

University of South Wales



2064769

A Study of Modulation and Error Control Coding in Thick Film Digital Magnetic Recording

DAVID ALEXANDER MYLES (B.Sc.)

A thesis submitted in partial fulfilment of the requirements to the Council
for National Academic Awards for the degree of Master of Philosophy

May 1992

Polytechnic of Wales in Collaboration with British Gas Research
Newcastle

Department of Mathematics and Computing
Polytechnic of Wales
Llanwit Road
Pontypridd
CF37 1DL

Acknowledgements

I would like to express my unreserved gratitude to my research supervisors Prof. Alan Ryley and Dr. Ron Wiltshire of the Polytechnic of Wales for their help, advice and encouragement during the course of my research.

I am also indebted to the many members of staff throughout the Polytechnic for their support and interest in my research.

I wish to thank Mr Martin Morey of British Gas' Inspection Laboratories Newcastle Upon Tyne for his Technical Guidance.

Certificate of Research

This is to certify that, except where specific reference is made, the work presented in this thesis is the result of the investigation undertaken by the candidate.

Candidate.....*DAMPHIA*.....

Director of Studies.....*A. Hughes*.....

Declaration

This is to certify that neither this thesis or any part of it has been presented or is being currently submitted in candidature for any degree other than the degree of Master of Philosophy of the C.N.N.A.

Candidate....*D.H. Meyer*.....

Chapter 1	
Introduction and Summary	1
 Chapter 2	
The Recording Process	5
 2.1 Introduction	5
2.1:1 Introduction	5
2.1:2 Schematic Diagram of the Recording Process	5
2.2 The Recording Medium as a Channel	11
2.2:1 Properties of a Channel	11
2.2:2 Timing Windows	12
2.2:3 Bitshift	13
2.3 The Recording Process	14
2.3:1 Pulse Shape	14
2.3:2 Channel Errors	19
2.3:3 The System Clock	21
2.3:4 Clockloss and Non-Clockloss Errors	22
2.4 Non-Clockloss Error Mechanisms	23
2.4:1 Noise	23
2.4:2 Inter Symbol Interference	26
2.4:3 Crosstalk and Overwrite	30
2.5 Clockloss Error Mechanisms	31
2.5:1 Drop-outs	31
2.5:1(i) Tape Defect Drop-outs	31
2.5:1 (ii) Drop-outs Caused by Head Errors	33
2.5:1 (ii) Drop-out Effects	36
2.5:3 Jitter	38

Chapter 3	
Recording Media and Devices	43
3.1 Recording Principles	43
3.1:1 Introduction	43
3.1:2 Magnetic Recording Media	44
3.1:3 Fixed Head Tape Recording	46
3.2 Rotary Head	49
3.2:1 Principles of Rotary Head Recording	49
3.2:2 Digital Video Recording	51
3.2:3 R-DAT Rotary Digital Audio Tape	52
3.3 Floppy Discs	60
3.4 Thin Film Recording	62
3.4:1 Thin Film Structure	62
3.4:2 The Winchester Disc System	63
3.5 Optical Discs	66
 Chapter 4	
Error Correction Coding	71
4.1 General Principles	71
4.1:1 Introduction	71
4.1:2 Error Correction by Adding Redundancy	71
4.1:3 Information Rate	72
4.1:4 Hamming Weight	73
4.1:5 Hamming Distance	73
4.1:6 Varashamov-Gilbert Bound	75
4.1:7 Perfect Codes	76

4.1:8 Parity Check Codes	77
4.1:9 Error Syndromes	78
4.1:10 Aliasing	78
4.1:11 Codes for Asymmetric Channels	79
4.1.12 Galois Field Theory	81
 4.2 Single Error Correction:The Hamming Code	 82
4.2:1 Hamming Code	82
4.2.2 Example of the Hamming Code	84
4.2:3 Extended Hamming Code	85
 4.3 Multiple Error Correction Codes	 86
4.3:1 Introduction	86
4.3:2 Reed Muller Codes	87
4.3:3 BCH Codes	89
4.3:4 Block Encoding of the BCH Code	91
4.3:5 Irreducible Polynomials over Galois Fields	91
4.3:6 The Cyclic Property of BCH Codes	92
4.3:7 Cyclic Encoding of BCH codes	92
4.3:8 Erasure Decoding by BCH codes	94
4.3:9 Bursty Errors	95
 4.4 Reed Solomon Codes	 97
4.4:1 Introduction	97
4.4:2 Burst Error Correction	97
4.4:3 Fire Codes	98
4.4:4 B-adjacent Codes	99
4.4:5 Definition of Reed Solomon Code	100
4.4:6 Example of Reed Solomon Encoding	101
4.4:7 Reed Solomon in High Integrity Recording	103

4.5 Convolutional Codes	105
4.5:1 Introduction	105
4.5:2 Encoding of a Convolutional Code	106
4.5:3 Catastrophic Error Propagation	107
4.5:4 Decoding and the Viterbi Algorithm	109
4.6 Coding Enhancements	110
4.6:1 Interleaving	110
4.6:2 Cross-Interleaving	111
4.6:3 Concatanted Codes	113
 Chapter 5	
Modulation codes	119
 5.1 Introduction	 119
5.1:1 Introduction	119
 5.2 Modulation Coding Principles	 120
5.2:1 Rationale for Modulation	120
5.2:2 Parameters of Modulation	121
 5.3 Standard Groups of Modulation Code	 123
5.3:1 Runlength Limited Codes	124
5.3:2 Non-Return to Zero Codes	127
5.3:3 Group Codes	127
5.3:3 i)Rice Code	128
5.3:3 ii)Octal Coded Binary	130
 5.4 Codes considered in this Thesis	 131
5.4:1 Manchester Modulation or FM	131
5.4:1 Miller Code or MFM	132

5.5 Enhancements on the Miller Code	133
5.5:1 Miller ² Code or M ² FM	133
5.5:2 Zero Modulation	135
5.5:3 Comparison of the Miller Codes	137
5.6 Modulation Codes Used in Fixed Head Recording	138
5.6:1 Codes in Fixed Head Recording	138
5.6:2 Zero Modulation	139
5.6:3 Jacoby and Kost's (1,7) code	140
5.7 Modulation Codes in High Integrity Media	142
5.7:1 The Advantages of Group Codes in High Integrity Media	142
5.7:2 EFM: The Modulation in Optical Recording	143
5.7:3 Alternative Modulation for Optical Recording	147
5.7:4 8/10 Modulation in R-DAT	148
5.8 Combined Modulation and Error Correcting Codes	152
5.8:1 Reasons for Combined Codes	152
5.8:2 Combined ECC and Runlength Limited Codes	153
5.8:3 Error Correction Using Runlength constraints	154
5.8:4 Trellis Codes	156

Chapter 6

Theoretical Results

6.1 Introduction	163
6.2 Noise Induced Bitshift	164
6.2:1 The PDF of Noise Induced Bitshift	164
6.2:2 Comparison of the Noise PDF's	167
6.2:3 The Effect of Windows on the Noise induced Error Rate	172
6.2:4 The Auto Correlation of Noise	175

6.3 Theoretical Results involving ISI	182
6.3:1 The Effect of ISI in Magnetic Recording	182
6.3:2 Formulation of ISI	184
6.3:3 The Effect of the Choice of Modulation Code on ISI	191
6.3:4 Correlation of ISI	193
 6.4 Modulation Coding Results	 194
6.4:1 Introduction	194
6.4:2 Method of Decoding Manchester Modulation Codes	194
6.4:3 Miller Decoding	197
 6.5 Error Correcting Codes	 199
6.5:1 Aliasing	199
6.5:2 Bit and Byte Error Correction in Non Linear Codes	200
6.5:3 Error Correction with Runlength Constraints	201
 Chapter 7	
Computer Simulation Design and Implementation	208
 7.1 Introduction	 208
 7.2 Computer System Design	 209
7.2:1 Design of Filter for Noise Correlation	209
7.2:2 Design of Model for BCH codes	212
7.2:3 Design of Model for Reed Solomon codes	215
7.2:4 Design of Model for Noise and ISI	218
7.2:4(i) Model for Uncorrelated Noise	218
7.2:4(ii) Model for Correlated Noise	218
7.2:4(iii) Model for ISI	219
7.2:5 Improvements in the Error Rate Due to Demodulation	221

7.3 Results for BCH coding	225
7.3:1 Uncorrelated Noise with ISI	225
7.3:2 Correlated Noise with no ISI	226
7.3:3 Effect of adding ISI on BCH encoded sequences	230
7.4 Results Involving Reed Solomon Error Correction Encoding	232
7.4:1 Noise Only No ISI	232
7.4:2 The Effect of the addition of ISI to Reed Solomon Codes	238
7.5 Results Involving Ferrieria codes	
7.5:1 Bitflip	242
Chapter 8	
Conclusions and Further Work	247

Abstract

This thesis is concerned with the application of Modulation and Error Correction codes to Digitally Recorded Magnetic Data.

The effects of these coding schemes were measured against bitshift errors which are induced by different Non Clockloss Error Mechanisms. These Error mechanisms are Noise and Inter Symbol Interference, ISI. The Thesis has an extensive literature survey which can be divided into four main areas: Error Control Coding, Modulation Codes, Recording Techniques and the Recording Process.

The theoretical and experimental results involving the Non Clockloss Error Mechanisms and their effect on the data, were derived by Monte Carlo computer simulation. Four channel models were used, for uncorrelated noise, correlated noise and the convolution of these with ISI. The model for ISI was developed assuming a symmetrical replay pulse for two separate types of modulation coding. Two different Error correcting codes were used, these are the linear BCH code and the Non-Linear Reed Solomon Code.

The results of this thesis conclude that the effects of Tape Noise and ISI are independent pulse to pulse. A strategy was developed for pulse reconstruction for Manchester Modulation code. The phenomenon of bitflip was found to be more prevalent in fixed pulse rather than fixed rate codes.

Chapter 1

Introduction and Summary

All Recording media whether a disc or a tape can be regarded as a channel between the input and output heads. The characteristics of the recording channel vary according to the media used. There are three types of Recording Media in common use:

Thick Film Magnetic Recording. Which is cheap erasable and robust, though giving only a limited media life, low data densities and high error rates,

Thin Film Magnetic Recording. This is erasable with high data densities and low error rates. However it is not robust. A thin film drive system when in operation cannot be moved for fear of the heads crashing into the media, and is comparatively expensive to buy,

Optical Recording. This has high data rates and low error rates in addition to being robust and offering a permanent method of storing data. However an Optical recording cannot be erased and overwritten. In addition optical technology is still comparatively expensive.

Optical Recording is different to Magnetic Recording as it stores information by the length and the gaps between a series of pits etched into the surface of the media. In Magnetic Recording a transition in the current of a coil induces a reversal in the magnetic flux in the media. This flux reversal is translated into a voltage pulse at the recording head. This work is primarily concerned with thick film magnetic recording. This work proceeds by theoretical analysis and computer simulation.

The simulation results confirm the theory and allow trends for lower error rates to be drawn.

The work can be divided into two sections; the first consists of a survey of the salient points in Magnetic recording:

- (i) A model of the Recording Channel detailing how the recording is stored on the Media. In this section the actual physical process of recording is discussed. This is followed by a discussion of the inherent mechanisms by which errors are propagated in the magnetic recording system. The effect of recording density on the Error Mechanisms and their propagation is discussed.
- (ii) The different methodologies for recording and storing data are discussed. These are compared and contrasted for their differing advantages and disadvantages for high density recording. This includes a discussion of media, recording heads and the mechanics of the recording system.
- (iii) Several different schemes for error correction and detection are examined and compared. These can be partitioned into two distinct groups, Linear and Non-linear codes. Linear codes treat the data as a series of single bits each bit being independent from the next. Non-Linear codes group the data into a block of bits and the algorithm then encodes over these blocks.
- (iv) A comprehensive study of Modulation codes was conducted, both

direct and block codes were considered. These were contrasted and compared for their runlength constraints, ability to reduce error propagation and their resistance to the common error mechanisms. Those codes which combine both the runlength properties of Modulation codes and Hamming distance properties of error correction codes are also discussed in the section on modulation codes.

The second section consists of the theoretical results which were derived and tested by computer simulation. The principle areas of interest to the investigator include:

- (i) The correlation pulse to pulse of the two primary non-clockloss error mechanisms. The effect of this correlation was measured against both Linear error correcting codes, BCH, and Non-Linear block code Reed Solomon;
- (ii) The phenomenon of code aliasing was examined. This occurs in error correction codes where the number of errors is greater than that which will be corrected by the Hamming distance. The code then "correct" the bits in error shown by the syndrome whether these are erroneous or not;
- (iii) The ability of some modulation codes to allow the recovery of a transition lost from the centre of a data window. This is achieved by relying on the redundancy of the code caused by the extra transitions which may be present at the end of the data window.

However there are a number of assumptions and simplifications used in this work

these are:

- (i) The assumption of a symmetrical replay pulse ensures that the model for ISI induced bitshift is not stringent enough, as the symmetric replay pulse leads to cancellation in the ISI from pulses directly opposite the pulse in question;
- (ii) Due to the constraints on computing time, most simulation results are not presented at operational error rates. However these results are published and may act as a comparison. Indeed the models for the higher error rates may be extrapolated from the theoretical results and those error rates achievable within the constrained computing time.

Chapter 2

The Digital Recording Process

2.1 Introduction to Chapter 2

2.1.1 Introduction

This chapter offers an account of the recording process, and the mechanisms which may cause error in the recording system. A schematic diagram of the recording process is given in section 2.1.1. The blocks of this diagram are then described, giving the purpose of each one. However several areas of the recording system are dealt with in greater detail in sections. 2.2-2.5 These include:

- (i) the method by which data is recorded onto magnetic media;
- (ii) the perception of the recording media as a channel;
- (iii) a description of errors in the readout of data;
- (iv) a discussion of the importance of good clock recovery;
- (v) a discussion of the mechanisms by which the errors are generated.

2.1.2 Schematic Diagram of the Recording Process

A block diagram of the recording process is shown in Figure 2.1

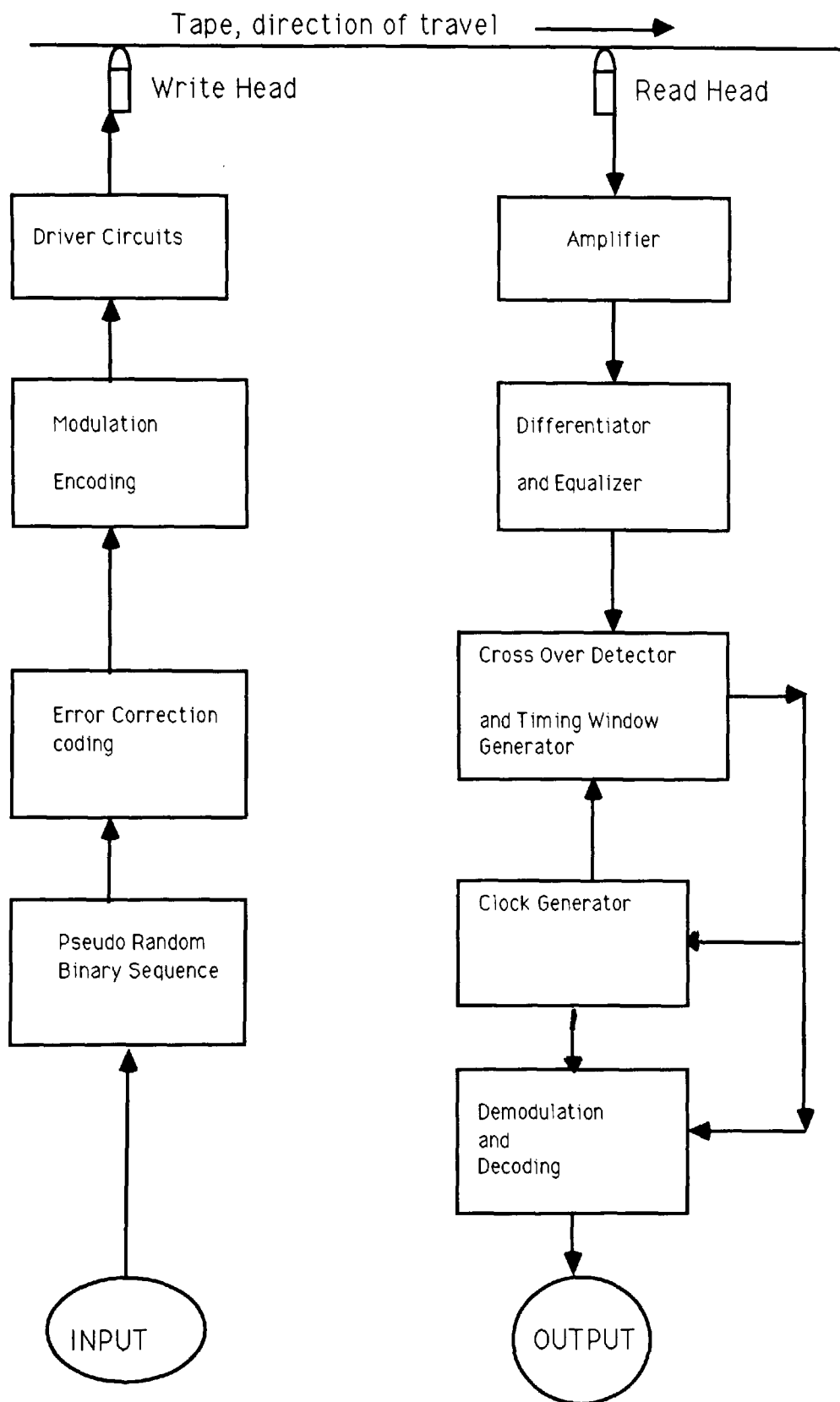


Figure 2.1 Schematic Diagram of the Recording Process

In the above diagram the function of the individual boxes are given by:

Input Data as a Pseudo Random Binary Sequence

All the statistics of the considered error mechanisms depend upon a stationary random distribution of magnetic pulses upon the tape. Hence a Pseudo Binary Sequence is used for the Raw Data.

Error Correction Code

Error Correction Coding adds redundancy to compensate for the effect of errors introduced into the recording process. This may be done using several algebraic methods which are described in Chapter . However in selecting the type of error correction code it is first necessary to derive a model of the tape errors.

Modulation Codes

The modulation codes match the raw encoded data to the channel. They add redundancy to the original data and hence create a facility for error detection and correction in addition to that given by the error correction code. The modulation code also allows the frequency response of the recorded data to be matched to that of the channel. The system clock is established at this juncture, to enable the recording to avoid bit-slip errors.

Driver Circuits

The driver circuits convert the modulation code to a series of electric pulses in the recording head.

Recording Heads

In the recording head a step change in current induces a change in the magnetic flux in the tape. The recording is made at the trailing pole of the recording head[2.1], as the magnetic flux falls below the critical threshold required to alter the state of the particles.

Recording Media

The recording tape is assumed to be a thick film (greater than $3\mu\text{m}$ in depth). Recording occurs in both the longitudinal and perpendicular planes. However in most models of the pulse only the longitudinal plane is modelled. The tape moves past the heads at a fixed speed. The tape is the dominant source of error in the recording system [2.1]. To give the best results the media should not be saturated with ϕ_{ux} as this will cause fringing fields around the head as well as crosstalk from the adjacent recording tracks [2.2].

