produce using much smaller initial tree sizes than in the systems employing OOCs.

6.5 Summary

Results have been presented for optical CDMA systems employing Gold-like codes in place of the OOCs used in chapters 4 and 5. The performance of the simple correlator was shown to be worse than in the OOC based systems, as expected, while that of the hard-limited correlator was found to be improved. Once more, an analysis of input sequences causing errors with the correlation receivers was used to investigate bounds on performance, showing that:

- Using only the chips of the desired user's sequence limits performance to be identical to that of the hard-limited correlator, as was the case with the OOC based
systems.

- The potential for significant reductions in error rate over the correlation receivers exists, provided additional temporal and intensity information regarding interfering users is taken into account. Moreover, in contrast to the systems employing OOCs, it is possible for MAI effects to be completely eliminated;

- Using only additional temporal information about the interfering users again limits performance to be identical to that of the hard-limited correlator. Again hard-limiting is demonstrated to discard useful information.

A study of the results produced by simulation of ALNs then demonstrated that:

- ALNs were capable of realising the improvements in error rate suggested by the analysis of the input data. All MAI effects causing errors with the correlation receivers were eliminated;

- For a given input encoding scheme, ALN performance improved with increasing initial tree size. Lower error rates were obtained at smaller initial tree sizes than in the systems using OOCs;

- At small initial tree sizes, ALNs using high distance, low length input encoding schemes produced the lowest error rates. However, in contrast to the OOC based systems, the performance of all intensity preserving encoding methods was seen to converge to best case performance as initial tree size increased;

- The higher weight of the Gold-like codes enables superior ALN performance than in systems using OOCs. More information is available from the interfering sequences to enable a better classification. The incorporation of this is reflected by higher final tree sizes than achieved in any OOC system using the same initial tree size.
Chapter 7  Conclusions

7.1 Introduction

This work has investigated the application of Adaptive Logic Networks (ALNs) to enhance performance in an optical Code Division Multiple Access (CDMA) Local Area Network (LAN) environment by reducing the bit error rate for a given number of concurrent users with respect to the correlation receivers conventionally employed in such systems. The approach has been to allow an ALN to consider both the structure of the desired and interfering users’ sequences in deciding the sequence received rather than considering the desired sequence alone in an assumed interference distribution or by ignoring Multiple Access Interference (MAI) altogether. Another motivation behind the use of neural network techniques has been that, being non-parametric, they make no assumptions regarding the statistical nature of input signals in their classification. This is useful when interference distributions cannot easily be modelled, such as occurs in CDMA systems using sparse codes or those using low length Pseudo-Noise (PN) codes and low numbers of users. Furthermore, they are capable of forming more complex decision functions than the simple sum and threshold approach of correlation receivers, which becomes even more limited when used in the “positive” unipolar systems considered where the sum to zero ability is lost.

To this end, a model of an optical CDMA channel was developed in order to permit simulation of signals at the input to receivers in such systems. The input signals were generated for users employing an On-Off Keying (OOK) modulation format and assuming that, whilst transmission between users was asynchronous at the bit level, it was synchronised at the chip level. It was also assumed that external sources of noise were negligible i.e. MAI was dominant. Consideration was given to the assignment of two contrasting types of spreading sequences to system users: Optical Orthogonal Codes (OOCs), developed specifically for use in incoherent optical CDMA systems, and Gold-like codes, a form of PN sequence usually associated with conventional coherent (copper- or radio-based) systems.

The input signals generated were first used to test simulations of the previously proposed simple and hard-limited correlation receivers in order to provide a benchmark for ALN performance. The types of inputs causing errors with these receivers were studied in order to establish the limits set on performance by the information contained within them and to examine ways in which this information might be used to reduce receiver error rates.
This formed the basis for the evaluation of the ALN receivers, also carried out by simulation, and an investigation into their associated training parameters to establish ways of minimising their error rates efficiently.