A binary digital recording has only two possible states, represented by a one and a zero. In magnetic media these are represented by a magnetisation of the particles. This is shown below in Figure 2.2 [2.4]. The arrows represent direction of flux of the particles and TP represents a transition point

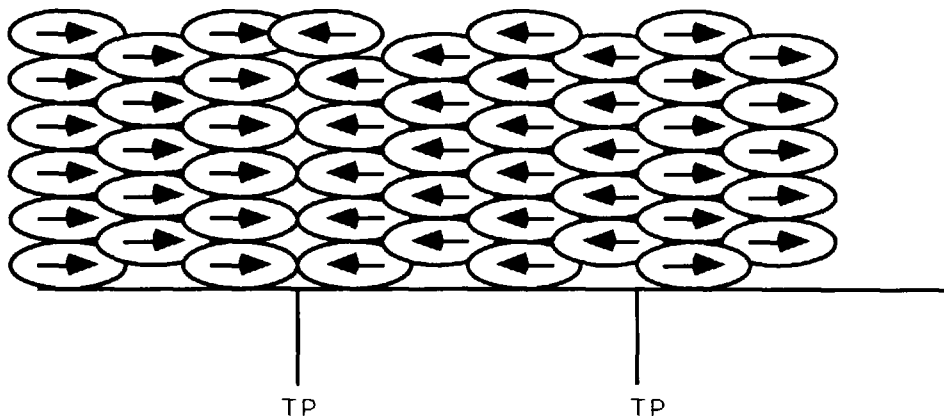


Figure 2.2 Magnetisation of Particulate media

Read Head

At the read head the change in magnetic flux induces a voltage pulse which is proportional to the rate of change of the current in the write head.

Equaliser and Amplifier

The signal from the read head is first amplified and then equalised. Equalisation reduces the channel errors, equalises the effect of the channel and is used in the pulse shaping which is necessary for differentiation.

Differentiator

Differentiation is used in modern recording systems instead of integration, as differentiation slims the replay pulse while integration broadens it. Pulse slimming is necessary as the dominant error source is Inter Symbol Interference,

(ISI) which is reduced by slimmer pulses[2.2,2.3]. However the differentiator increases the level of noise which would have been reduced by integration. In addition the differentiator acts as part of the equaliser, as it is a high pass filter.

Detection

The detection of the differentiated pulse is through a zero crossing detector. Recording errors are manifested by the zero-crossing being translated out of the region of tolerance. The phenomenon of translation is called bitshift. The region of tolerance is called the Timing Window.

Clock Generation

It was stated above that the clock generation is achieved by the modulation code. For good clock generation the modulation code should have a relatively uniform length of gap between neighbouring pulses.

Demodulation and Decoding

Demodulation and Decoding are the inverse procedures from modulation and Error Control Coding.

2.2 The Recording Media as a Channel

2.2:1 Properties of a Channel

The theory of digital recording discussed in this thesis may be regarded as an analogue of communication theory because both are concerned with transmitting and processing of signals in channels. However there are several important differences between the two topics.

Firstly in communication theory the channel is normally regarded as being linear while it has been shown by that the channel for data recording is actually non-linear[2.5]. These non linearities in magnetic recording can be caused inter alia, by:

- (i) tape surface asperities and substrate inconsistencies. These alter the head to tape gap, which in turn alters the response of the channel. This has a greater effect upon high frequencies than it does on low frequencies;
- (ii) the positioning of the read/write head which effects the head to tape gap;
- (iii) tape saturation, this leads to non linear readback effects;

- (iv) the varying radius of the tape tracks in disc drives. This leads to a variation in linear density of up to 100% of the original value, in addition the thickness of the head to media air film can be altered by the changes in the disc head speed.

Such non-linearities necessarily lead to a more complex model of the recording process. This is described by Lin and Wood [2.6].

The second major difference between recording and communications channels is in the sources of error. In the latter this is electronic noise, while in the recording channel most errors are introduced on the recording media.

2.2:2 Timing Windows

In modulation codes the presence of a zero-crossing within a timing window is represented by a one, while a timing window without a zero-crossing is represented as a zero. A recording error will occur when either a zero-crossing is not present in a timing window where it should be or if a zero-crossing is translated from one timing window to the next. To prevent the translation of bits into adjacent timing windows, these may be separated by guard bands. The size of the timing window is thereby reduced.

2.2:3 Bitshift Induced By Non-Clockloss Mechanisms

The position of the peak of a transition will be shifted from the centre of the timing window by Noise and Inter Symbol Interference. The amount by which the peak is shifted can be found from the equation given by Schouhammer Immink[2.7]

$$r(t) = \sum_i x_i g(t - iT_b) + N_w(t) \quad (2.1)$$

where :

$r(t)$ is the received signal;

i is the number of interfering symbols;

T_b is the fundamental distance of a digital tape recording, which is defined to be half the reciprocal of the linear density;

$x_i \in \{-1,1\}$ are two valued quantities which are produced every T_b seconds;

$g(t)$ is impulse response of the channel;

$N_w(t)$ is additive white Gaussian noise.

The translation of the replay pulse peak is known as bitshift and can be caused by most error mechanisms. To generate an error a replay pulse peak is translated out of the timing window. The effect of bitshift can be illustrated below in Figure 2.3.

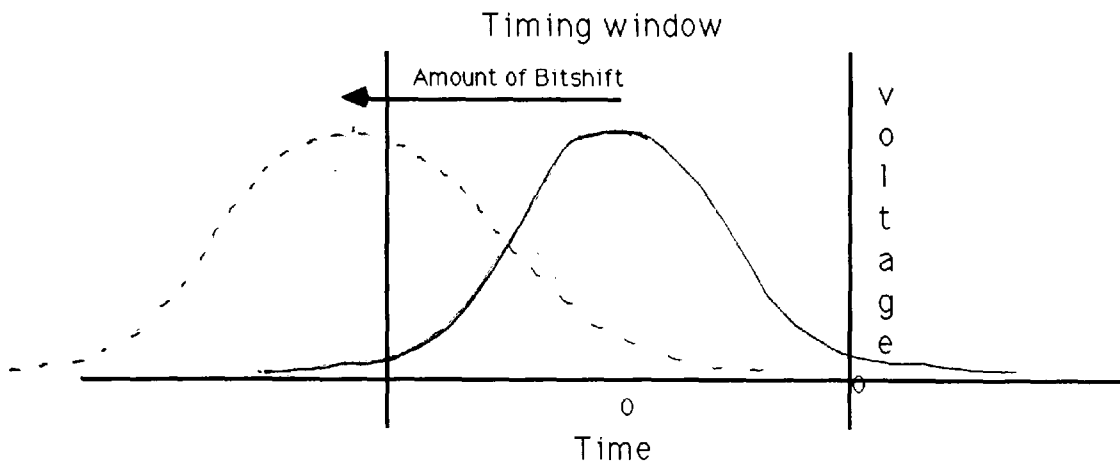


Figure 2.3 Timing Window showing Bitshift

In the figure the *bold* line represents the correct position of the replay pulse in the timing window, while the *hash* line represents the replay pulse after the bitshift due to noise and ISI.

2.3 The Recording Process

2.3:1 Pulse Shape

The shape of the recorded pulse is very complex. It consists of a longitudinal path and a lateral spread across the recording media[2.8]. This can be seen in the diagram below where the longitudinal direction is marked as M_x and the perpendicular direction as M_y . However in this work only the longitudinal direction of recording will be discussed.

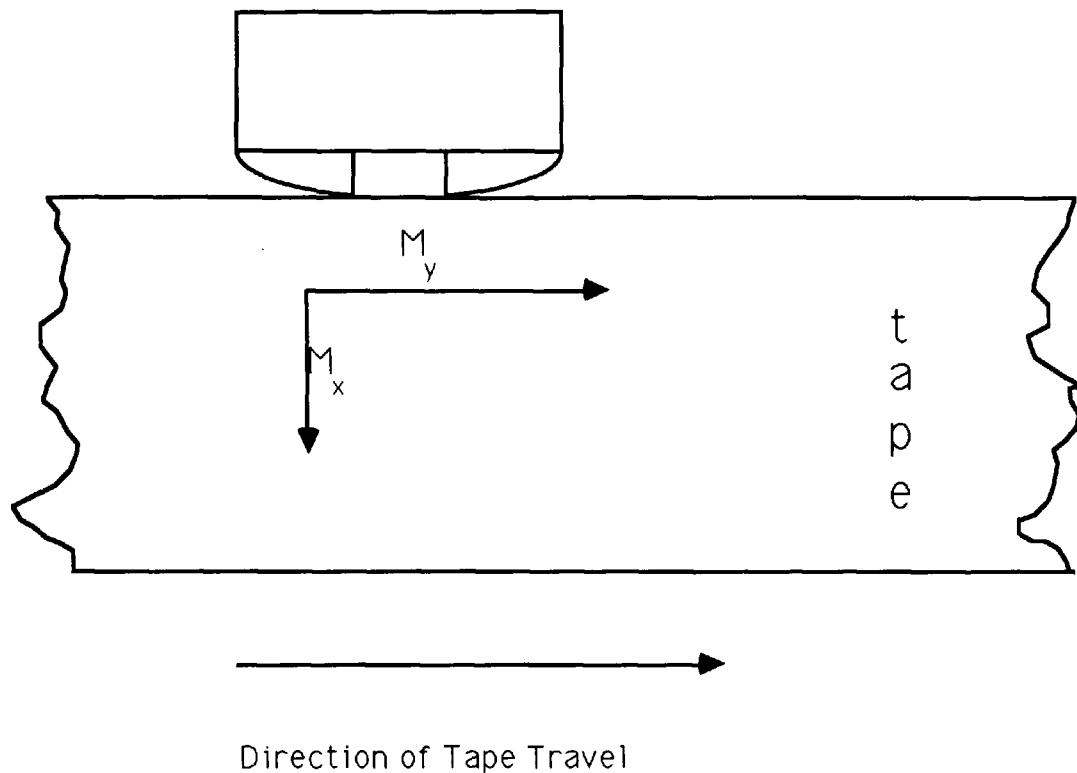


Figure 2.4 Perpendicular and Longitudinal recording

In common with all mathematical modelling several assumptions have been made. The most significant are[2.9]:

- i) uniform media thickness;
- ii) infinite Head permeability;
- iii) infinite track width;
- iv) magnetisation is purely longitudinal and of constant magnitude across the medium.

The implications of these assumptions may be assessed by comparison with practical measurements. There are several models for the isolated replay pulse produced by the longitudinal recording component. These are listed below in

order of increasing accuracy;

Gaussian [2.10]

Lorenzian [2.11]

Eldridge[2.12] and Speliotis [2.13]

Middleton and Wisely [2.14]

Gaussian:

Early work in this field assumed that the pulse has a Gaussian shape[2.10] :

$$e(t) = \exp \left(-\left\{ \frac{t}{\sigma} \right\}^2 \right) \quad (2.2)$$

Where;

t is time, relative to the pulse peak;

σ is the half pulse width, defined at the (1/e) point.

This expression for the replay pulse is purely longitudinal with no allowance for perpendicular recording. The lack of lateral spread assures symmetry, while the true replay pulse is asymmetric.

Lorenzian :

Mathematically this is represented by [2.11];

$$e(t) = \frac{V_0}{(1 + (t/t_{50})^2)} \quad (2.3)$$

Where;

V_0 is the peak replay voltage;

t is time relative to the pulse peak;

t_{50} is the half pulse width.

The Lorentzian expression assumes an arc-tangential recording current transition which is an improvement upon the transition assumed by the Gaussian expression given above in Equation 2.2.

Eldrige and Speliotis:

Eldrige[2.12] and Speliotis[2.13] derived a pulse shape:

$$e(x) = K \ln \left[\frac{(x^2 + (d+D)^2)}{(x^2 + d^2)} \right] \quad (2.4)$$

Where ;

t is time, relative to the pulse peak;

v is the head to media replay speed;

x is vt ;

K is a constant of proportionality;

d is the head to media separation;

D is the lesser of the medium thickness and depth of recording.

The expression in equation (2.4) above again gives a symmetrical replay due to the assumption that $M_y=0$. It is more accurate than the equations (2.2) and (2.3) as tape depth, head to tape separation and speed are all considered here.

Middleton and Wisely[2.14] modelled the pulse as:

$$e(x) = \frac{KM_r}{1 + \alpha} \ln \left[\frac{((a_t + d) + D(1 + \alpha))^2 + x^2}{(a_t + d)^2 + x^2} \right] \quad (2.5)$$

Where:

t is time, relative to the pulse peak;

v is the head to media replay speed;

x is vt ;

K is a constant of proportionality;

d is the head to media separation;

D is the lesser of the medium thickness and depth of recording ;

a_t is the recorded transition width at the top surface of the media;

M_r is the remanance of the recording;

α the rate of change of transition width with depth into the media

The expression in equation (2.5) above again gives a symmetrical replay due to the assumption that $M_y=0$. It is however, a more accurate representation of the recording pulse than those given by the equations (2.2), (2.3) and (2.4). This accuracy is due to the inclusion of recorded transition width at the top surface of the media. The recording remanance and the rate of change of transition width are all incorporated.

By contrast for a detailed description of both the longitudinal and the perpendicular replay pulses it is necessary to use the expression given by Middleton, Miles and Noyau[2.15]. This model gives an accurate description of the pulse accounting for the possibilities of a lack of symmetry and variations in the

pulse across the media. To get to the level of accuracy of the Middleton, Miles and Noyau model several complex measurements were taken.

However, since the present work is theoretical a simplified version of the replay pulse is given by the Lorentzian pulse. This disregards any variation in the lateral spread of the pulse. Although this is less accurate than other models, it is used due to its lack of complexity, lack of dependence upon accurate measurements and the use of pulse symmetry in the calculation of the ISI probability distribution.

2.3:2 Channel Errors [2.16]

As was stated in section 2.2:2 the majority of errors are generated within the recording channel. The form of error reflects the mode of operation of the channel. The theoretical models commonly fall into one of three main types either:

- i) the Binary Symmetric Channel (BSC). In this model a "1" is changed to a "0" and "0" to a "1" with equal probability;
- ii) the Asymmetric Channel (ASC). In this model a "1" represents a replay pulse and a "0" no replay pulse. Errors are caused by a "1" being changed into a "0" while a "0" is never changed into a "1";
- iii) the Shift Channel (SC) In this model a recording "1" can be translated into adjacent timing windows in either direction. So a segment of data that reads as 010 will be

transformed by in a shift channel to either 100 or 001. This type of error is common in high density recording.

A realistic model of the channel would be a combination of these three idealised modes. However an asymmetric channel ($1 \rightarrow 0$) is used in this work as all shift errors are assumed to be within the timing window guard band.

Howell[2.16] found that the distribution of errors for the IBM 3380 disk file were as follows:

- (i) the majority of errors (85%-97% of total errors) were SC errors (though not necessarily evenly distributed left-shift or right-shift);
- (ii) some (14%-3% of total) errors were ($1 \rightarrow 0$)'s ;
- (ii) practically no errors were ($0 \rightarrow 1$)'s.

Guard bands are introduced to the edges of the timing window to compensate for a SC error. If a bit is moved into these guard bands then it is treated as an ASC, error by the output. This is illustrated in Figure 2.5 below;

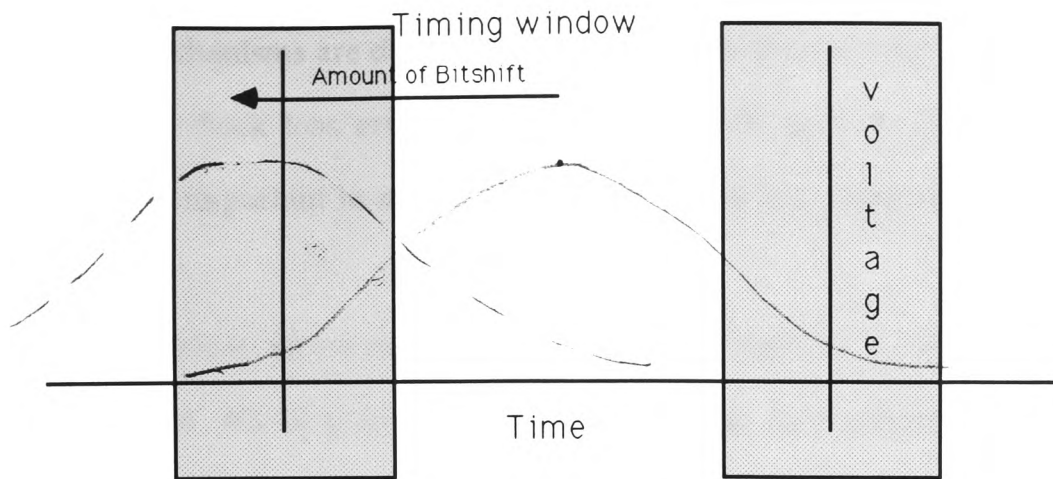


Figure 2.5 Timing window Showing the effect of guard bands on a S.C.

The size of the guard bands can be optimised relative to the timing windows to minimise the resultant error rate due to both ASC and SC errors. The lower the error rate then the smaller the guard bands need to be.

2.3.3 The System Clock

The system clock enables the detector to position each of the zero crossings on the recording track. Hence if the clocking is lost the data becomes unreliable and may produce bit-slip. Bit-slip leads to catastrophic errors as it is no longer possible to match the replay pulses to the input transitions. Errors occur because the position of the replay pulses do not correspond to the original data.

The effect of bit-slip in recording is therefore devastating, as the resultant disparity between the data-bit count and the bit stream of even just one bit will corrupt every word in the block. This as a read back error is as bad as a massive dropout covering the entire block removing 80% of the signal[2.17].

Error Mechanisms are divided into two different groups; those that lead to a loss of clocking, clock loss errors and those which do not, non-clockloss errors. It is therefore important to distinguish between these two types of error mechanisms.

Non-clockloss errors are due to mechanisms such as noise, ISI, crosstalk and overwrite. All of these error mechanisms can be represented by independent Probability Density Functions (PDF) and hence these can be convolved. These errors will occur in digital recording as Bitshift errors.

The most important errors causing a loss of clocking are jitter and drop-outs.

2.3.4 Clockloss and Non-Clockloss Errors

Clockloss as stated above is a primary cause of system error. It results in an inability to position the pulses from the replay head relative to the input signal to the tape recorder. Those error mechanisms which do not cause clockloss, have a probability density function which is broadly independent pulse to pulse. While the error mechanisms which cause clockloss tend to generate errors which are correlated over consecutive pulses. It is important to note here that clockloss error mechanisms are media dependent.

Clockloss is caused either by;

- i) a loss of a large number of consecutive replay pulses from the replay head of the tape recorder. This is called a drop-out;
- ii) the translation of a large number of consecutive replay pulses beyond the fundamental distance. This is called jitter.

Clockloss is generally not caused by bitshift errors, as the effect of bitshift is assumed to be independent from pulse to pulse. As clockloss is a major source of readout errors, it is essential to be able to recover clocking after a burst error. Clock recovery is an important factor in the choice of the runlength constraints of the modulation code. For good clock recovery a modulation code must have both the maximum runlength and the ratio of maximum to minimum runlengths to be as small as possible. Furthermore the higher the probability of a transition the better the resistance to clockloss error mechanisms.

2.4 Non Clockloss Error Mechanisms

2.4.1 Noise In Particulate Thick Film Recording [2.1,2.17]

Additive, non-white, Gaussian noise from the tape is dominant[2.1]; though additive, white, Gaussian electronic noise, which is associated with the preamplifier[2.8] is also present.

Noise is defined in terms the signal to noise ratio, (SNR). The signal for a thick film particulate media derives from the mean magnetisation of the particles. The noise arises due to deviations from the mean magnetisation. Tape noise is assumed to be stationary and additive, although it is dependent upon the signal. However, on good tape at low densities the noise increase with signal strength is slight[2.1].

Tape noise is defined [2.1] using the Power Density Spectra (PDS) for signal and

noise. To derive these spectra it is first necessary to define the Auto Correlation Function, ACF, of the pole strengths of the particles of an erased particulate tape. The magnetic pole strengths can be regarded as a random variable. Particles are contained within a lamina. For an example of the particulate structure of the tape see Figure 2.2. The ACF of the magnetic pole strengths is defined by;

$$ACF(x') = n \omega \delta y \int_{-\infty}^{\infty} p(x) p(x-x') dx \quad (2.6)$$

Where;

$x-x'$ is the relative distance along the tape;

ω is the width of lamina;

δy is the thickness of the lamina;

n is the density of particles;

$p(x)$ is the strength of the magnetic pole at point x .

The Noise Power Spectrum (NPS), $W_t(f)$, of the lamina pole is then given by the Fourier cosine transform of the ACF. To obtain the output voltage Noise Power Spectrum, $W_o(f)$, the Noise Power Spectrum is multiplied by the reproduction head power transfer function and integrated over the tape thickness.

$$W_o(f) = \int_0^a W_t(f) \cdot |H(f)|^2 dy \quad (2.7)$$

Where;

$W_o(f)$ is the output Noise Power Spectra

$W_t(f)$ is the Noise Power Spectra of the Lamina Pole

$|H(f)|^2$ is the head power transfer function.

a is the tape thickness

This gives the result below first derived by Daniel and Stein, [2.19],

$$E_n^2(k) = 4\pi \mu^2 n \omega V^2 |k| (1 - e^{-2|k|d}) e^{-2|k|a} \quad (2.8)$$

Where:

$E_n^2(k)$ is the Noise Power Spectrum;

ω is the width of lamina;

n is the density of particles;

μ is the dipole moment of the particles;

V is the replay voltage;

d is the head to tape gap;

k is a constant of proportionality;

a is the recording width at the top of the media.

If this is then integrated it will yield an upper bound for the noise power[2.20]. To compute the output SNR it is first necessary to find the output signal power spectrum. This is given by the following integral:

$$E_S(k) = 1/2 \int_{-\infty}^{\infty} [4\pi \mu n \omega f V (1 - e^{-kd}) e^{-kn}]^2 dk \quad (2.9)$$

Where:

ω , width of lamina;

n , density of particles;

μ is the dipole moment of the particles;

V is the replay voltage;

a is the recording width at the top of the media;

d is the head to tape gap.

The output SNR can be obtained from the above with the following assumptions:

- i) the SPS and the NPS have the same dependence upon head to tape spacing;
- ii) the measured spectrum deviates from the expected spectra at high wavelengths;
- iii) the measured noise spectra has a lower slope than expected;

Thus far the SNR has been assumed to be independent of the linear density of the recording. However at high recording densities the channel error mechanisms change [2.21,2.22]. ISI becomes the dominant source of error and the dependence of the output SNR to density of recording cannot be ignored.

2.4.2 Inter Symbol Interference

Due to the finite channel bandwidth the replay pulse is broadened so that adjacent pulses overlap. This phenomenon is called Intersymbol Interference, ISI. The effect of this interference is a translation of the position of the replay pulse peak, bitshift. If the raw data is assumed to be random then the effect of ISI on bitshift can be modelled as an independent random variable.

Due to the higher pulse densities used in digital recording, ISI is a more significant source of error in data recording than it is in communication channels where its effects may be reduced by the use of a linear filter[2.24,2.7]. If either the fundamental distance or the system bandwidth are reduced, then the level of ISI will increase. It has been shown that ISI is the limiting factor in the reliability of optical recording [2.25].

The effect of ISI as a bitshift error mechanism is due to its convolution with additive noise. This was modelled by Katz and Campbell[2.23] and Loze Middleton and Ryley[2.26]. ISI is a discrete random variable with a finite sample space. This sample space is due to the finite number of possible interfering discrete replay pulse patterns, 2^n where n is the number of interfering replay pulses.

$$P(b) = f_{\text{isi}}(b) * f_{\text{noise}}(b) \quad (2.10)$$

where;

$P(b)$ is the convolved bitshift distribution for noise and ISI;

$f_{\text{isi}}(b)$ is the bitshift distribution due to ISI;

$f_{\text{noise}}(b)$ is the bitshift distribution due to noise.

The discrete ISI induced bitshift distribution is convolved with the noise induced bitshift distribution to give a distribution for the combined effect of ISI and noise induced bitshift. For low packing density and high SNR the bitshift PDF due to ISI is given by:

$$B(E) = \delta(E - T_c) \quad (2.11)$$

Where:

E is the bitshift;

$\delta(\)$ the Dirac delta function;

T_c is the centre of the timing window.

With low density and high SNR the probability of an ISI induced bitshift error is zero, as the zero crossings occur at the centre of the timing windows.

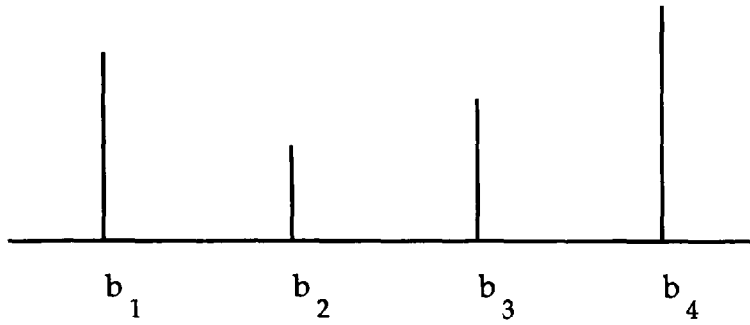


Figure 2.6 Dirac Deltas Representing Low Density ISI.

However when the packing density is increased, ISI is introduced. This leads to the following PDF for ISI induced bitshift :

$$P_S(n) = \sum_j P_j \delta(n-n_j) \quad (2.12)$$

Where;

P_j , the occurrence probability of the data bit at point j ;

n_j is the ISI induced bitshift from pulse j ;

$\delta(\)$ the Dirac delta function.

Assuming a Gaussian Probability Density Function for the noise induced bitshift, the ISI induced bitshift is now convolved with the noise induced bitshift. This results in a series of replicated noise induced bitshift PDF's centred on the dirac delta pulses for the ISI induced bitshift.

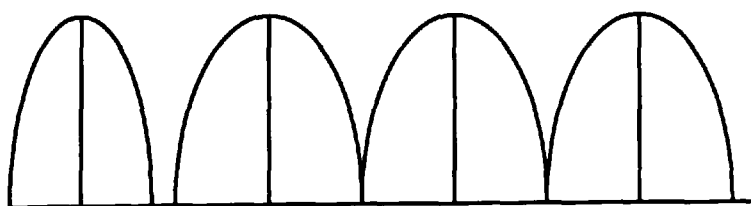


Figure 2.7 ISI Broadened by noise

The convolution of the PDF's for the discrete ISI induced bitshift with the continuous noise induced bitshift gives the following expression[2.23] :

$$P_k(t) = \frac{1}{2} \sum_i A_i (\exp[-(t-\tau_{1i})^2/2\tau_{2i}^2] + \exp[-(t+\tau_{1i})^2/2\tau_{2i}^2]) \quad (2.13)$$

Where:

$P_k(t)$ is the convolved PDF for Noise and ISI induced bitshift;

A_i is the fraction of pulse peaks in the i^{th} local data pattern ;

τ_{1i} is the bitshift induced by ISI ;

τ_{2i} is the rms value of noise induced bitshift;

t is time .

The model given above is limited as it only considers the effect of ISI from adjacent pulses. Convolving the PDF for ISI given above in (2.12) with the noise PDF, assuming that as the tape is of finite length only a finite number of symbols interfere, then the probability of error can be given as [2.26]:

$$P_e = \frac{1}{4M} \sum_j \frac{\text{erfc}(T_w + |n_j| + T_c)}{\sqrt{2} \sigma_j} + \frac{\text{erfc}(T_w + |n_j| - T_c)}{\sqrt{2} \sigma_j} + \frac{\text{erfc}(T_w - |n_j| + T_c)}{\sqrt{2} \sigma_j} + \frac{\text{erfc}(T_w - |n_j| - T_c)}{\sqrt{2} \sigma_j} \quad (2.14)$$

Where;

σ_j is the rms value of the noise induced bitshift acting on that pulse;

n_j is the ISI induced bitshift acting on that pulse;

erfc is the complimentary error function;

T_c centre of the timing window;

T_w width of half the timing window.

2.4.3 Crosstalk and Overwrite[2.27,2.28]

Crosstalk and Overwrite are similar to ISI[2.29] as both are caused by the interference of other pulses. It is often difficult to distinguish between crosstalk and ISI induced bitshifts. Overwrite is caused by poor DC-erasure of the media. This leads to interference from the unerased signals. Crosstalk in magnetic recording occurs only in multi-head recording media. There are two main sources of crosstalk in the recording system.

(i) The recording head This is due to leakage of the magnetic flux between the cores in the recording head. This can be displayed by DC-erasure of the signal from the two adjacent tracks. This leaves the central track unaffected by the erasure process. Any remaining crosstalk on this middle track is due solely to the data on the track. Recording head crosstalk is reduced by reducing the flux leakage in the recording head.

(ii) Play back crosstalk. This occurs at the read head and is caused by two separate mechanisms. To display the playback crosstalk the opposite procedure is followed; only the centre track is erased and the crosstalk comes from the two adjacent tracks which are not erased. To compensate for crosstalk in the medium the opposite signal to that of the crosstalk is added. This results in cancellation of the crosstalk.

It is difficult to assess the number of playback errors which are due to the effects of crosstalk. This leads to several complex expressions to describe the effects of crosstalk induced bitshift on a recorded channel[2.28].

2.5 Clockloss Error Mechanisms

2.5.1 Drop-outs

Drop-outs are reductions in the signal amplitude which occur as a result of the variation in the local properties of the tape and during the recording or playback processes. Low amplitude drop-outs cause a bitshift error which does not cause any loss of clocking. However high amplitude drop-outs may cause a near total loss of signal, leading to a loss of clocking. Drop-outs occur either due to tape defects or errors in positioning the tape heads over the media.

2.5.1(i) Tape Defect Drop-outs [2.30]

Tape defect drop-outs are a major factor in determining the reliability of magnetic-storage devices. Large drop-outs are usually flagged by the recording system. The areas of defective tape are then skipped for data recording. Those defective regions which escape the 'write skip' process are a major source of error especially for a channel with high SNR.

The percentage of drop-out depth is plotted against the log of its probability. However where the levels of drop-out are deeper than 80% their effects are hard to assess[2.30], because at this level the drop-outs will be masked by the amount of noise in the system. So Gene So Honu[2.30] extrapolated the levels of the lower drop-outs from the three higher received signals to give a better plot for the amount of residual signal in these lower cases.

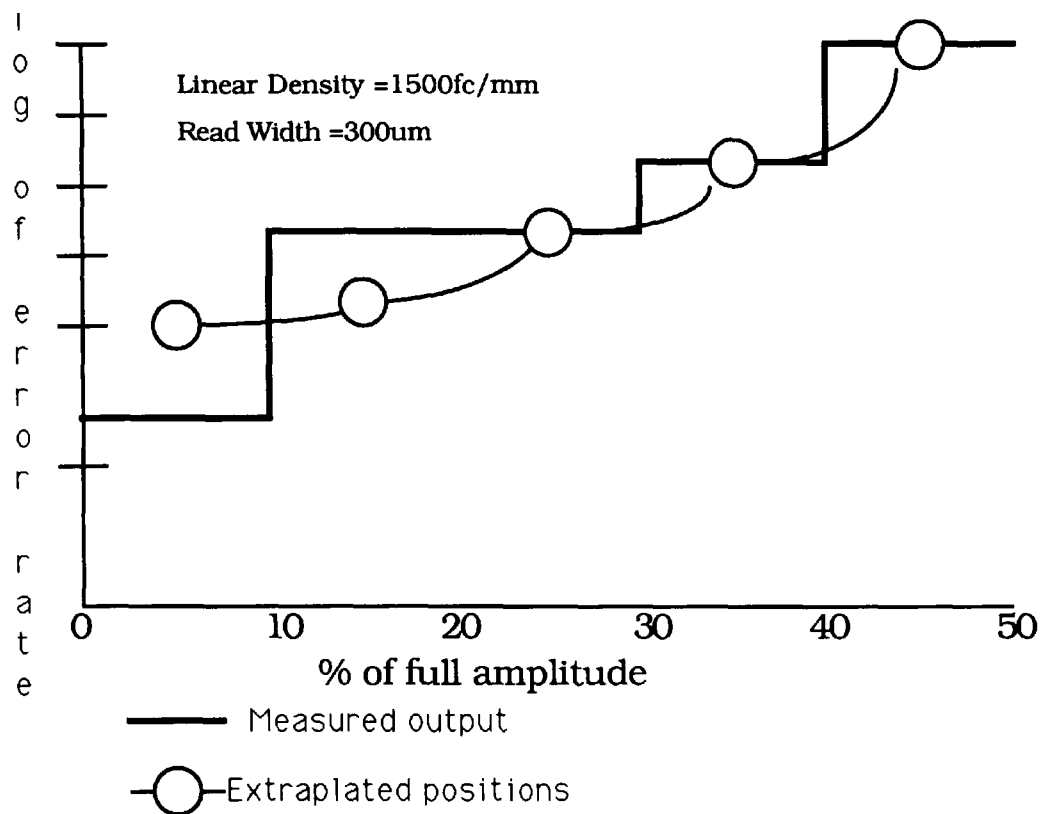


Figure 2.8 Log Error Rate against % of signal Loss [2.30]

For a single drop-out of diameter ϕ , on a single track of width h the drop-out will affect the measured playback level of the tape at that point by a factor of at most :

$$K = \frac{h}{(\phi - 1)} \quad \text{if } \phi \leq h \quad (2.15)$$

where:

ϕ drop-out diameter;

h track width.

This can occur only if the drop-out lies directly on the track and is not larger than the track width. Once Equation (2.15) for the observed effect of the drop-out has been derived, there remains to find an expression for the drop-out frequency. The analysis of Maediger et al [2.37] gave the following integral:

$$F = \int_{\phi_{\min}}^{\phi_{\max}} 2B(\phi) - B(\phi).h.(1-2.10^{D/\phi}) d\phi \quad (2.16)$$

Where:

$B(\phi)$ is the specific drop-out load per unit area for the track;

h is the tape width;

D is the permissible drop in SNR;

ϕ is the diameter of the drop-out.

2.5:1(ii) Drop-outs Caused by Head Errors [2.31,2.32]

The Errors caused by the head to tape separation were first studied by Baker [2.32] using a random sequence to check for errors . The head to tape distance was given by:

$$y(x)=x^2/2r \quad (2.17)$$

Where;

$y()$ is the head to tape spacing;

x is the distance from start of lift off to tangent point;

r is the head radius near the gap.

A Miller Modulation encoded channel which was subject to additive white Gaussian noise was assumed to have a bit error probability related to the Complimentary Error Function[2.32].

$$P_e = 1/2 \operatorname{erfc}(\operatorname{SNR}/\sqrt{2}) \quad (2.18)$$

An increase in the SNR was noted as the head to tape separation was increased. This increase can be expressed as a ratio of the two SNR's for differing head gaps.

$$\frac{\operatorname{SNR}}{\operatorname{SNR}_{y=0}} = \frac{1}{\left[y \int_0^a S(k) dk \right]} \triangleq \frac{f(L)}{y(x)} \quad (2.19)$$

where;

SNR is the output signal to noise ratio including the drop-out;

$\operatorname{SNR}_{y=0}$ the signal to noise ratio when head to tape gap is zero;

$S(k)$ is the system response;

k is the wave number, $(2\pi/\lambda)$;

$f(L)$ is the peak fraction of particles recorded upon;

L is the bit length;

$y(x)$ is the head to tape gap;

a is the cut off point for the separation induced loss.

Then discounting the ISI this gave the following equation for the bit error probability for $y > 1/k_m$.

$$P_e = 1/2 \operatorname{erfc}(\operatorname{SNR}_0 f(L)/[\sqrt{2} y(x)]) \quad (2.20)$$

where;

SNR_0 is the SNR when the head to tape gap is zero;

$y()$ is the head to tape gap ;

$f()$ is defined above;

L is the bit length.

In addition a system of all ones was used [2.31] to give the initial theoretical output voltage amplitude of the system with no head to tape separation as:

$$V_0 = K \frac{4M_0}{1+\alpha} [1 - \exp(-2\pi(a_1 + d)/\lambda)] \quad (2.21)$$

where;

M_0 is the maximum level of magnetisation;

K is a constant of proportionality;

a_1 is the transition width parameter at the top of the tape;

d is the head to tape separation on replay;

λ is the fundamental distance;

α is the linear variance of transition width with depth.

In addition the error rate was also altered suggesting that the higher values of d produce larger error probabilities. In practice it is likely that burst errors will occur as a result of lower values of d than are strictly necessary, caused by the increase in the gap height as discussed in the section above.[2.32]

Many experiments have been undertaken to examine the effect of drop-outs on a recording channel.

In one case[2.34] the top layers of the recording media are burnished to give the effect of media deterioration with time. The signal is recorded on to the unburnished media, and is then replayed prior to burnishing. The media are then burnished, by stripping away their upper layers. The two replay signals may then be compared to estimate the drop-out effects.

Other work[2.35] has shown that the error rate depends upon the recording signal current.

2.5.1(iii)Drop-out Effects [2.36]

Although two quite different methods by which drop-outs may occur have been discussed, their effects are the same. The ways in which the system performance of the tape is effected by a drop-out include:

- i) drop-outs tend to be isolated and widely separated events;
- ii) drop-outs are randomly distributed throughout the tape;
- iii) drop-out lengths usually obey a logarithmic law of decaying frequency;
- iv) drop-outs on edge tracks occur at a higher frequency ;
- v) drop-outs are unlikely to occur simultaneously on two tracks;
- vi) the maximum length and frequency of the drop-out are media dependent;
- vii) the dominant error characteristic from a drop-out is a burst error.

It is possible to plot the read back errors for drop-outs either as a function of probability of error or as the probability of drop-out versus length of the drop-out. This then gives rise to a method of calculating the effect of drop-out induced errors in a statistical fashion. The results were tabulated for various lengths of drop-outs.[2.33]

Figure 2.9a, plots the measured signal amplitude against the number of times which it occurs.

Figure 2.9b, plots the percentage of bits in error given the measured signal amplitude from above;

From the two diagrams[2.33] it is possible to gain an additional figure for the error probability due to drop-outs. This is achieved from the multiplication of the two sets of figures in the final column of each of the diagrams.