7.2 Summary of Results

The simulation results obtained using the simple correlation receivers were useful in several respects. Firstly, the results obtained in the systems employing OOCs agreed with those expected from the theory and therefore confirmed that the input generation methods were working correctly. More importantly however, they served to highlight the particular limitations of considering only the desired user's sequence and also of the sum and threshold approach to signal classification. Analysis of the input sequences causing the receiver output threshold to be exceeded showed that errors could be generated in two basic ways. One method was found to involve the summation of other users' code sequences to create interference patterns mimicking the desired user's data '1' sequence, when in fact no such code was being sent. In this case interfering users contribute single chips equally to all correlator taps and as such there is no way of distinguishing these patterns from a genuine data '1' sequence – the correlator only considers the sequence it expects to receive and does not take into account further information regarding the presence of other users. As such, using this number of taps, the occurrence of these types of errors is inevitable.

The second method by which errors were seen to arise exposes another shortcoming of the sum and threshold approach. In this case the threshold is reached, despite the fact that the desired sequence template represented by the correlator taps is not correctly filled. This occurs because the lack of chips at some correlator taps is offset by the summation of optical intensity from multiple chips at others. Clearly there is information present in terms of the absence of chips and in intensity values of occupied taps that could be used to differentiate such cases from the receipt of the desired sequence. Such errors are therefore correctable, despite the fact that only the taps associated with desired sequence are examined. However, this is not possible using the simple correlator, since it cannot discriminate how the threshold is achieved.

The results obtained by the simulation of the hard-limited correlation receivers were consistent with those expected; both systems employing OOCs and Gold-like codes showing significant reductions in error rates. Hard-limiting the input is one method of eliminating the second type of error described above. Contributions of multiple chips at single taps are clipped back to single chips and the absence of chips at other taps means that the summing operation no longer achieves the output threshold, provided it is set at

115
the weight of the desired code. The remaining error rate arises from the generation of errors by the first method described above.

Examination of the inputs causing errors at the outputs of the correlation receivers formed the basis for developing performance bounds for CDMA receivers in general, given a particular form of input, and for establishing the potential for further reduction in error rate over that of the hard-limited correlator in each system. In the case of the simple correlator, the inputs do not undergo preprocessing and thus were used to determine the limits imposed on performance by the information contained in the input chip data presented, given the amount of this information accessible to a notional receiver through its particular input tap configuration. This approach was based on the fact that conflicts in the data set requiring the classification of an input both as a '1' and a '0' will give rise to errors in any receiver considering this form of input. As such, the best possible performance obtainable is by the correct classification of inputs associated with unique target pairs and the minimisation of remaining errors by classifying sequences with conflicts, as corresponding to the higher frequency target. Using this technique, it was confirmed that a method exists using only the taps associated with the desired user's sequence, either OOC or Gold-like, whereby significant reductions in error rate are possible by eradication of errors caused by summation of optical intensity, since they are uniquely associated with targets of '0'. Hard-limiting the input, as previously described, provides one method of achieving this. However, it is conceivable that this could equally be achieved by retaining and utilising intensity-based information instead of hard-limiting the inputs. Nevertheless, the results obtained using either method would be the same and therefore the best possible performance of a receiver using only the taps associated with the desired user's sequence is constrained to be the same as that of the hard-limited correlator.