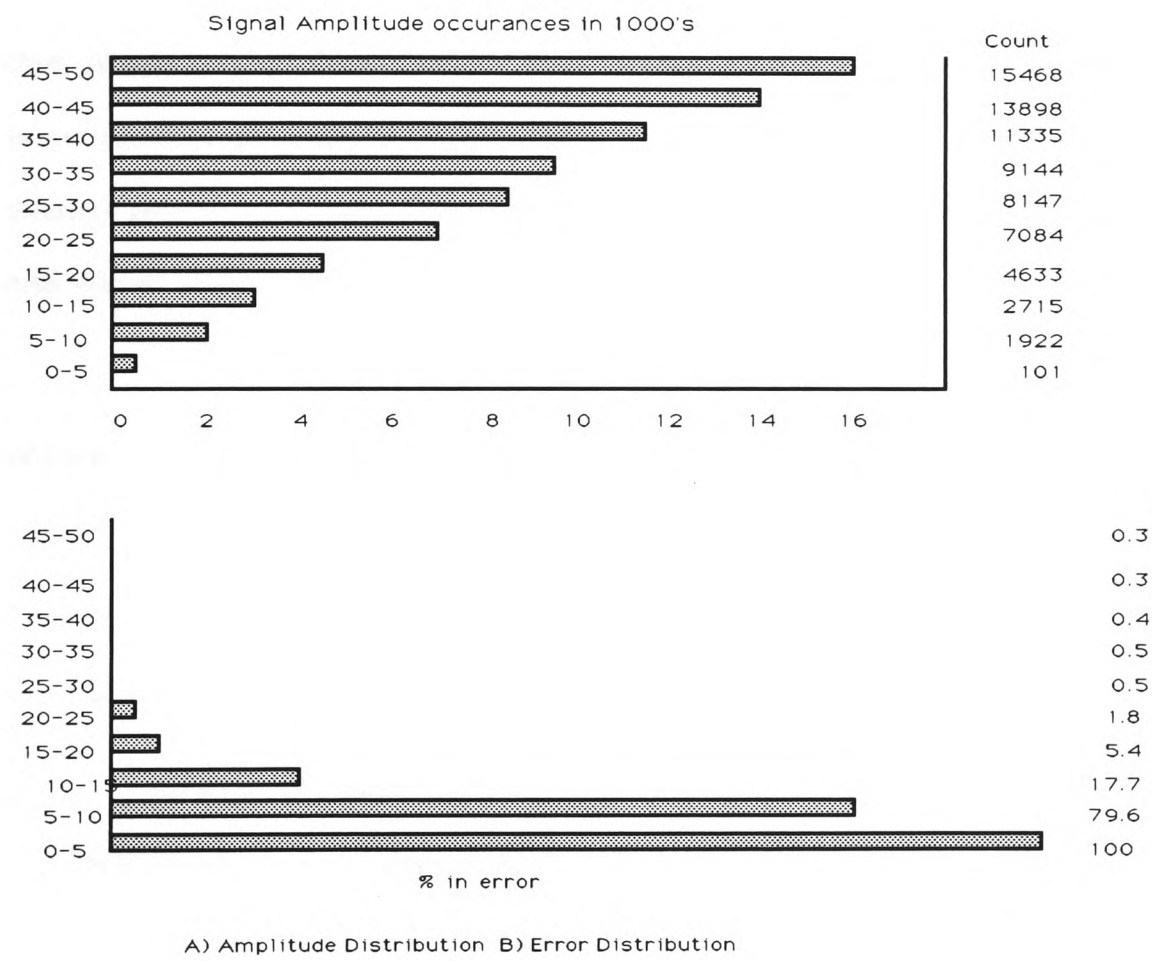


Figure 2.9.a) and Figure 2.9 b) Amplitude and error distribution for drop-outs[2.33]

2.5.2 Jitter

Considered as a clockloss error mechanism, jitter is strongly medium dependent. In magnetic data storage it is most prevalent in thin film media. As with drop-outs, jitter can result in either bitshift or clockloss errors[2.39]. Jitter in thin film media is concentrated in the region around the magnetic transitions[2.38]. This can be represented by assuming the transitions are all the same shape but are then subject to jitter uncertainty in their location. The additive noise in the system is assumed to be constant while the effect of jitter increases linearly with the average recording density. The amount of jitter in the system is defined by Madrid and Wood[2.40] to be:

$$N(f) = D |F(2\pi f)|^2 (2\pi f)^2 \tau_j^2 \quad (2.22)$$

Where;

D is the average transmission density;

$F(2\pi f)$ is the transfer function of an isolated replay pulse;

D average transition density;

τ_j^2 variance of Gaussian jitter;

$N(f)$ Power spectral density of noise from independent jitter.

Although primarily an error mechanism associated with thin film media, jitter is also present in thick film recording. To counteract its effect, the fundamental distance of recording must be greater than the jitter time uncertainty. Hence jitter is a limiting factor on the packing density of the recording.

The effects of jitter and noise can be illustrated by an eye pattern. This is used to assess the effect of modulation coding upon a channel. If the eye pattern remains open then a binary decision may be made about the signal. Noise closes the eye in a vertical direction and jitter in a horizontal direction[2.41]. Hence although noise and jitter are independent, the tolerance of a modulation coding system to jitter is inversely related to its tolerance to noise.

REFERENCES

- 2.1) J.Mallinson** "Maximum Signal to Noise Ratio of a Tape Recorder" IEEE Trans. on Magnetics Vol MAG-5 No.3 Sept 1969 pp 182-186
- 2.2) J.Watkinson** "The Art of Digital Audio" Section 6.3 pp 163-170 Focal Press London 1989
- 2.3) E.M. Deeley** "Integrating and differentiating Channels in Digital tape recording" Radio and Electronic Engineer Vol 56 1969 pp 169-173
- 2.4) C.D. Mee and E. Daniel (Editors)** "Magnetic recording' Volume 1 "Technology" Chapter 3 "Recording Media" E Koster and T.C. Arnoldussen p 186 McGraw Hill 1987
- 2.5) D.Palmer P.Ziperovich R.Wood and T. Howell** "Identification of Non-Linear Write Effects using Pseudo random Sequences" IEEE Transactions on Magnetics Vol MAG-23 No.5 Sept 1987 pp 2377-2379
- 2.6) Y.Lin and R.Wood** "An Estimation technique for Accurately modelling the Magnetic Recording Channel Including Non-Linearities" IEEE Transactions on Magnetics Vol MAG-25 No.5 Sept 1989 pp 4084-4086.
- 2.7) K.A.S Immink** " Coding Techniques for digital Recorders" Prentice Hall International 1991 Chapter 1 p 5
- 2.8) M.K. Loze** "Computer aided Modelling of Digital Magnetic Recording Systems" Ph.d. Thesis 1987 Manchester Polytechnic
- 2.9) Li Hui Hong** "A Computer Simulation Package for Longitudinal and Perpendicular Digital Recording Systems' M.Sc. Thesis 1991 Manchester University
- 2.10) W.W. Chu** " Computer Simulation of Waveform Distortions in Digital Magnetic Recording" IEEE Trans on Electronic Computers Vol EL-15 No.2 1966 pp 328-336
- 2.11) A.J. Koster and D.E.Speliotis** "Predicting Magnetic Recording Performance by using Single Pulse Superposition" IEEE Trans Vol Mag-7 Sept 1971. p 544
- 2.12) F.Eldrige** "Magnetic Recording and Reproduction of Pulses" IRE Trans on Audio Vol.3 March 1960 pp 42-57
- 2.13) D.E.Speliotis and J.R.Morrison** "A Theoretical Analysis of Saturation Magnetic Recording" IBM Journal of Research and Development Vol 10 May 1966 pp
- 2.14) B.K. Middleton and P.L.Wisely** "Pulse Superposition and High Density Recording" IEEE Trans on Magnetics Vol MAG-14 No.5 Sept 1978 pp 1043-1050
- 2.15) B.K.Middleton, J.J. Miles and R.H.Noyau** "Magnetic Properties of Materials, Transition Shapes and Digital Recording Properties" IEEE Trans on

- 2.16) **T.D. Howell** " Analysis of Correctable Errors in the IBM 3380 disk file" IBM J.Res. Dev. Vol 28 1984 pp 206-211.
- 2.17) **F.A.Griffiths** "A Digital Audio Recording System" Presented at 65th Audio Engineering Society Convention (London 1980)
- 2.18) **B.K.Middleton and T.Brown** "Recording Properties and Digital Recording System Performance" The Radio and Electronic Engineer Vol.50 No.9 pp 467-473
- 2.19) **I.Stein** "Analysis of tape noise' IRE 1962 Intern.Conv. Rec.Vol 10 part 7 pp 42-65
- 2.20) **C.D.Mee** " The Physics of Magnetic Recording"Amsterdam: North Holland Publishing 1964 p131
- 2.21) **J.Mallinson** "On Extremely High Density Recording" IEEE Trans. on Magnetics Vol.MAG-10 No.2 June 1974 pp 368-373
- 2.22) **B.I.Finklestein and E.R.Christensen**"Signal to Noise Ratio Models for high Density Recording "IEEE Trans on Magnetics Vol MAG-22 No.5 Sept 1984 pp 898-900
- 2.23) **E.Katz and T.Campbell** "Effect of Bitshift Distribution on Error Rate in Magnetic Recording" IEEE Transactions on Magnetics Vol MAG-13 No.3 May 1979 pp 1050-1053
- 2.24) **H. Nyquist** "Certain Topics in Telegraph Transmision Theory" Trans AIEE Feb 1928 vol 47 pp 617-644
- 2.25) **C.M.J. van Uijen** 'Measured Performance of Modulation Strategies for Digital Optical Recording Media' Topical Meeting on Optical Data Storage,Technical Digest Series 1987 Vol 10 pp 102-105
- 2.26) **M.K. Loze B.K. Middleton and A.Ryley**"Simulation of Digital Magnetic Recording Systems" IERE Conference on Video Audio and Data Recordings Brighton 1988 pp 185-187
- 2.27) **T.Tanaka** "Some Considerations of Crosstalk in Multi-head Digital Recording" IEEE Trans. on Magnetcs Vol.MAG-20 No.1 Jan 1984 pp 160-165
- 2.28) **W.J.van Gestel** "The Influence of Crosstalk on the Bit error Rate in Magnetic Recording" 7th IERE Conference on Video Audio and Data Recording York 1988 pp 35-43
- 2.29) **N.C.Beaulieu** "The Evaluation of the Error Probabilites for Intersymbol and Cochannel Interference" IEEE Trans. on Communications Vol39 No.12 Dec 1991 pp1740-1749
- 2.30) **G. Ho Sonu** "Study of drop-out Rate for Maximum Density Recording" IEEE Trans on Magnetics MAG-21 No.5 Sept 1985 pp 1392-1394

- 2.31) **B.K. Middleton and T. Jack-Kee** "Performance of digital Magnetic Recording Channels subject to Noise and Drop-outs" The Radio and Electronic Engineer Vol.53 No.11/12 1983 pp 393-402
- 2.32) **B.Baker** "Drop-out Model For Digital Tape Recorder" IEEE Trans on Magnetism MAG-13 No.5 pp 1196-98
- 2.33) **D.B. Seymour**"Characterization of Drop-outs with Digital Read-Back Errors for Digital Magnetic-Tape Recorders" IEEE Transactions on Magnetism Vol Mag-22 No.1 January 1986 pp 44-46
- 2.34) **A.S. Hoagland W.F.Oehme and F.E.Talke** "Experimental study of Drop-out Signal Loss"IEEE Trans on Magnetism Vol MAG-16 N0.5 Sept 1980 pp 982-984
- 2.35) **B.K. Middleton and T. Jack-Kee**"Drop-outs and their Effects on Error Rates in Digital Magnetic Tape Recording System" Conference on Video Audio and Data Recording IERE Conference. Publication. No. 54 pp 43-49
- 2.36) **L.A. Meeks** "Drop-out Characteristics of Instrument Tape and Approaches to Error Correction" Electronic Engineering March 1982 pp 117-127
- 2.37) **C.Maediger H.Voight H. Voels and K. Willaschek** "Analysis of Signals Statistics and Drop-out Behaviour of Magnetic Tapes" IEEE Transactions on Magnetism Vol MAG-20 No.5 Sept1984 pp 765-767
- 2.38) **R.Wood** "Jitter Versus Additive Noise" IEEE Trans on Magnetism Vol MAG-23 No.5 Sept 1987 pp 2683-2685
- 2.39) **L. Nunnally R.D. Harper and M.Burleson** "Time Domain Noise Induced Jitter: Theory and Precise Measurement" IEEE Trans on Magnetism Vol MAG-23 No.5 Sept 1987 pp 2383-2385
- 2.40) **M.Madrid and R.Wood** "Transition Noise in Thin Film Media" IEEE Trans on Magnetism Vol MAG-22 No.5 Sept 1986 pp 892-894
- 2.41) **J.Watkinson** "The Art of Digital Audio" Section 6.4 pp 170-174 Focal Press London 1989

Chapter 3

Recording Media and Devices

3.1 Recording Principles

3.1.1 Introduction

In this chapter the different types of data recording media are reviewed. These fall into three main categories:

- (i) thick film magnetic recording;
- (ii) thin film magnetic recording;
- (iii) optical recording.

The two types of magnetic recording media can be erased and then overwritten with alternative data. However optical recording only possesses the ability to be written to once though it may be read many times (WORM). Optical discs cannot be erased. A recording on an optical disc is therefore permanent.

Modern developments in magnetic and optical recording technology have lead to high integrity media. These have high recording densities, fast accessing of data and low output error rate. To obtain such a recording performance, complex error correction and modulation coding techniques are used.

The work described in this thesis has principally been concerned with fixed head thick film recording. However the new high integrity media are either thin film, Winchester disc, rotary head, Rotary Digital Audio Tape, (R-DAT) or Optical, Compact Disc Read Only Memory, (CD-ROM). The last two have been developed for use as consumer audio products so allowing the economies of scale of the domestic market.

3.1.2 Magnetic Recording Media[3.1]

Historically the first magnetic recording systems to gain widespread use employed thick film magnetic storage media, using a fixed position recording and reading head.

The system uses a lamina containing a series of ferrous particles in which electric pulses induce a change in the magnetic field so allowing the data to be recorded. This was shown in Section 2.2.2, figure 2.1. The chemical constituents of the ferrous material may change from media to media. Different materials have different properties which are dependent upon use.

The most common type of ferrous material used is gamma iron oxide, $\gamma\text{-Fe}_2\text{O}_3$ [3.1]. This is diluted by an organic binder so that about 20-45 % of the tape by volume is ferrous material. This reduces the magnetic saturation from the original figure of 350 kA/m^2 for undiluted ferrous material. In tape and most other recording media the magnetic particles are carried on a polymer substrate, where the substrate and the magnetic material can be optimised independently.

The properties of various types of particulate tape used for different applications are shown in Table 3.1 [3.1]:

Application	Material	$M(\infty)$ kA/m	H kA/m	∂h	N
Reel to Reel	$\gamma\text{-Fe}_2\text{O}_3$	100-120	23-26	.3-.35	0.3
Instrument Tape	$\gamma\text{-Fe}_2\text{O}_3$	90	27	.35	0.6
	$\gamma\text{-Fe}_2\text{O}_3 + \text{Co}$	105	56	.50	0.8
Floppy disk	$\gamma\text{-Fe}_2\text{O}_3$	56	27	.34	0.3
	$\gamma\text{-Fe}_2\text{O}_3 + \text{Co}$	60	50	.34	0.5
Computer tape	$\gamma\text{-Fe}_2\text{O}_3$	87	23	.35	0.16
	Cr_2O_3	120	40	.29	1.4

Table 3.1 Tape Properties for Different Mechanisms[3.1]

Metal film media were introduced to allow for faster accessing of stored information. Thin film media were originally stored on spinning drums, allowing the Read-Write head to be quickly positioned on any part of the drum. The thin film system presently allows a higher areal storage density than particulate tape media. However the recent advances in recording densities obtained by R-DAT suggest densities of the same order as thin film will be possible.

3.1.3 Fixed Head Tape Recording

The first of particulate recording media used for instrument recording was developed from analogue audio tape recorders. These had fixed heads and employed either the reel-to-reel system or the compact cassette.

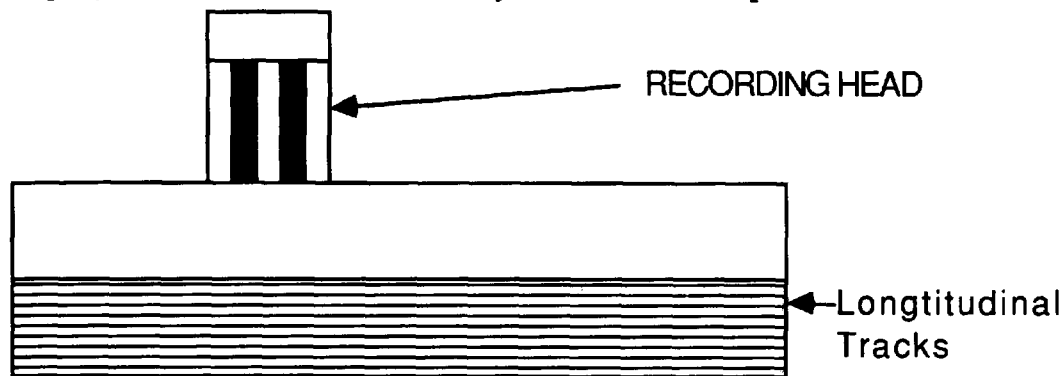


Figure 3.1 Diagram of Fixed Head Recorder showing Tracks

Fixed head recorders have been shown to have a number of specific advantages and disadvantage compared to a rotary head system.

Disadvantages of Fixed Head Recorders

- i) The storing of digital information without a clear format structure presents problems. These are in part due to the lack of a scheme for error correction. A system of format blocks coupled with redundancy was developed to reduce system errors [3.2];
- ii) In fixed head recording fast retrieval of stored data cannot be achieved. As data can only be accessed from the tape in a serial fashion. All other types of recording media have a random access facility for file retrieval[3.2];

- iii) Fixed head recorders have lower areal storage densities than rotary head recorders so using more tape to store the same amount of data. For example a standard 24-track quarter inch tape recorder with a tape speed of 16 ips will require a data rate of only 24 Mbs⁻¹. This provides an areal data density of less than 100 Mbpi²[3.3] .

Advantages of Fixed Head Recorders [3.2]

- i) Fixed head recorders possess larger reliability than rotary head recorders such as Video and R-DAT;
- ii) Fixed head recorders have a capstan controlled tape feed. This allows various recording and playback speeds, autolocation and the ability to synchronise several recording machines;
- iii) Fixed head machines permit tape cut edits as the tracks run in the direction of the tape travel. In Helical scan recorders these edits may cause severe errors;
- iv) Fixed head tape is inherently simple compared with the other methods of digital recording. This simplicity is illustrated by the block diagram shown in Figure 3.2;

- v) Due to their simplicity fixed head recorders are generally more robust than those with rotary heads. However in recent years a robust rotary head machine has been produced [3.3].

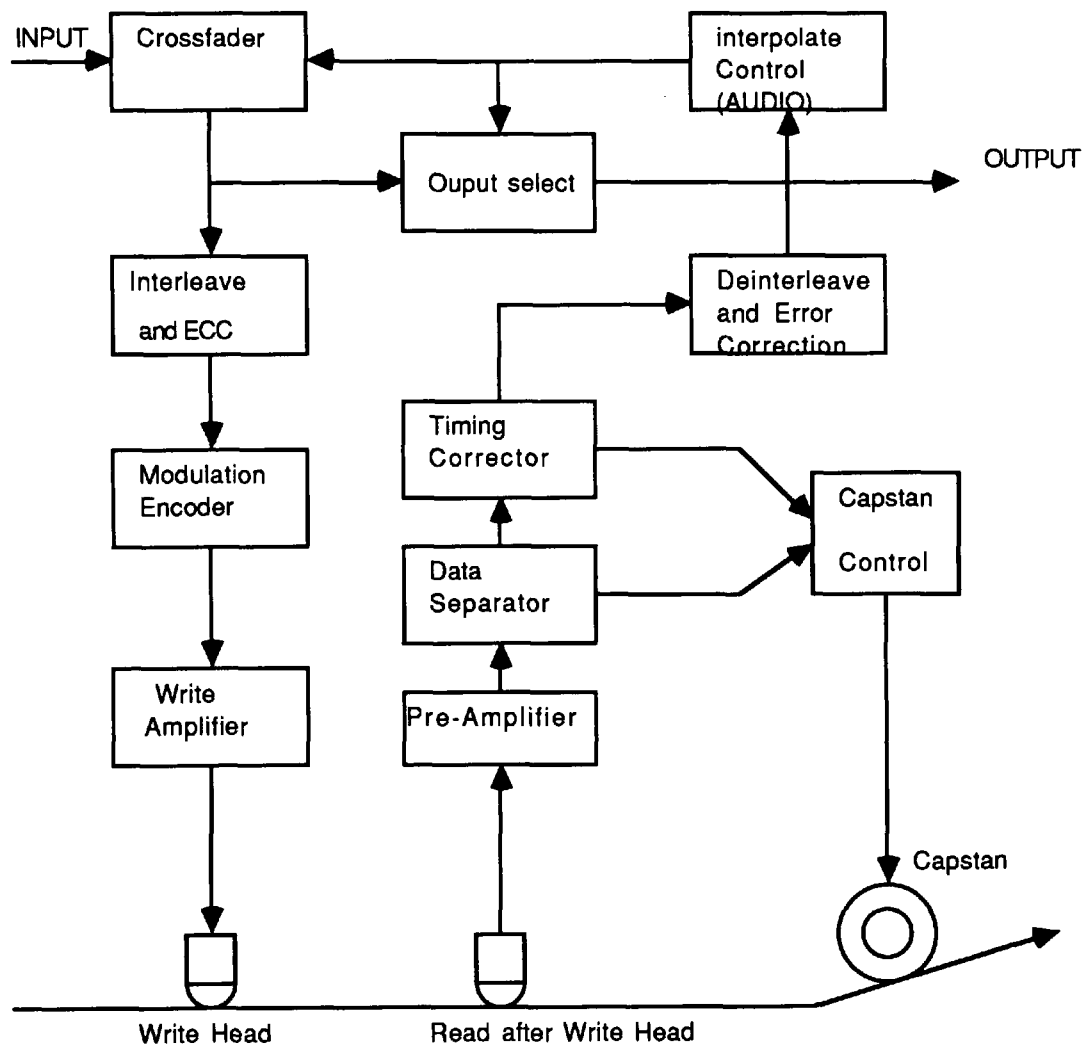


Figure 3.2 Schematic Diagram of the Fixed Head Recording Process

3.2 ROTARY HEAD TAPE RECORDING

3.2:1 Principle of Rotary Head Recording

There are two distinct categories of rotary head mechanisms:

(i) the helical Scan recorder. Here the heads are positioned on a drum which rotates in a plane parallel to the tape travel and records diagonal fields. These are recorded as the tape is wrapped around the recording head. This is shown in Figure 3.3 A;

ii) the transverse Scan recorder. The heads are fixed on a disc which rotates in a plane perpendicular to that of the tape travel. These heads record short tracks across the tape surface. This is shown in Figure 3.3B.

These both record data across the media. This is in contrast to the stationary head recorder which records the tracks parallel to the direction of tape travel, along the length of the recording medium. The different tape tracks are shown in Figure 3.4 below

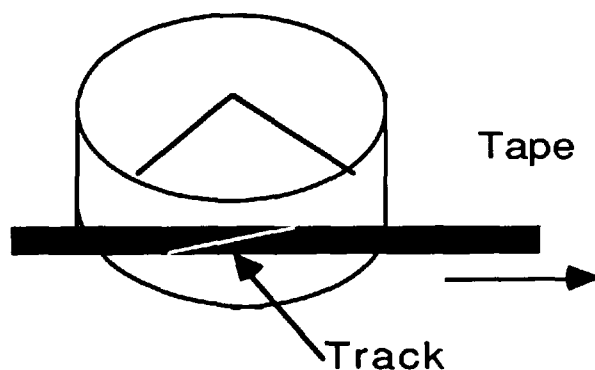


Figure 3.3A Helical Scan Recorder[3.2]

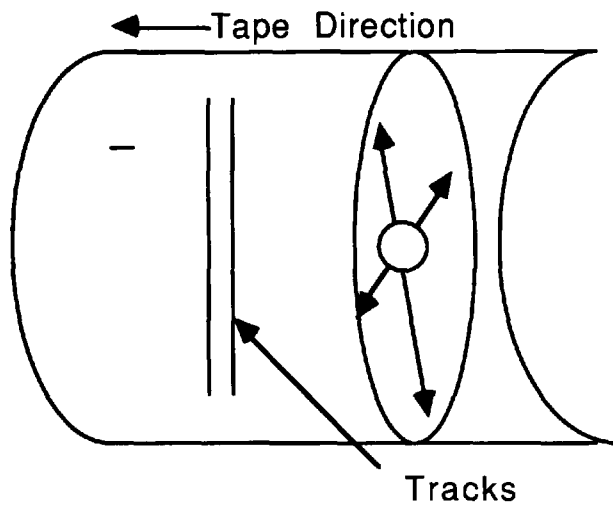
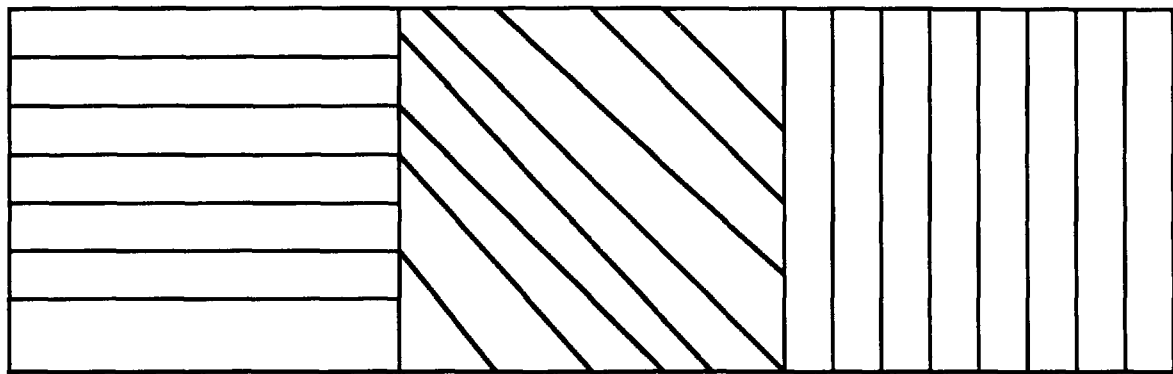


Figure 3.3 B Transverse Scan Recorder[3.3]



Fixed Head Tracks

Helical Scan Head Tracks

Transverse Scan Head Scan

Figure 3.4 Track Layout is Dependent upon the choice Recording Head

Instrument helical scan recorders have been available for many years. However these are expensive to produce compared to a commercial rotary head recorder. Rotary head recorders are high integrity devices due to error correction and modulation coding techniques. Time compression is used to store increased amounts of data on the readable part of the tape track. This leads to a high capacity low cost recording system with excellent data rates. For example the areal density for R-DAT is 114 MBPI².

3.2:2 Digital Video Recorders

The advantages of video Cassettes are that they produce a low cost, high capacity backup data storage. Their storage capacity is 3.4 Gigabytes with a transfer rate of 3.75 Mbs^{-1} . In addition to the high capacity and transfer rate the density of the recording is high at $4450 \text{ Gigabyte/M}^2$ [3.4].

Digital video recorders avoid errors by using a Read After Write (RAW) technique. This process does however lead to a reduction in the amount of recording heads on the helical drum. In addition there is a reduction in the data density since only every second track is now written on. Crosstalk which is reduced by recording on every second track is further reduced by the use of an azimuth angle on the recording heads of ± 6 degrees. This error avoidance reduces the error rate from 10^{-3} to 10^{-9} [3.4].

Certain video cassette recorders offer a choice of recording speed, 33.35 mm/sec , normal play, 16.67 mm/sec , long play and 11.12 mm/sec , super long play. This gives a choice of areal density for the recording. For the faster speeds a method of data recovery is necessary. However at very high recording densities it is not possible to accurately recover the data due to the severe level of crosstalk.[3.5]

Further work by the Japanese[3.6] has led to increasingly successful methods of storing digital data on video cassette. The Pulse Code Modulation, (PCM) adaptor system uses standard video tape to record either digital or analogue signals.

The principle use of digital video machines is to master optical discs (CD) for audio use. Therefore a system using digital video recorders has applications in the mastering of optical discs for prerecorded data, (CD-ROMs). Although video recorders have been very successful applied to the digital audio and data storage fields. There are several drawbacks in using them as the primary recording medium [3.8]

3.2:3 R-DAT Rotary Digital Audio Tape

The rotary digital tape recorder was developed as a cheap magnetic alternative to the digital compact (laser/optical) disc. The replay quality of the two systems are comparable[3.8].

R-DAT was originally developed for audio use in the first half of the 1980's and interest in its wider application is quite recent. This delay is partially a result of the lack of a uniform standard for tape and players. This standard was only set in March 1988[3.7,3.9] .

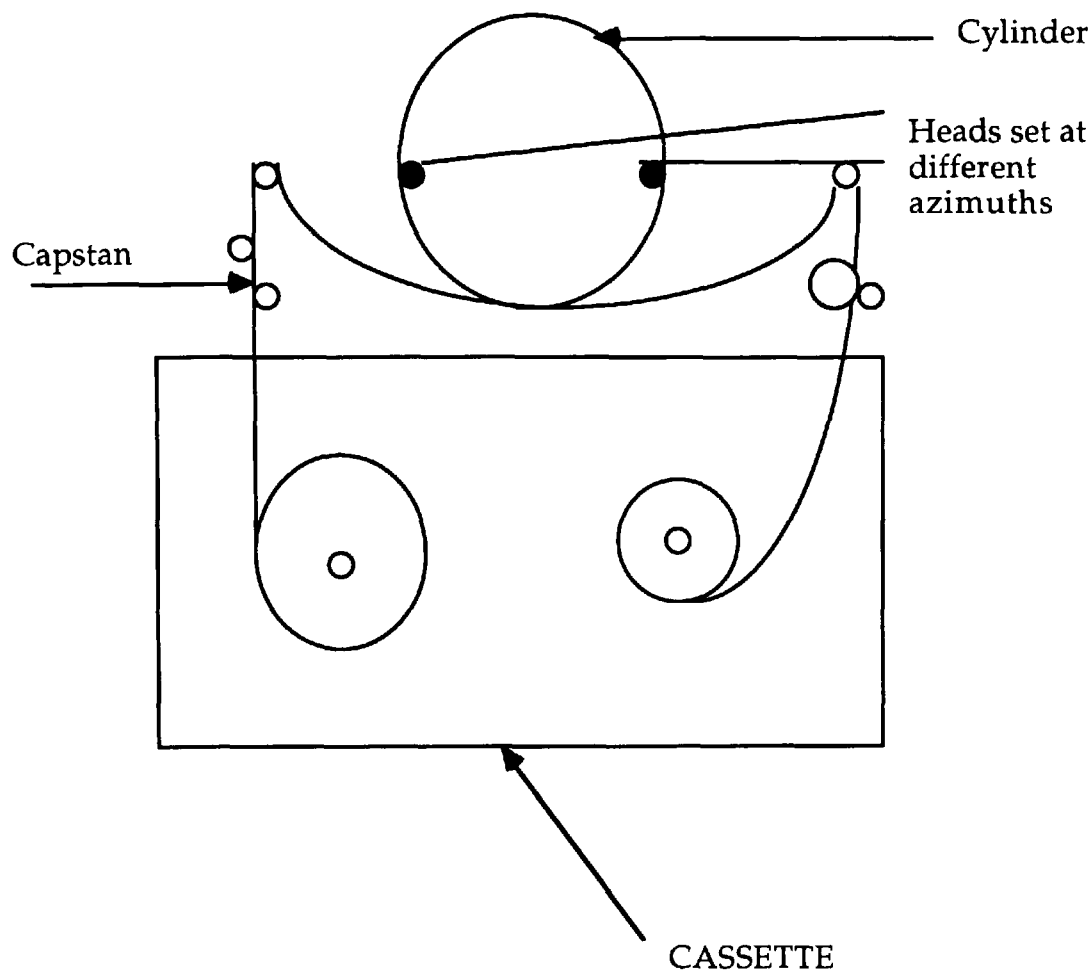


Figure 3.5 R-DAT Cassette and Replay Head[3.8]

The recent development of a format for data storage on R-DAT, Digital Data Storage, (DDS)[3.10] has encouraged a uniformity in approach to the resolution of errors within the system. The DDS system gives higher data integrity at the expense of a loss of data capacity and transfer rate. A comparison between the methods of audio and DDS is shown in Table 3.2. Standardisation has conferred considerable advantages upon R-DAT. By contrast neither the video cassette and the analogue audio cassette had any uniformity at their inception.

The R-DAT tape player employs a helical scan tape head shown in Figure 3.3A which is angled [3.16] relative to the tape passing over its surface. The data is then arranged in small fields which run diagonally relative to the general movement of

the tape through the player. R-DAT gains part of its integrity by the use of a read after write (RAW). This is achieved by using of multiple heads on the drum and a wrap angle of 90° [3.12]. This wrap angle is small compared with that employed by videos (180° to 270°) [3.11]. This enables the R-DAT player to:

- i) reduce tape damage and increase search speeds since only a small length of tape is in contact with the drum;
- ii) extend the life of the heads, due to low tape tension.
- iii) employ a four head system on the drum, in which the heads are separated by 90° . This will allow simultaneous monitoring.

Figure 3.3A showed how the head is inclined to give the fields running across the tape as shown in Figure 3.4. This diagram of the tape and head mechanism shows both the wrap angle of the tape around the drum and the positioning of the heads.

The tracks of data on the R-DAT recorder overlap each other. However overwrite and crosstalk errors are avoided by setting the heads at different azimuth angles. This angle is $\pm 20^\circ$ from the perpendicular to the direction of travel of the head. This enables the R-DAT cassette to possess high areal densities. The effects of the azimuths on the recording heads can be seen in Figure 3.6[3.12].

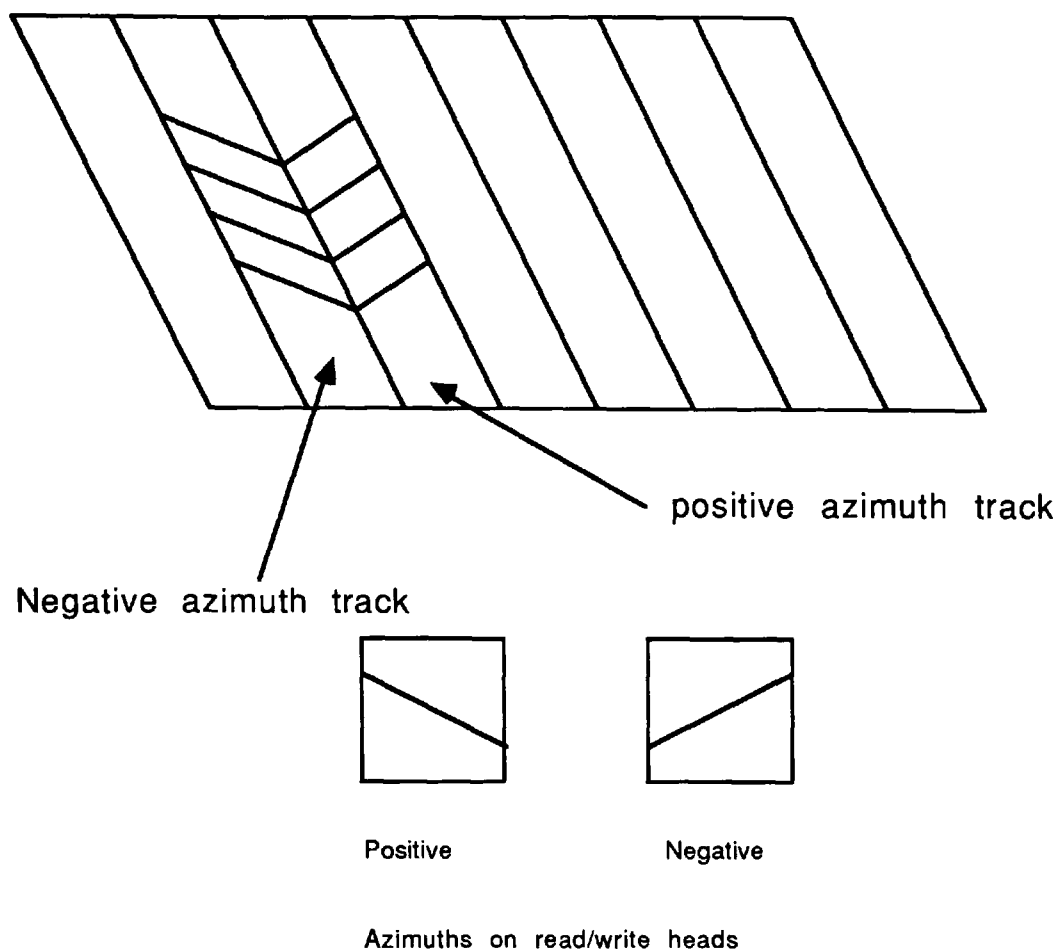


Figure 3.6 Tracks and Head Azimuth to Prevent Overwrite and Crosstalk[3.12]

The R-DAT cassette can store 1.3 Gigabytes of data on a completely sealed package measuring just 73mm by 54 mm by 10.5mm and weighing just 20g [2.9]. It is also a relatively cheap method of storage, costing about the same as a floppy disc but capable of storing 1000 times as much data. It also has the ability to find files using Random Access (RA).

Random Access enables the correct file to be found within 20 seconds compared with the minutes taken to locate a file on fixed head recorders[3.13]. The RA scheme of the R-DAT recorder does not trace the recorded tracks but it traverses across the recorded signals.

This traverse of the recorded tracks then produces a recorded envelope which will pick up only the tracks recorded with one azimuth. Hence valid data is read from alternate tracks, as shown in Figure 3.7. For fast random accessing of R-DAT the tape runs at 200 times its normal speed. This can lead to an increase in the bias on the tape. The tape and cylinder are maintained at a Constant Relative Velocity, CRV, by reference to the detected bit rate of signal at the head [3.16,3.15].

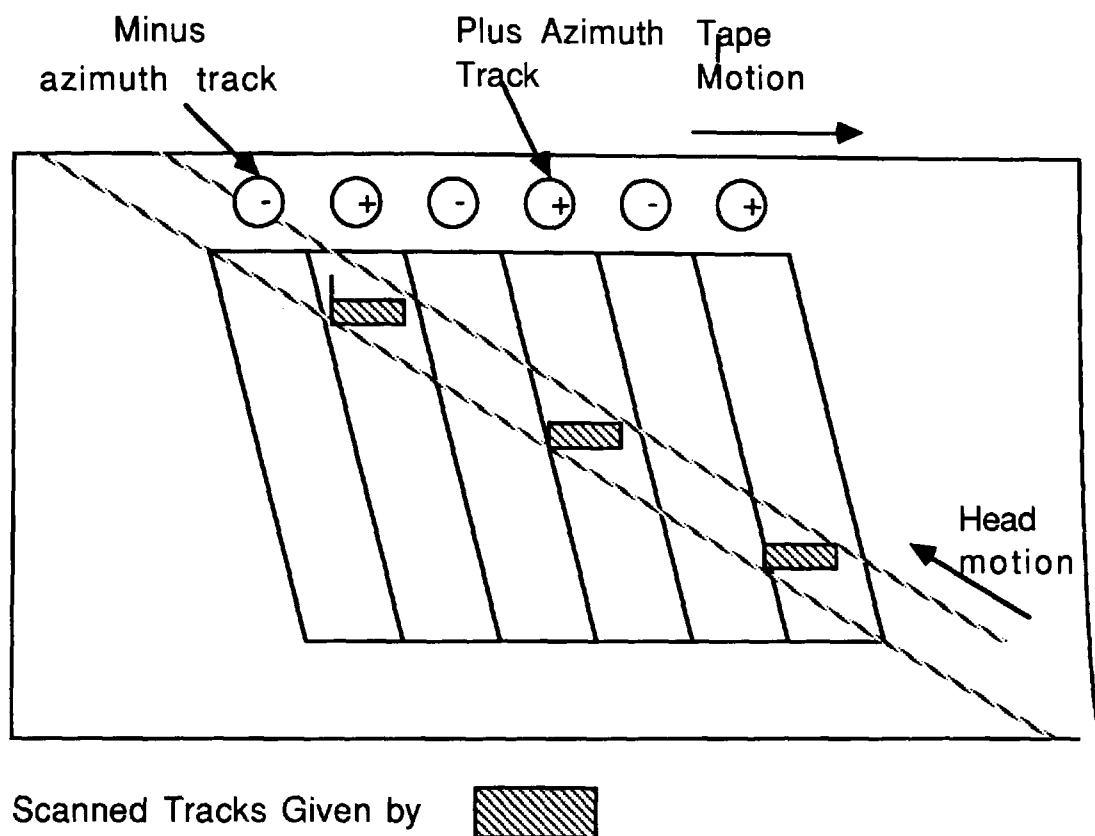


Figure 3.7 Scanning Technique used for Random Access R-DAT[3.13]

Furthermore the error rate for R-DAT is extremely low. With rates in the region of 10^{-15} it is comparable to the error rates of optical discs. Reel-to-Reel guarantees an error rate of only about 10^{-10} and cassette tape is only marginally improved at a rate of 10^{-11} [3.12].