This process was then extended by examining the input composite sequences causing errors with the simple correlator at chip positions unavailable to receivers considering only the positive chips of the desired user's code sequence. Since the sequences causing errors by localised summation of optical intensity had already been shown to be correctable, attention was concentrated on the sequences with conflicting targets i.e. those mimicking the desired data '1' sequence at the relevant chip positions. The aim was to determine whether the provision of additional chip data, containing more information regarding the structure of interfering users and accessible via the use of extra taps at a receiver, could be used to resolve the target conflicts and thus allow further correction of errors arising due to MAI. Results of the analyses showed that in both the systems employing OOCs and in those using Gold-like codes, correction of these types of inputs was feasible. The potential for improvement was found to be dependent on the spreading
sequences used and the positions of the additional taps examined. The configuration of
the OOC based system in chapter 5 was found to offer more capacity for improvement
than that of chapter 4, since the former always had access to the full structure of the
interfering codes whereas the latter did not. However, MAI was still found to cause target
conflicts and therefore a finite error rate would be expected in OOC-based systems. By
contrast, the analysis for the Gold-like code system in chapter 6 showed that
consideration of the additional taps meant that it was possible to completely overcome
errors caused by MAI.

A similar analysis was performed for the sequences causing errors with hard-limited
correlation receivers. However, the performance limits are now set by the input data
under transformation and a receiver using this form of input may consider only temporal
information in classification of input signals. In the transformed set there are no longer
any sequences causing errors by localised summation of optical intensity, consequently
attention was focused solely on the those inputs mimicking the desired sequence at the
relevant taps to see if, by considering further chips, conflicts could once more be
resolved. It was found that the conflicts remained for all spreading sequences and receiver
input configurations studied, despite the provision of additional temporal data. This was
found to be caused either by the mapping of multi-level sequences associated with targets
of ‘1’ to sequences invariant under the operation and previously uniquely paired with
targets of ‘0’ or by the mapping of distinct multi-level sequences with unique targets to
identical boolean sequences in the transformed input space. It was therefore demonstrated
that an operation designed to eliminate certain types of errors actually precludes further
error correction by generation of conflicts in the transformed data set and that under hard-
limiting conditions performance cannot be improved by consideration of extra taps.

The results produced by simulation of ALNs showed that the performance improvements
suggested by the analysis of input data sequences were realisable. Those obtained by
ALNs trained using hard-limited inputs confirmed the limits imposed on performance by
this encoding technique. Indeed, in the systems employing OOCs, instances of final trees
composed of a single AND gate were found. This corresponds exactly to the hard-limited
correlator model and indicates that the additional redundant information supplied has
been ignored. All other ALNs trained using encoding schemes permitting the retention of
intensity-based information were found to be capable of superior performance than the
correlation receivers.

However, the degree of improvement did not necessarily correspond to the best possible
error rate derived from the analysis of conflicts in the training sets used. In general,
performance of ALNs in all systems considered was seen to improve with increasing
initial tree size and, for a given tree size, the best results were obtained with input
encoding schemes having high hamming distances and low lengths. However, using the OOC-based system synchronised to the ‘1100000000000000’ code phase (chapter 4), achievement of best case error rates was found to be particularly difficult, even at the largest tree sizes. This situation was improved by changing to the configuration synchronised to the ‘0000000110000000’ code phase (chapter 5) where it was found that given a suitable input encoding scheme a predominance of best case trees could be achieved and at lower initial tree sizes. Replacing the OOCs with Gold-like codes (chapter 6), it was found that all intensity preserving input encodings were capable of producing predominantly best case ALNs using even smaller initial tree sizes.

The best case error rates attained by ALN receivers were found to correspond exactly to the limits set on performance by the results of the analyses of the input sequences for each system summarised above. Of the configurations considered, only in the systems using Gold-like codes were the effects of MAI able to be eliminated completely by ALNs. In the systems using OOCs, performance was still found to be limited by instances where the overlap of interfering code sequences was indistinguishable from the desired user’s code sequence. It was shown that the effects of MAI can be reduced by careful selection of the phase of the OOC to which the receiver is synchronised, as demonstrated by the achievement of lower error rates using the configuration of chapter 5. Nevertheless, as expected, the error rate remained finite even with complete access to the full structure of the interfering users’ signals.

7.3 Conclusions

The results obtained by simulation of ALNs have shown that receiver performance can be improved by consideration of the signals of interfering users in deciding the nature of the data bit received. Details of the relative timing, data bit value and code sequences of interfering users are not explicitly specified to the receiver nor are any assumptions made about the distribution of MAI. Rather the ALN learns details of the structure of interference based on information contained within the input data. Indeed it has been demonstrated that the level of performance achieved need only be constrained by the limitations imposed by the information contained in the input data. For this reason, it has been shown that achievement of lower error rates than those obtained with the hard-limited correlator, the previous best case receiver, requires the consideration of both additional temporal and intensity-based information. Although hard-limiting the input allows the elimination of certain types of errors, further reduction in error rate by provision of supplementary temporal data alone is prevented by the creation of conflicts in the transformed input space. Hard-limiting therefore removes information that could be used to perform a better classification before it can be used.
Given the retention and utilisation of intensity-based information via suitable encoding methods, it has been shown that the performance of the hard-limited correlator may be surpassed. This is significant since it relies on the correct classification of input signals where the interference is such that it reproduces the desired user’s data ‘1’ sequence at the relevant tap positions yet the desired transmitter is sending nothing i.e. the desired output is ‘0’. This demonstrates the superior ability of ALNs to form more complex decision functions than the simple sum and threshold type of the correlation receivers, since such classifications must be of a conditional nature, with the additional chip information providing the context. This ability would become even more important as the number of users grows. Errors arising with correlation receivers would then be predominantly caused by interference contributions at all taps and in learning to correct these errors ALNs would have to use information derived from the additional tap data rather than the absence of chips at taps associated with the desired user. Furthermore, interference would no longer be limited to one chip per tap.

The ease with which the performance limits due to the data set analysis may be attained and the degree to which performance may be improved with respect to the correlation receivers was observed to depend on both the ALN training parameters and the amount of information provided by the additional input taps. The general trend for larger tree sizes to permit higher levels of performance is not surprising. Larger tree sizes are less likely to be limited by connection restrictions to the encoded input vectors and should also be able to form more complex decision functions better suited to the task. The latter point is likely to be most important in incorporating the cases where interference mimics the desired user’s code sequence. The observations made in chapter 6 that an increase in tree size can give rise to a sudden predominance of best case trees, reinforces these comments and suggests the achievement of a size threshold whereby learning of the task becomes much easier.

Tree size alone cannot account for the level of performance attained, since the error rates achieved at a given size were found to vary, depending on the input encoding schemes and types of spreading sequences used. Although in the systems using Gold-like codes the differences in performance achieved using different encoding schemes were less pronounced and converged with tree size, in both systems using OOCs and Gold-like codes the ordering of input encoding schemes in terms of their effectiveness was basically the same. The best results were consistently achieved by higher hamming distance encodings with low encoding lengths. The higher the distance between encoding levels, the greater difference there is between encoded input vectors to permit their distinction. During adaptation this means that, for a given input encoding length and tree size, a higher distance code will affect more nodes and therefore there is the opportunity
to form a more complex function. Similarly, lower lengths mean that there are more connections to each element of an input vector than with higher length encodings for a given tree size, since there are effectively fewer input variables to consider. Therefore, a change in such an element will again affect more nodes during adaptation. These conclusions are supported by the fact that larger final tree sizes are obtained with the more effective encoding schemes.

The remaining differences in performance observed between the systems studied must be attributable to the different structure of the spreading sequences themselves since, all ALN receivers considered input composite sequences over a similar length. In the OOC-based systems, the sequences have weight 2 and interfering users may therefore contribute at most 2 chips each to an input composite sequence. In the presence of two interfering users and the absence of the desired user’s sequence, an input sequence may therefore contain a maximum of 4 chips. Given the worst case scenario of interference mimicking the desired user’s data ‘1’ sequence, there are at most 2 chips available to enable distinction from the data ‘1’ sequence and permit correct classification as a data ‘0’. Using the configuration synchronised to the ‘1100000000000000’ phase of the desired OOC (chapter 4), either one or both of these chips may be unavailable to an ALN, due to the choice of tap window. Furthermore, although in such cases there may be chips present from the subsequent data symbol’s code sequence, which give positional (temporal) information regarding the preceding interfering sequence, it does not provide any information regarding the nature of the data bit since each consecutive data bit is independent and equally likely to be ‘1’ or ‘0’. As such, the performance obtained is not as good as it might be and the need to carefully consider the range of input taps selected is emphasised.