ERROR SOURCE	ERROR MANAGEMENT
DROPOUTS	C1,C2,RAW,MEDIA
PARTICLES AND SCRATCHES	C1,C2,RAW,RETRY,MEDIA
SURFACE DISCREPENCIES	RAW,RETRY
TAPE WIDTH VARIATIONS	RAW,MEDIA
HELICAL DAMAGE	C3,EMR,CHECKSUM
LONGTITUDINAL DAMAGE	C1,C2
TRANSVERSE DAMAGE	C1,C2
TAPE SURFACE DAMAGE	MEDIA ,RAW
TRACKING ERRORS	DRIVE DESIGN
MECHANICLA JITTER	DRIVE DESIGN
ISI	RANDOMIZED DATA
RF FLUCTUATIONS	MEDIA
TAPE AGEING	MEDIA
HEAD CLOGGING	RAW,RETRY,C3,CHECKSUM

In the above:

C1,C2,C3 Error Correcting Codes ;

RAW Read After Write;

MEDIA Media Specification;

RETRY Retry on Read;

CHECKSUM Track Checksum[3.12,3.14]

Table 3.2 Error Management of the DDS System[3.14]

The instrumentation R-DAT recorder (DDS) is substantially the same as a domestic audio recorder with an additional third circuit board of error correction to give refined error correction. In addition to this extra level of error correction there are several methods of trapping errors which are dependent on the error mechanism concerned[3.14]. This is shown in Table 3.2:

At the current rate of development in the near future R-DAT will be able to fulfil all the requirements of the computer industry in at least five distinct areas of digital storage[3.9]. These are:

- i) to act as back-up storage for information stored on hard disc. This is due to R-DAT's high integrity, robustness, compactness, high areal density of recording and data accessibility;
- ii) software distribution. The relative size and robustness of R-DAT make it suitable for the distribution of software by post. Fast duplication processes presently being developed for R-DAT[3.9] will make this medium more attractive for distribution. The standardised format will also aid distribution as all recordings on R-DAT will be interchangeable unlike the various different formats for Video Cassette;
- iii) data interchange between computers. It's low cost, high data integrity and standard format make R-DAT a viable alternative to floppy discs and other kinds of tape for non-electronic mail;
- iv) file retrieval; R-DAT has an average file access time of 20 seconds, although slower than semi-conductor memories or hard discs it is faster than the access time for other tape media;
- v) hostile environments. A more rugged form of R-DAT will provide a robust alternative to the fixed head tape recorders which are presently used in hostile environments. Work has already been undertaken to create a

rugged Helical Scan Recorder based on domestic Video[3.3].

The signal processing and error correction of R-DAT will be discussed in Chapter 5. It uses Cross Interleaved Reed Solomon error correction to compensate for burst and timing loss errors. R-DAT allows easier editing than do traditional rotary head recordings[3.10].

Possible ways to increase the amount of information which can be stored on a R-DAT tape include[3.11,3.16];

- i) data compression, where data is stored at higher areal density on the tape to that presently used in the audio R-DAT recorder;
- ii) automatic tape changers to allow several tapes to run unattended for long periods;
- iii) an increase in the amount of data overlap to 60% from 33%. This will allow for a real increase in areal density of about 40% compared to that currently in use;
- iv) an increase in tape speed. This will lead to faster rates of data transfer, up to five times faster than at present.

3.3 Floppy Discs

This is the most common method now employed of storing computer data on magnetic medium. There is no agreed standard format[3.17] for floppy discs; disc sizes range from 3 to 8 inches in diameter. However due to commercial pressure certain disc sizes are more common than others. These are:

- i) the 3 ¹/₂ inch disc developed originally by Sony[3.20]. This has 135 tracks per inch and can store up to 360 Kilobytes of data for the single sided disc, at a packing density of 8000 bit/in. These discs suffer from the disadvantage that the drives are about 10% more expensive than those of the alternative 5 ¹/₄ inch discs;
- ii) the 5 ¹/₄ inch disc[3.20]. This is the most widely used floppy disc. However it suffers from problems of format incompatibility between the discs of different manufacturers. The 5 ¹/₄ inch disc has a capacity of 360 Kilobytes at a packing density of 5000 bit/in. As there are more manufactures of the 5 ¹/₄ inch disc it is cheaper than the 3 ¹/₂ inch version. In addition to the low price of the 5 ¹/₄ inch disc, the discs are becoming more reliable, though in comparison with the 3 ¹/₂ inch discs they are a fragile media with a limited useful life.

The 3 ¹/₂ inch version is more reliable than the 5 ¹/₄ inch. In addition, the smaller format has the advantages of the toughened plastic shell covering the media. [3.20].

The floppy disc is normally made from ferrous-oxide. However, this may be mixed with either cobalt or chromium dioxide to gain higher coercivity levels. It is anticipated that in the future, floppy discs will become a truly metallic medium gaining the same improvements in playback SNR as tape gained from a similar switch from oxide to metallic recording media[3.17].

Data formatting of a floppy disc is similar to the case of tape. Early methods of data storage relied upon a simple division of the disc into segments for the fields. Modern methods now use a fixed-bit-cell method of storage to enable an increase in data density of up to 40% [3.18].

On floppy discs the error detection is contained within the field. A cyclic redundancy check, (CRC) is used to ensure that the data in all of the blocks is correct. The CRC is held at the end of every field. It establishes the integrity of the preceding data by recomputing the CRC and comparing the result to the value on the disc. The error correction algorithm varies with the corresponding different types of disc from a simple parity check to an algorithm as complex as the Reed Solomon code.[3.19]

3.4 Thin Film Recording

3.4:1 Thin Film Structure[3.21]

It is important to distinguish between the two types of Magnetic recording, Thick and Thin Film. The Recording Media described above have all been Thick Film. That is to say that the Magnetic coat of the recording media is greater or equal to $3\mu\text{m}$. The depth of coat is an important factor in determining both, the recording density and the dominant sources of error in the media.

The structure of a thin film recording media is shown in Figure 3.8 below. It comprises of four different layers;

- Substrate;
- Undercoat;
- Magnetic coating;
- Overcoat;

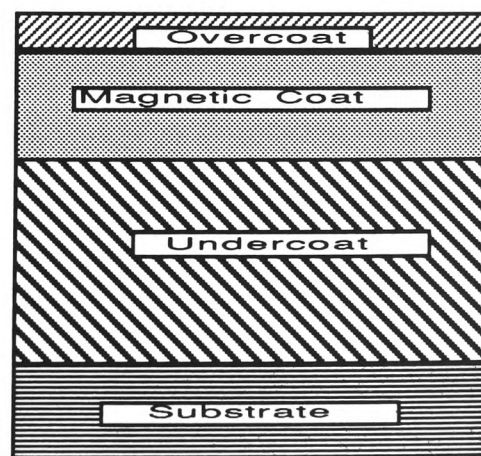


Figure 3.8 General Structure of a Thin film Media[3.2]

The substrate can be either rigid, such as Aluminium-Magnesium alloy, or flexible such as Polyethylene Terephthalate. Rigid substrates require a hard undercoat as the Aluminium-Magnesium alloy is too soft to provide adequate impact resistance for the magnetic coat. Flexible substrates require an undercoat of an adhesive layer to promote bonding of the organic substrate and the inorganic film. The magnetic film is normally a cobalt based metal alloy[3.21]. A hard, wear resistant, overcoat covers all thin film media to prevent the loss of the magnetic film by abrasion[3.2].

Types of thin film recording media may be distinguished by the process used to deposit the magnetic layer onto the media. Thin film media may be; plated, sputtered or evaporated, though recently multilayered media have been developed using two or more different methods of coating. [3.2]

3.4:2 The Winchester Disc System

The Winchester disc is approximately the same size as the standard 5 1/4 inch floppy disc but has a higher storage capacity; that of a floppy disc is only 360 Kilobytes, while a Winchester disc has a maximum storage capacity of 1 Gigabyte. Although it may be considered as being similar to the floppy disc the Winchester disc differs from the simpler technology in several important respects [3.22]:

- i) it is a thin film media with a magnetic coating of less than 3 μm . Thin film media are then prone to totally different error mechanisms than thick film magnetic storage discussed previously;

- ii) the read write head flies very close to the magnetic recording surface of the disc compared to the flight height of a floppy disc. This requires a flow of filtered air to keep the disc drives free of dust to prevent the head from crashing and to keep the head flying at the correct height;
- iii) Winchester discs allow a very high capacity for storing data compared with a floppy discs. This is due to the low flight of the read head, which also leads to the fast data retrieval rate of approximately 80 ms;
- iv) unlike floppy discs, Winchester discs cannot be readily transported. This is due in part to the low flight height of the heads and to the complex structure of the mechanism;
- v) until quite recently Winchester discs have been a prohibitively expensive method of storing data, as all data stored on a Winchester unit has to be backed up. This may be done using either tape or floppy discs which is in either case, time consuming and expensive when the Winchester disc is full;
- vi) the susceptibility of Winchester drives to vibration means that they can not be fitted to a portable computer.

A cut away diagram of the Winchester disc drive is shown in the Figure 3.9 the figure demonstrates how the Winchester system manages to pack so much data on to a simple system.

The use of the Winchester disc has been enhanced by the availability of R-DAT and Optical Discs for high performance, high density data back-up storage. The Winchester Disc continues to be primarily used as a data store for the computer.

The Winchester system can be improved in several ways, these include:

- i) a further reduction in the flying height;
- ii) the introduction of vertical bit recording ;
- iii) new formats, the micro Winchester disc and the Winchester cartridge disc.

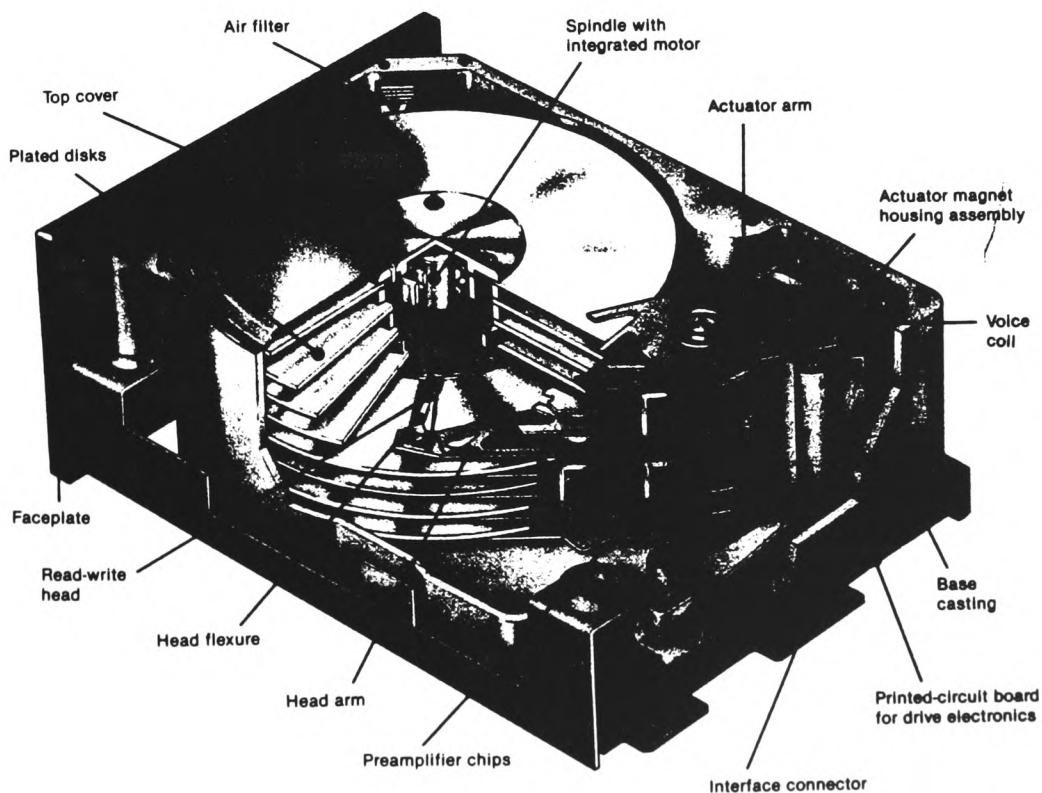


Figure 3.9 Winchester Disc System[3.22]

3.5 Optical Disc

Like R-DAT optical discs are a high integrity media. In both cases the integrity is due to the Modulation and Error Control coding. The storage of data in this medium is not magnetic; information is stored by a series of minute pits which are etched into the mirrored surface of the disc[3.27]. These pits are then read by a low powered solid state laser.

Laser light passes through a focusing mechanism consisting of mirrors, lenses and prisms. The light is then shone on to the disc where the pits diffract the beam. This diffraction is measured by an array of photo diodes[3.23]. The information is then extracted from the disc via these diodes which measure light intensity differences due to the diffraction caused by the pits.

In addition to the widespread use of optical discs for audio reproduction these discs can be used for a digital video recording system. The video disc system is based on a 12 inch disc[3.24,3.25].

In its application to data recording, the optical disc has been used mainly as a method of permanent storage as a CD-ROM. Like R-DAT the system can reliably store 550 Megabytes of data. More recently optical discs have been used to store both sound and video data for multimedia packages[3.26].

The qualities of optical discs outlined above imply that optical recording will become increasingly important in the field of data storage. These new fields will be in addition to its present use as a store for permanent databases which require

the large storage capabilities and fast recall. The database applications include the transcribing of British Telecom telephone directories and the use of CD-ROMs as electronic manuals by Siemens[3.26].

Data is formatted on to the Optical Disc as a sequence of frames. Each frame consists of 1 control byte, 24 data bytes, and two groups of 4 error correction bytes. One of these groups of correction blocks in the middle of the frame the other at the end.

As with other methods of high density recording the CD-ROM requires encoding algorithms which will compensate for both single random errors and more complex burst errors. Just as in R-DAT the Cross Interleaved Reed Solomon code (CIRC) combined with a group modulation code provide solution[3.26]. The Error Correction for optical discs is discussed fully in Chapter 4 while the Modulation is considered in Chapter 5.

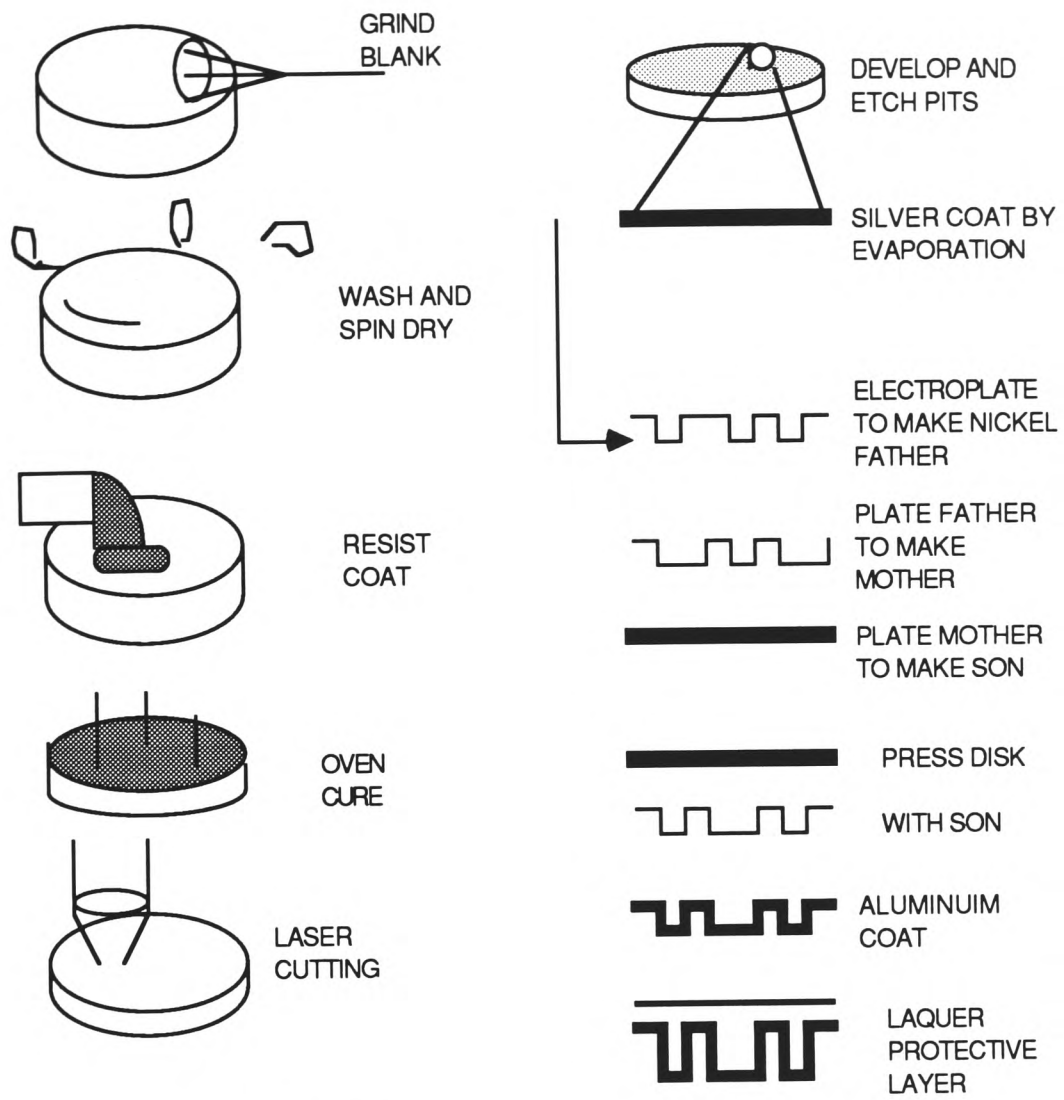


Figure 3.10 The Stages of Optical Disc manufacture[2.28]

REFERENCES

- 3.1) **C. D. Mee and E. D. Daniel (Editors)** "Magnetic Recording" McGraw Hill Volume 1 "Technology" Chapter 3 'Recording Media' E.Koster T.C.. Arnoldson p 230
- 3.2) **J. Watkinson** "The Art of Digital Audio" Focal Press 1989 Chapter 9 'Stationary Head Recorders' pp 297-349
- 3.3) **B.A. Winn** "Developing Helical Scan for Adverse Conditions" Proceedings of IEE Colloquium on Data Storage Technology, Savoy House London 13 Feb 1990 Paper 2
- 3.4) **R.A. Baugh D.J.Bromley and B.F Spenner** "Extremely Low Error Rate Digital Recording with a Helical Scan Recorder" IEEE Transactions on Magnetics Vol. MAG-22 No. 5 Sept 1986 pp 1179-1181
- 3.5) **P.S.Newby and J.L.Yen** "High Density Digital Recording Using a Videocassette Recorder" IEEE Transactions on Magnetics Vol. MAG-19 No..5 Sept 1985 pp 2245-2252
- 3.6) **Y.Yamada Y. Fujii M. Moriyama and S.Sathoi** "Professional Use PCM Audio Processor with High Efficiency Error Correction System" Presented at the 66th Audio Engineering Society Convention Los Angeles 1980 preprint 1628
- 3.7) **H. Nakajima and K.Odaka** " A Rotary Head High Density Digital Audio Tape Recorder" IEEE Transactions On Consumer Electronics Vol. CE-29 No. 3 Aug 1983 pp 430-435
- 3.8) **J. Watkinson** "The Art of Digital Audio" Focal Press 1989 Chapter 8' Rotary Head Recorders ' pp 248-296
- 3.9) **E. Tan and B.Vermeulen** 'Digital Audio Tape for data storage' IEEE Spectrum October 1989 pp 34-38
- 3.10) **T. Arai. M. Kobayshi T. Noghui and H. Okamoto** 'Digital Signal Processing for R-DAT' IEEE Transactions on Consumer Electronics Vol..CE-32 No. 3 August ` 1986 pp 416-424
- 3.11) **M.Finer** "Sony Digital Audio Tape" IBM Journal of Research and Development 198 pp 321-331
- 3.12) **B. Vermeulen** "Helical Scan and DAT a Revolution in Computer Storage' Hewlett-Packard Company paper
- 3.13) **T.Ardachi K.Arai K.Kwamoto H.Taki and K.Murashe** "A Fast Random Accessing Scheme for R-DAT" IEEE Transactions on Consumer Electronics Vol. CE-32 No. 3 August 1987 pp 275-285
- 3.14) **B. Vermeulen E Tan and T.Arai** "DDS- a Digital Data Storage format for R-Dat" Hewlett Packard/ Sony publication

- 3.15) **J.Watkinson** "Editing in R-DAT" Eighth VADR Conference. Birmingham 1990 pp 127-132
- 3.16) **H. Higgins** "High Capacity, High Reliability Data storage" Eighth VADR Conference. Birmingham 1990 pp 101-108
- 3.17) **C. Poulton** "Micro Floppy Media" Systems International March 1984 pp 51-52
- 3.18) **L.E.Thompson** "Floppy-Disc Formats" Byte Magazine Sept 1984 pp 147 and 430-445
- 3.19) **T. Sterling** "The Theory of Disc-Error Correction" Byte Magazine Sept 1984 pp 145-147
- 3.20) **S.Lewis** "Bits and Pieces" Practical Computing Oct. 1985 pp 100-114
- 3.21) **T.C. Arnoldussen, E.M. Rossi, A. Ting, A. Brunsch, J. Schneider and G. Trippel** "Obliquely Evaporated Iron-Cobalt and Iron Cobalt-Chromium Thin Film Recording Media" IEEE Transactions on Magnetics Vol. MAG-20 No. 5 Sept 1984 pp 821-824
- 3.22) **J.Voelchek** 'Winchester Discs Reach for a Gigabyte' IEEE SPECTRUM Feb 1987 pp 64-67
- 3.23) **G.Bouwhuis and J.J.Braat** "Video Disc Player Optics" Applied Optics 1 July 1978 pp 1993-2000
- 3.24) **P. Chen** "The Compact Disc ROM. How it Works" IEEE SPECTRUM Apr 1986 pp 44-49
- 3.25) **T.Oren and G.Kiddell** "The Compact Disc ROM Applications to Software" IEEE SPECTRUM Apr 1986 pp 49-54
- 3.26) **J.Kappetijn and G.Lutz** "Electronic Manuals Eliminate a Mountain of Paper" Siemens Engineering and Automation Vol. XIII No. 1. 1991 pp 4-7
- 3.27) **D. Cook** 'Optical Discs permanent Storage " SYSTEMS INTERNATIONAL pp 79-82
- 3.28) **J.Watkinson** "The Art of Digital Audio" Focal Press 1989 Chapter 13 'The Compact Disc' pp 440-486

Chapter4

Error Correcting Codes

4.1 Introduction

4.1:1 Introduction

This chapter will offer an account of the development of error correcting codes from both the communications perspective, where they were originally used by Hamming, through to their use in modern magnetic recording. It will discuss codes both for burst errors, where several adjacent bits are in error and the Gaussian noise channel, where the errors occur independently of each other. In addition to error correction, methods of enhancing the performance of error correcting codes, against burst errors will be shown.

Although this chapter will deal solely with binary codes defined over extended binary Galois Fields which are prevalent in digital recording. It is also possible to encode the data for any number of possible states provided that there exists a Galois field or an extended Galois Field for that state.

4.1:2 Correcting Errors by Adding Redundancy [4.1]

Wherever data is transmitted noise may corrupt the correct signal. For example it is possible to send a message with symbols drawn from a four symbol alphabet containing just the four two bit codewords or code vectors; (00,01,10,11).

However in this data set it cannot be ascertained if the signal received was that which was transmitted. The problem of data uncertainty is resolved by adding redundancy which is then used to detect signal errors. It is possible to add a single extra bit for each codeword and thereby enable the system to detect an error and then seek its retransmission. This gives two possible alphabets both with single error detection. Either the set of elements; (000,011,101,110) , which have even parity or the set; (001,010,100,111) ,which have odd parity.

Both of the above cases are not error correction codes although they will detect errors. In both the two sets for error detection given above there are at least two bits which are different for any pair of words in that set.

4.1:3 Information Rate

The Information Rate, R , of a code is the percentage or fraction of the transmitted bits which contain information. So for the single error detecting code, used above the Information Rate is 66% or $2/3$.

The Information rate of a code is important in comparing the efficiency of different error correcting codes. It gives a measure of the amount of redundancy in the error correcting code which can be measured against the number of errors the code can correct.

4.1:4 Hamming Weight

The number of ones in a codevector of binary data is called the Hamming weight of the codevector. The Hamming weight H_W of a codevector X is defined as:

$$H_W(X) = (\#i) \{x_i=1\} \quad x_i \in X \quad (4.1)$$

Where:

$\#$ is the number of elements in a set:

$\{ \}$ for which.

This can be shown to be the sum of the individual binary elements in the codevector.

4.1:5 Hamming Distance[4.1]

This definition of the separation of the elements into a code set was classified by R.W. Hamming[4.5] and bears his name. The Hamming distance, H , between two codewords X and Y is defined as being the number of places in which the codewords differ. This can be represented mathematically by;

$$H_D(X,Y) = (\#i + \#j) \{i: x_i=0 \text{ and } y_i=1 \mid j: x_j=1 \text{ and } y_j=0\}. \quad (x_i \in X \text{ and } y_i \in Y) \quad (4.2)$$

Where:

$\#$ represents the number of elements;

$:$ represents such that;

\mid represents and;

$\{ \}$ represents for which.

The Hamming distance can be used to find how much redundancy is necessary before the code set will detect/correct a given amount of errors.

For error detection a code will detect all errors in codewords which differ from the original codeword in less than $\frac{1}{2}H$ places. In addition a code will correct all errors in codewords which differ from the original codeword in less than $\frac{1}{2}(H-1)$. The number of differences between a received codeword and the original is said to be the length of an error. So for the example in section 4.1:2 a possible single error correcting code would be; (00000,10011,01101,11110).

The Hamming distances of these codewords can be tabulated to show that this code is a single error correcting code. The first row also gives the Hamming weight of the code words. The minimum distance of three shows that this code is a single error correction code.

	00000 i	10011 ii	01101 iii	11110 iv
i	0	3	3	4
ii	3	0	3	4
iii	3	3	0	3
iv	4	4	3	0

Table 4.1 Hamming Distance of a single Error Correcting code of Length 5

4.1:6 Varashamov -Gilbert Bound [4.2]

The use of bounds in Error Correcting Codes place limits upon the number of errors which may be corrected and the necessary amount of redundancy required so to do. These bounds are present in both asymmetric and symmetric error correcting codes.

The Varashamov bound gives an upper limit to the possible number of asymmetric code vectors which can be used in a single error correcting code of fixed codeword length. The BCH family of codes discussed in sections 4.2 and 4.3 obey the Varashamov bound. These bounds act as an upper bound on the number of single error correcting codevectors of a given length. The closer a code is to the Varashamov bound gives a measure of its efficiency.

The Varashamov-Gilbert bound is used to fix the efficiency of an Error correction code. Those codes whose code length coincide with the Varashamov-Gilbert bound are optimal single error correcting codes, giving the maximum number of codewords. This can be illustrated by Table 4.2 which shows the Hamming error correction codes and the Varashamov-Gilbert bound for different code lengths. Those Hamming codes which are in frequent use obey the bound.

Code Length	Number of Code words	
	Hamming Code	Varashamov Code
2	1	2
3	2	2
4	2	4
5	4	6
6	8	10
7	16	16
8	16	30
9	32	52
10	64	94
11	128	172
12	256	316
13	512	586
14	1024	1096
15	2048	2048
16	2048	3856

Table 4.2 Varashamov and Hamming bounds for codewords

4.1:7 Perfect Codes[4.1]

Perfect error correcting codes are those which allow the maximum amount of error detection from the minimum amount of redundancy. Error correcting codes are said to be perfect if:

the entire set of vectors in the vector space over which the error syndromes for the code is defined is partitioned into subsets each one of which contains one and only one element from the set of correctable errors;

in addition no member of another subset is as close in terms of Hamming distance to the subsets permitted element. This means that there are no elements in the vector field which do not lie within the specified Hamming distance of a member of the encoded data set.

Perfect binary symmetric codes are those which fulfil the Varashmov-Gilbert lower bound for their code length. The distinction between asymmetric and symmetric error correction codes will be shown in section 4.1:10. However the correction code will be dependent upon the type of errors present in the channel.

4.1:8 Parity Sum Check Error Detection in Computer Tape [4.3]

In Magnetic Recording there exist several simple non-algebraic codes which rely upon parity check mechanisms for their ability to correct errors. These redundancy checks have been used in digital magnetic tapes for computers where information is stored in a series of tracks. These codes include Optimal Rectangular Code , ORC [4.4] and Axial Cross Parity, AXP Code[4.5].

The first code ORC has been designed to correct errors on any single track. However this can be modified and with use of erasure pointers it may correct any double track errors on the tape. Codewords in ORC have a rectangular format, with the check bits positioned on two orthogonal sides of the rectangle

The AXP code was developed for high density recording tape. This tape has an 18 track format, which in the coding scheme are divided into two sets of nine tracks, consisting as in ORC of 7 data tracks and 2 parity bit checks. This system employs adaptive checks which can correct up to 3 erased errors in either set of 9 tracks up to a total limit of 4 tracks in the two sets. However the coding structure avoids complex Galois Field Theory by use of vertical and cross-parity checks.

4.1:9 Error Syndromes

Algebraic error correcting codes detect the position of both bit and byte errors by the use of an error syndrome. For any code there is a unique mapping between the set of correctable errors and the set of possible error syndromes. It is usual practice in error correcting codes to regard the zero vector as the error syndrome for zero errors.

4.1:10 Aliasing

With all methods of error correction aliasing will occur. It is most present in perfect codes as these do not allow any additional ability for error detection for errors of length greater than the code is able to correct. The effects of aliasing are dealt with in the section on Hamming codes Section 4.2.2, including a parity method to help prevent aliasing from occurring in the Hamming code.

4.1:11 Error Correction Codes for Asymmetric channels[4.6,4.7]

Although the main body of this chapter, sections 4.2 to 4.4 deal with Binary symmetric codes, there exist asymmetric correcting codes. These codes are of specific use with the ASC described in section 2.3:5. In this channel a 1 was transformed to a 0.

Varashamov and Tenegol'ts proposed a code for correcting a single asymmetric error. The code defined a weighted sum of a codeword to be:

$$W = \sum_{i=1}^n i a_i \pmod{(n+1)} \quad (4.3)$$

Where:

a_i is the codeword bit at point i .

The codewords for the Varashamov code are defined as being those words for which $W=0$. For example the vector:

$$A=(10011001)$$

is of length eight and is a codevector in the the Varashamov code. This can be shown as its summation from (4.3) is given by:

$$W=1+4+5+8=18=0 \pmod{9}$$

Single Asymmetric Errors are detected by the summation given in (4.3). The position of an error in the codeword is given by the summation. For a (1→0) asymmetric channel the modulus gives the position from the right hand side of the code vector.

$$A'=(10010001)$$

$$W'=1+4+8=13=4 \pmod{9}$$

However these codes are not acceptable in the BSC as two different errors will then be able to generate the same error syndrome.

To illustrate this it is necessary to use a code vector which is not symmetrical. An example of this is the codevector;

$$a = 00010101$$

$$W = 4+6+8=0 \pmod{9}$$

For a (1→0) error which occurs in the fourth position of the codeword one gets the following codevector and error syndrome.

$$a_0 = 00000101$$

$$W_0 = 6+8 = 5 \pmod{9}$$

For a (0→1) error which occurs in the fifth position in the codeword one gets the following codevector and error syndrome.

$$a_0 = 00011101$$

$$W_0 = 4+5+6+8 = 5 \pmod{9}$$

So clearly these codes are of little use in a binary symmetric channel.

Rao and Constantin [4.7] proposed group theoretic codes which are an improvement upon the Varashamov codes. These codes had a better information rate than the Varashamov codes in some cases exceeding the limit given by the Varashamov upper bound on Information rate.

4.1:12 Galois Field Theory[4.8]

All algebraic methods of error correction regard each codeword of data as a vector in a vector space. These vector spaces are governed by the rules of a Galois field. To be able to comprehend the working of an Error Correcting Code it is therefore necessary to first define a Galois field.

Mathematically a Galois Field (GF) has the properties that;

there exist two defined operations ; multiplication and addition,

the field is closed under these operations,

there exist a zero element within the field such that $\alpha + 0 = \alpha : \forall \alpha \in GF$,

there exists a Unity element of multiplicative identity such that

$$\alpha * 1 = \alpha : \forall \alpha \in GF,$$

there exist an additive inverse for every element, $\forall \alpha \in GF; \exists -\alpha \in GF$, such

$$\text{that } \alpha + (-\alpha) = 0,$$

there exists a multiplicative inverse for every element, $\forall \alpha \in GF; \exists \alpha^{-1} \in GF$,

$$\text{such that } \alpha * \alpha^{-1} = 1,$$

the elements of the field obey the associative, commutative and distributive laws,

An example of a Galois field is given by the multiplication table in Table 4.3.

Addition in this Galois field is standard modulo 2, an example of this type of addition is given by the two codewords: $101 + 111 = 010$.

		0	a	a ²	a ³	a ⁴	a ⁵	a ⁶	a ⁷
		000	010	100	011	110	111	101	001
0	000	000	000	000	000	000	000	000	000
a	010	000	100	011	110	111	101	001	010
a ²	100	000	011	110	111	101	001	010	100
a ³	011	000	110	111	101	001	010	100	011
a ⁴	110	000	111	101	001	010	100	011	110
a ⁵	111	000	101	001	010	100	011	110	111
a ⁶	101	000	001	010	100	011	110	111	101
a ⁷	001	000	010	100	011	110	111	101	001

Table 4.3 The Multiplication table for GF(2³) [4.9]

4.2 Single Error Correction

4.2:1 The Hamming Code. [4.8,4.10,4.11,4.12]

Hamming codes were discovered by Golay (1949) and Hamming (1950)[4.12]. Code matrices are used to both encode and decode the transmitted or stored data. These are Linear block codes which are defined over a Vector Space V .

One of the shorter Hamming error correcting codes is defined over the Galois field GF(2³). This is illustrated below:

A Generator or encoding matrix is the set of basis vectors for the linear block code. The row space of G is the linear code V . A vector is a code vector if it is a linear combination of the rows of G . An example of such a matrix is given by:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (4.4)$$

All generating matrices have a decoding matrix with which they are paired. This decoding matrix H of rank $n-k$ maps elements from the Vector Space V on to the null space V' . This null space is the vector set for the matrix H . The corresponding matrix for the above case this is given by:

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

A vector v is only in the original vector space V if it is orthogonal to every row of the matrix H . This implies that for all Generator and decoding matrices the product of the generator and the transpose of the decoding matrices is always zero, modulo 2,

$$GH^T = 0 \pmod{2} \quad (4.6)$$

To show how this works in practice, an example follows:

3.2:2 Example of Hamming Code

The data segment of information bits given by 1010 is encoded by post-multiplying modulo 2 by the matrix G . This operation gives the resulting codevector as being 1010101.

Pre-multiplying this vector by the matrix H , we obtain the error syndrome S_e . For no additional errors, the syndrome is the zero vector, (000). This is because all possible codevectors are mapped into the null space V' , this is illustrated in $GH^T=0$ above. This zero vector signifies that there is no error in the codevector.

Adding an error in the fourth position of the bitstream will give the following bitstream 1011101. This is not an element of the set of codevectors V , so when pre-multiplied by the matrix H it will not map into the null space. This gives a non zero vector, (111), which is the fourth row of the matrix H signifying an error in that position.

The columns of H , of the binary Hamming code consist always of non-zero elements of the Galois Field $GF(2^t)$ where t is the matrix dimension. It is standard practice to order the matrix so that the first t columns of the generator matrix is an identity matrix. H is then so arranged to allow this generator matrix.

4.2:3 Extended Hamming Code[4.12]

Pure Hamming codes are limited to the ability to correct single errors, but can be extended by the use of an additional parity check. This prevents double errors being corrected as single errors and hence causing aliasing. The matrix used to encode this extended version of the Hamming code is similar to that used for the normal Hamming with an additional row of all ones in the decoding matrix H' . The matrix given in 4.5 now becomes:

$$H' = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (4.7)$$

The decoding operation results in a double syndrome with two elements, (S_0, S_1) in which:

S_0 is the normal error correcting syndrome from the first three rows;

S_1 a parity check bit.

Returning to the example given in Section 4.2:2 of a code vector of 1010101 and impose a double error in bits 2 and 6. This will lead to an error syndrome of (111) from the single error correcting Hamming, this produces a false correction of bit 4 which is not in error.

By using the extended Hamming system with the additional parity check, the double syndrome (111,0) is generated. The zero parity check bit shows that there has been a double error. This parity check prevents a worse error being propagated by the decoder, without the addition of any extra redundancy in the transmitted signal.

The decoding algorithm for the extended Hamming code is given below:

$S_0=(000)$ and $S_1=0$ assume no error and accept the transmission;

$S_0 \neq (000)$ and $S_1=1$ assume a single error, correct for the error indicated by the syndrome S_0 ;

$S_0 \neq (000)$ and $S_1=0$ assume a double error and seek retransmission.

Aliasing is common to all perfect codes as there is no allowance for the detection of additional errors within the codevector. The extended Hamming code while maintaining the high information rate of a perfect code, does prevent double errors leading to aliasing.

4.3 Multiple Error Correction

4.3:1 Introduction

Codes which will correct more than one error within a codevector have been developed. However the facility to correct multiple errors reduces the information rate as the level of redundancy necessary to correct additional errors is increased[4,15].

It was shown in Chapter 2 that errors do not only occur as isolated single events[4,19]. Hamming error correction cannot control either multiple or bursty errors.

Several methods for the correction of multiple randomly generated errors have been developed. In particular the Reed Muller code developed by Reed (1954)[4.16] and the BCH code which was developed independently by Hocquenghem(1959) and Bose & Ray-Chaudhuri(1960)[4.17] .

4.3:2 Reed Muller Codes[4.16,4.18]

Reed Muller codes are an important class of multiple error correcting binary codes which are simple to decode by using majority logic. The Parameters of the Reed Muller codes are given below[4.18]:

$$\begin{aligned}
 \text{Length} & \quad n=2^m \\
 \text{Information Symbols} \quad k &= \sum_{i=0}^r \binom{m}{i} \\
 \text{Minimum Distance} \quad d &= 2^{m-r} \\
 \text{Error Control Capacity} & \quad 2^{m-r-1}-1
 \end{aligned} \tag{4.8}$$

The Reed Muller code of length $n=2^m$ and degree r is written $\mathcal{R}(r,m)$. These consist of the binary code vectors of length n , which are represented as polynomial over $GF[2]$, of degree at most r . The table below shows the truth table for length three.