By synchronising to the ‘0000000110000000’ phase (chapter 5), the time window is such that when the desired data ‘1’ sequence is mimicked all the other chips of the interfering sequences are available and no positive chips are contributed from either the sequences of preceding or succeeding data symbols. Therefore, the maximum possible detail regarding the interfering users is provided, enabling better discrimination than with the previous configuration (chapter 4) and a further reduction in error rate. Nevertheless, the structure of the OOCs is such that there remain cases where the overlap of two interfering users mimics the desired user’s sequence in the presence of a single interfering user’s sequence and thus performance is still limited by MAI.

In the systems using Gold-like codes, the two interfering users may contribute a maximum of 6 chips each to the input composite sequence. Therefore, despite the fact that the total number of input taps considered is similar to the OOC-based system, the amount of information provided for the discrimination of input sequences from false data
'1' sequences is greater. The OOC systems considered 14 extra taps with respect to the correlation receivers but were limited to a maximum of 2 chip contributions in this field to make a decision. Using Gold-like codes of length 15, effectively only 7 extra taps were considered, since the desired code has a weight of 8, yet interfering codes can contribute up to 12 chips over the full 15 chip length. Therefore, even if interference mimics the desired user's code, there could be up to 4 chips out of 7 to provide the conditional information required for a correct classification. The higher weight of the Gold-like codes effectively builds in a larger distance property between unencoded input vectors than with the OOCs and this is amplified by the input encoding mechanism. The fact that higher levels of performance are achieved at lower tree sizes reflects this distance advantage over OOCs and the fact that all encoding schemes achieve good results for larger tree sizes suggest that it is the weight property of the sequences, rather than that of the input encoding scheme, that has most effect. Moreover, the ability of ALNs to completely eliminate the effects of MAI in systems employing Gold-like codes indicates that such sequences are suitable for use in an optical CDMA environment, provided the presence of other users' signals is taken into account.

The relative performance of the ALN receivers using the different spreading sequences and receiver configurations is also reflected in the final sizes of trained trees. For given initial sizes and encoding schemes, the final tree size observed using OOCs was found to be greater using the configuration of chapter 5 than using that of chapter 4. Similarly, the tree sizes produced using Gold-like spreading sequences in chapter 6 were greater than either of the systems using OOCs. This reflects the incorporation of the greater knowledge available regarding the interfering users in the systems of each successive chapter.

This work has investigated the ability of ALNs to detect a desired user's signal in the presence of the signals of other interfering users. The approach used has been to train the ALNs with an exhaustive data set comprising the desired user's data '1' and data '0' sequences in the presence of all possible combinations of interfering users' data values and relative timeshifts, given the assumption of chip synchronism between users. This means that the number of sequences presented grows with the number of users and the problem is exacerbated by allowing asynchronous transmission on the network. Under asynchronous conditions, the relative time delays of interfering users must be taken into account and, as such, interference from other users typically consists of partial contributions from the two code sequences of consecutive data symbols. Such a data set rapidly becomes unmanageable for larger numbers of users and longer codelengths. Furthermore, if the chip synchronous restriction were to be removed then the number of sequences would become infinite. In related work proposing the use of neural networks as
receivers in conventional (non-optical) CDMA systems, this problem has been circumvented by concentrating on the less complicated case of symbol synchronous users where the relative time delays between users are assumed to be zero [Paris, 1988][Aazhang, 1990]. In other work, where asynchronous users have been considered, it has been assumed that, while unknown the relative time delays and phases of interfering users’ signals are constant during training, these signals are then evaluated in the presence of additive noise during testing [Aazhang, 1992]. In both of these studies attention was focused on low length CDMA codes and low numbers of users, in some instances both being lower than the codelengths and number of users examined here.

The approach taken here is not to advocate the exhaustive training of neural network receivers since, given the comments above, this is highly impractical and even if it were not, significant changes to a real network would require extensive retraining. The aim of this work has been to demonstrate that ALNs can be used to eliminate the effects of MAI in optical CDMA networks and that such receivers do not require complicated design or the specification of interference conditions; they have the ability to adapt to prevailing conditions based on the information contained within their inputs. The use of an exhaustive data set has been to show that this is possible over the range of worst case interference conditions and that MAI need not limit performance.

7.4 Further Work

One obvious area for further research would be to investigate methods of producing more efficient low error rate trees by reducing their size. In order to do this a Genetic Algorithm (GA) [Goldberg, 1989][Holland, 1992] could be used to search the space of either the initial node assignments or the initial connection configuration of leaf nodes to the inputs and their complements for smaller fixed initial tree sizes. If the GA were configured to do both at the same time, then it could actually be used in place of the ALN learning algorithm used. This idea could be extended to allow for a dynamic tree size during learning by using Genetic Programming (GP) techniques [Koza, 1992]. GP is itself based on tree structures and its search method involves the exchange of subtrees between pairs of trees. This method is more in the spirit of ALNs and would lend itself much more readily to optimisation or training of ALNs than the pure GA.

As well as being more efficient, smaller trees might also permit the use of graphical methods to visualise ALNs. This could be used during training, to see how the tree structure changes, and during testing, to make qualitative observations regarding the insensitivity properties of the tree. Furthermore, if sufficiently compact trees could be produced, there is the potential for rule extraction from the final trained tree.
Another potential way of improving efficiency would be to look more carefully at the taps considered by the ALN input. The ALNs in this work considered input over a contiguous range of taps. However, in a manner similar to that used for the analysis of the input sequences in chapters 4, 5 and 6, one could assign don’t care states to various chip positions and examine the effect on the conflicts in the inputs. If a tap (or range of taps) is seen to have no affect on the conflicts and their distributions, it may be discarded.

The input encoding scheme could also be tailored to suit individual input taps. This is likely to be useful with sparse input sequences, where it is required to emphasise that a small difference in unencoded input vectors can be significant, if positioned at a particular location. In this way, by using a higher distance code than usual at a certain tap, it may be possible to classify apparently similar sequences with separate targets if necessary. Care would be needed in “tuning” the tap since this would also have an effect on other input codes. However, as noted in chapter 6, an encoded input vector need not be closer to it’s desired output than its complement in hamming space, for it to be relatively easily correctly classified.

The possibilities for improvements in efficiency mentioned above would also permit more realistic simulation of ALNs in optical CDMA LANs. For example, reductions in tree size would make the simulation of non-chip synchronous cases more manageable. This could be done by increasing the temporal representation of a chip in the unencoded input to the ALN. The effect of additive noise could also be taken into consideration by extending the resolution of the intensity encoding mechanism. However, if other sources of noise can be controlled so that MAI is dominant, then the input encoding levels could be set accordingly and this is no longer necessary.

Finally, the application of neural techniques to other aspects of LAN architecture could be considered. As well as performing classifications of the type used in this work, neural networks can be used to achieve mappings of input sequences to other output sequences. Using ALNs, this could be achieved by multiple trees; one for each bit of the desired output vector. This mapping ability lends itself to applications in routers and bridges to allow inter-network communication and the support of more users.
References


[Armstrong, 1992] Armstrong, W. W. Posting to comp.ai.neural-nets newsgroup, Usenet. See also adaptive logic network mailing list archives (alnl), available by anonymous ftp from menaik@cs.ualberta.ca.