X_0	0 1 0 1 0 1 0 1
X_1	0 0 1 1 0 0 1 1
X_2	0 0 0 0 1 1 1 1

Table 4.4 The truth table of the Boolean Functions ($m=3$)

For example the Reed Muller code $\mathcal{R}(1,5)$ has the generator matrix.

$$\mathbf{G} = \begin{bmatrix} 1 \\ X_0 \\ X_1 \\ X_2 \end{bmatrix} \begin{bmatrix} 1111111 \\ 01010101 \\ 00110011 \\ 00001111 \end{bmatrix} \quad (4.9)$$

This code is the extended Hamming code of length 7. The other Reed Muller codes commonly in use are the $\mathcal{R}(1,4)$ which is a (16,5) code and the $\mathcal{R}(1,5)$ which is a (32,6) code. The second of these codes corrects up to 7 bits in error and was used by the Mariner space probe to transmit pictures of the moon.

To decode the Reed Muller code a simple procedure of majority logic is followed. The binary points of the vector space \mathbb{Z}_2^3 form a cube as there are eight points. Planes are made up of a combination of four of the points. Each plane has a characteristic equation in which those points which are in the plane are represented by a 1 and those points not in the plane by a zero.

The received codevector is multiplied by the characteristic equation of each plane. If the resultant value is zero, even parity, then there is no error at any of points which make up the plane in the code vector. If the resultant value is one, odd parity, then there is an error in the code word. For the vector space \mathbb{Z}_2^3 there are 14 possible code planes, with each binary point lying in 7. Hence a single error would be displayed as 7 planes with odd parity.

The second stage is to compare the result for each of the lines in the planes against the received code vector. This is achieved by majority logic; since in the vector space \mathbb{Z}_2^3 each line lies in three planes. If the majority of these planes are of even parity then the line is said to possess even parity and vice versa.

4.3:3 BCH Codes

BCH codes are an algebraic multiple error correcting code and the Hamming codes discussed in Section 4.2 are a subset equivalent to the single error correction BCH code.

Once again the code makes use of the fact that the code vectors can be regarded as elements in a Galois field over a vector space. Two methods of encoding the BCH code will be described;

- (i) in Section 4.3:4, a block encoding method using matrices similar to that discussed for the Hamming codes in Section 4.2:3,
- (ii) in Section 4.3:7, a method of cyclic encoding using a set of polynomials which are irreducible over the Galois Field.

The cyclic method can be used as an alternative to generating the Hamming code by using matrices.

The length of these codes is always of the form $2^n - 1$, where n is an integer. The cyclic generator polynomials for these codes guarantee the largest minimum distance by the BCH bound, which is given below.

The BCH bound states that the generator polynomial $g(X)$ of a cyclic code of length n over $GF(q)$ whose roots:

$\alpha^{e_1}, \alpha^{e_{n-k}}$, where α is an element of the field of order n . The minimum distance is then greater than the largest set of consecutive integer modulo n in the the set $e=\{e_i, e_{n-k}\}$.

There exists a corollary to this statement which states that a cyclic code with roots $\alpha^{e_1}, \alpha^{e_2}, \dots, \alpha^{e_{d-2}}$, where α is an element of order n , has minimum distance d_i or greater provided that j and n are coprime.

A list of the shorter length binary BCH error correcting codes appears in Table 4.5 giving both the level of error correction and transmission rate for each code length:

t	n	k	R	E
1	15	11	0.73	0.067
1	31	26	0.84	0.032
1	63	57	0.90	0.016
2	15	7	0.47	0.133
2	31	21	0.68	0.065
2	63	51	0.81	0.032
3	15	5	0.33	0.2
3	31	16	0.52	0.097
3	63	45	0.71	0.048
3	127	106	0.83	0.024
4	31	11	0.35	0.129
4	63	39	0.64	0.063
4	127	99	0.78	0.031

Table 4.5 BCH Code Analysis

In this Table:

column 1 shows the number of errors correctable in each word;

column 2 the number of bits/word;

column 3 the number of data bits/word;

column 4 the transmission rate, $R=k/n$;

column 5 the correctional ability of the code $E=t/n$.

BCH codes maybe written as (n,k,t) codes where the symbols specify code length, data length and correctability.

4.3:4 Block encoding of BCH Code [4.22]

The Hamming codes are a special type of a BCH codes, those BCH codes which posses only the ability to correct a single error. The property that Hamming codes are a subset of BCH codes illustrates that the BCH codes can be generated by using matrices. These matrices are constructed by using the same method as those used for the Hamming code. For multiple correcting BCH codes a greater Hamming distance is used. This method of encoding BCH codes is simplistic and inferior to the more direct and elegant method given in Section 4.3:5 of using irreducible polynomials defined over a Galois Field to develop a cyclic encoding and decoding system.

4.3:5 Irreducible Polynomials over a Galois Field

A polynomial is said to be irreducible over a Galois field if it has no divisor polynomial from that field. For example; X^2+1 is not irreducible over $GF(2)$ as it

has divisor, $X+1$, though X^3+1 is an irreducible polynomial. A full list of these irreducible polynomials over $GF(2)$ up to degree 34 is given in Peterson and Weldon[4.21]

4.3:6 Cyclic property of BCH codes

BCH and Hamming codes are said to be cyclic. Cyclic codes have the following properties:

if the code vector $V=(a_1, \dots, a_n, a_0)$ is an element of the cyclic code, then the vector $V'=(a_0, a_1, \dots, a_n)$ obtained by shifting the units one space to the right is also in the code.[4.20]

4.3:7 Cyclic Encoding of BCH codes

The alternative method to develop BCH codes is to use a shift register to develop the codeword from the data block. In this method irreducible polynomials are used to generate the series of parity check bits. These parity check bits were developed in Block coding by using the non-unity columns of the matrix. This method shifts the data to the left, dependent upon the amount of redundancy required. The remainder from the division is added to the data block to create a multiple of the generator polynomial. To decode the received codevector the codevector is divided into the same polynomial which was used earlier to generate the parity check bits.

To generate and decode these codes using matrices would require matrices of dimension $(n-k, n)$ and (n, t) . Using the cyclic property of the BCH codes both

generation and decoding can be undertaken using division by irreducible polynomials. The method for which is displayed in the flow chart in Figure 4.1

The (31,26,1) BCH is generated by the polynomial x^5+x^3+1 . This use of generator polynomials gives a very fast and comparatively simple method of encoding BCH codes. Block encoding of these codes can involve matrices with dimension greater than a thousand. The single error correcting BCH code of length 1023 has 1013 data bits. It would be time consuming to develop such a matrix and any computation would involve complex matrix manipulation.

The (31,16,3) BCH code has a longer generator polynomial, because of the larger level of redundancy required $x^{15}+x^{12}+x^{11}+x^{10}+x^9+x^7+x^5+x^3+x+1$. This is the code which is used in the algorithm given in Figure 4.1. The code bits are generated by division of the data stream by the polynomial. The degree of the polynomial is given by the amount of redundant bits required, as these are the remainder after the data block has been divided by the polynomial. The number of cycles through the loop is determined by the number of bits in a given data block.

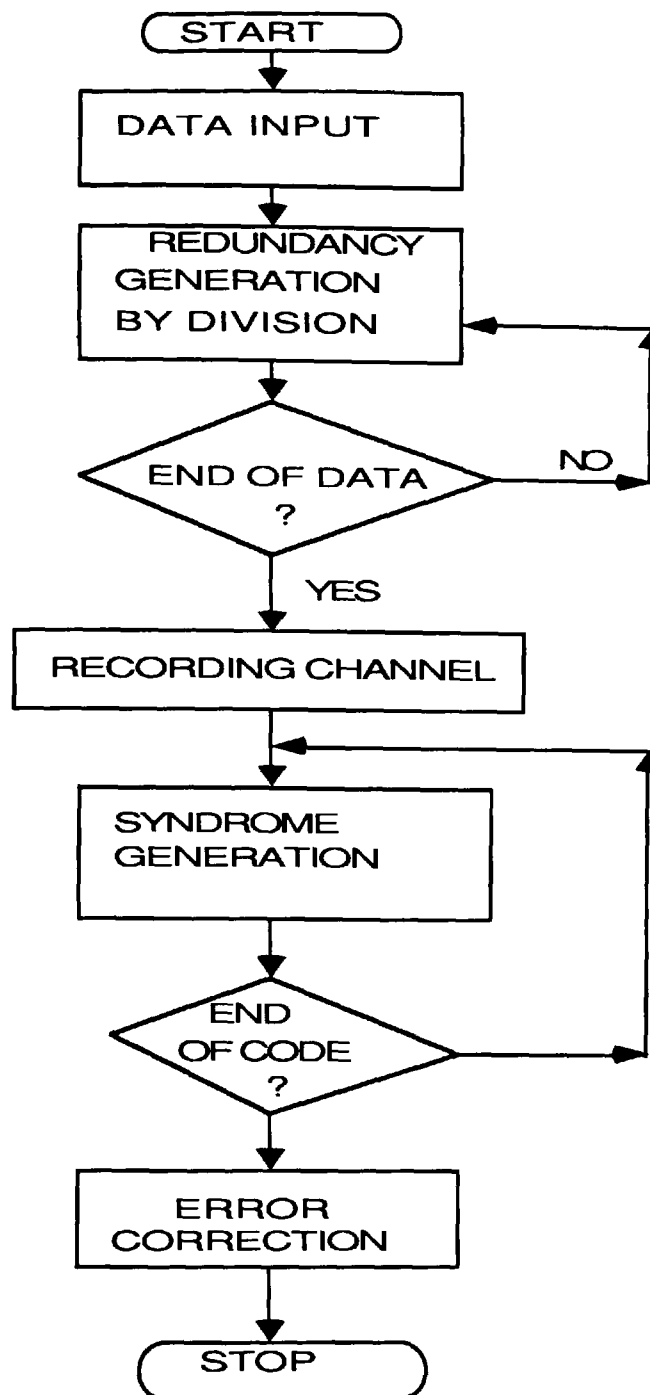


Figure 4.1 Cyclic Encoding and Decoding of BCH Code

4.3:8 Erasure Decoding with BCH codes[3.22].

One of the most important error mechanisms is the dropout or fade[4.19]. In the case of a severe dropout there is a total loss of the signal in that section of recorded data. However designing a code specifically to deal with this type of error is not appropriate for a Gaussian noise dominant channel.

In addition to being able to cope with multiple errors caused by bitshift there is a modified form of the BCH code which allows erasures to be handled as well. To decode codevectors which have suffered from erasures it must be known that a symbol should occur at a certain position, though it is not necessary to know what that symbol should be. This is particularly useful when combined with a Modulation code such as the Manchester code.

One method of erasure decoding uses a maximum likelihood decoder to compare the received code sequence without the erasures with all possible sequences.

With E erasures, in a code vector there is still a minimum distance of $d_{\min} - E$ for the unerased positions. It is therefore possible to gain a decoding as long as the now reduced minimum distance still exceeds $2t$. This method of erasure decoding is not the only one. An alternative is to duplicate the received signal.

For one of the two duplicate signals replace all the erased bits by ones and in the other by zeros. Decode each one separately using a standard BCH decoder. If the sequences differ then it is logical to choose the one which is closer to the received signal, hence accepting the one with the least amount of induced errors indicated.

4.3:9 Bursty Errors and False Error Correction

BCH codes are linear codes which are able to decode multiple errors from a Gaussian noise source. Random errors may be corrected wherever they occur in the received codeword, provided that the total number of errors in the codeword does not exceed the limit set by the Hamming distance.

The probability of erroneous decoding of a data stream for a Gaussian noise source by a single error correcting linear code is given by [4.7] ;

$$P_{\epsilon}(e) \approx \binom{n_1}{2} \alpha^2 + \binom{n_0}{2} \beta^2 \quad (4.10)$$

Where :

$P_{\epsilon}(e)$ is the probability of erroneous decoding;

n_1 is the Number of 1's in the data stream;

n_0 is the Number of 0's in the data stream;

α is the probability of detecting a one as a zero;

β is the probability of detecting a zero as a one;

In this it has been assumed that $1 \geq \alpha \geq \beta \geq 0$.

In this case the probability of a one being in error may not be equal to a zero being in error.

Equation 3.10 can be expanded for multiple correcting codes such as the BCH code. Given that the code can correct $r-1$ errors then the probability of false detection can be given by:

$$P_{\epsilon}(e) \approx \binom{n_1}{r} \alpha^r + \binom{n_0}{r} \beta^r \quad (4.11)$$

However not all errors are caused by a Gaussian error source so it is often necessary to employ non-linear codes which are able to correct these burst errors. The BCH code is a linear code, were it used to correct burst errors the redundancy required would be excessive. This would lead to a very low transmission rate and in addition an impracticably large database containing the error syndromes, for

example for the (31,16,3) BCH code there are 113491 possible error syndromes.

Furthermore were the burst error of greater length than the code can correct, aliasing would occur similar to that found in Hamming codes when multiple errors were introduced.

4.4 Burst Error Correcting Codes

4.4:1 Introduction

To combat the problems of burst errors and dropouts, an extension to the BCH code was proposed by G. Solomon and I.S. Reed [4.23] in 1960. This code takes its coding elements from extended binary Galois Fields which have the number of field elements in code vectors. Thus for a code of length 7 the elements used are those of the extended Galois field $GF(2^3)$, with the proviso that none of the data elements are zero. The multiplication table for this field is shown in Table 4.3.

4.4:2 Burst Error Correction

The Hamming distance is a useful measure of performance for codes in a Gaussian Noise channel but it is a less appropriate where the main error source is non-Gaussian and Bursty. Furthermore the use of modern Modulation codes (see Chapter 5) will lead to a high number of bit errors from a relatively low channel error rate.

This can be compensated for by using codes which treat the data as a series of discrete blocks rather than a continuous sequence. These code types include:

B-adjacent [4.26] and Reed Solomon codes[4.23]. The later of these is used extensively in such modern high integrity recording media as Optical disks and R-DAT. A discussion of Reed Solomon codes will be given in Sections 3.4:5 to 3.4:8. Here alternative strategies are reviewed.

Although codes do exist for both bit and byte error correction [4.22] in this section all the codes discussed here will be pure byte correcting codes. These correct a block or a byte of data rather than a single bit or symbol.

4.4:3 Fire code[4.26,4.27]

This code was developed in 1959. In common with the BCH codes this uses a Generator polynomial of the form:

$$g(X) = (X^{2^{\beta}-1} + 1) h(X) \quad (4.12)$$

where:

β is the length of the burst to be corrected;

$h(x)$ is an irreducible polynomial of degree $n \gg \beta$ which does not divide

$X^{2^{\beta}-1} + 1$;

These codes use a shift register to trap and locate the burst, needing a maximum of r shifts to do so where r is the position of the start of the burst. Although this code is designed to correct a single burst error determined by the degree of the polynomial, Fire codes regard the data as a continuous stream of bits rather than as a series of discrete blocks as the Reed-Solomon and B-adjacent codes do.

4.4.4 B-adjacent codes,[4.26,4.27]

These were developed in 1970 by D.C. Bossen. This code allows the correction of two bursts.

An example of this code follows:

Six data words A-F of length 8 bits are made into a code word by the addition of two redundancy words.

The first P is simply the exclusive OR of the data bits:

$$P = A \oplus B \oplus C \oplus D \oplus E \oplus F \quad (4.13)$$

This second redundancy word is then given by:

$$Q = T^6A \oplus T^5B \oplus T^4C \oplus T^3D \oplus T^2E \oplus TF \quad (4.14)$$

where T is a transformation Matrix given by:

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The words now form a code word and are recorded. If a double burst error is introduced in words A and C, then the P part of the syndrome would have the exclusive OR of the two errors while Q part will contain $T^6A \oplus T^4C$ so giving two equations in two unknowns which can be solved directly.

The B-adjacent code like the the Reed Solomon code can be regarded as a byte correcting code, correcting a group of data rather than a single bit. B-adjacent codes can be regarded as a subclass of Hamming codes with the following parity check matrix.

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \dots & 0 & 1 \end{bmatrix} \quad (4.15)$$

Where $\beta_1 \beta_2 \dots \beta_{k-1}, 1$ are all distinct elements of $GF(2^b)$.

These codes are referred to as 2-redundant codes as there are two parity check bits at the end of the matrix.

4.4:5 Definition of Reed Solomon Coding[4.9].

The Reed Solomon codes are again an extension of BCH coding in which the subwords come from the extended binary Galois field. The choice of Galois field is determined by the length of the code vector used. For example a code with thirty-one subwords will take these subwords from the Galois field, $GF(2^5)$ generated by an irreducible polynomial. In the case of $GF(2^5)$, this polynomial is given by X^5+X^2+1 and the subwords will all be of length five bits. The code then develops parity check subwords by using the multiplicative and additive properties of the Galois field discussed in Section 4.1:12.

This is illustrated by the use of an example in Section 4.4:6, however it is first necessary to define the Galois field $GF(2^3)$ in terms of the generating element and the binary bit pattern which the words represent. This is shown in Table 4.3.

4.4:6 Example of Reed Solomon Encoding[4.29]

For a data stream of fifteen bits, a single byte error correction Reed Solomon code defined over $GF(2^3)$ can be used. The table for which was given in section 4.1:12.

The data stream is first separated into five bytes of length three, which are used to generate the parity from equations 4.16 and 4.17. This provides the addition of two extra subwords at the end of the set of data bytes. The code now has the ability to detect, quantify and locate errors in the codeword.

Take the data stream:

101011110100001

Divide it into five words of length three:

101 011 110 100 001

Convert these to the representation in terms of α from Table 4.3:

α^6	α^5	α^4	α^0	α^2
A	B	C	D	E

Find the first parity check word by summing the data words:

$$P = \alpha^6 A \oplus \alpha^5 B \oplus \alpha^2 C \oplus \alpha^5 D \oplus \alpha^3 E \quad (4.16)$$

$$P = \alpha^5 = 111$$

Using the Equation that

$$Q = \alpha^2 A \oplus \alpha^3 B \oplus \alpha^6 C \oplus \alpha^4 D \oplus \alpha E \quad (4.17)$$

$$Q = \alpha^4 = 110$$

The data subwords and the parity check subwords will now form the two

equations for the generation of the error syndromes:

$$S_0 = A \oplus B \oplus C \oplus D \oplus E \oplus P \oplus Q = 0 \quad (4.18)$$

$$S_1 = \alpha^7 A \oplus \alpha^6 B \oplus \alpha^5 C \oplus \alpha^4 D \oplus \alpha^3 E \oplus \alpha^2 P \oplus \alpha Q = 0 \quad (4.19)$$

This can now be used to correct both errors and erasures from the recorded data, with the situation where there are no errors given by two Zero syndromes shown above.

As an example of error correction. First introduce an error into the third word, C, so that the corrupted word, C', now reads as 101.

To find the error take the binary summation of the elements over the Galois field. This gives the first syndrome representing the amount of error in the system.

Where:

$$S_0 = 101 \oplus 011 \oplus 101 \oplus 100 \oplus 001 \oplus 111 \oplus 110 = 011 \quad (4.20)$$

$$S_0 = 011 = \alpha^3.$$

Then use the second equation to generate S_1 . For this example this yields:

$$S_1 = 100 \oplus 100 \oplus 110 \oplus 101 \oplus 011 \oplus 101 \oplus 110 = 010 \quad (4.21)$$

$$S_1 = 010 = \alpha.$$

To locate the position of the error it is necessary to divide S_1 by S_0 to give the position of the error:

$$S_1/S_0 = \alpha/\alpha^3 = \alpha^{-2} \equiv \alpha^5 \quad (4.22)$$

This gives the word which was multiplied by α^5 as being in error. So to correct the code S_0 is added to C' to give the original value C .

One of the important advantages of Reed Solomon coding is that it allows the decoding of more than one burst error. For example there is a double error correcting (7,3) code over $GF(2^3)$. The level of redundancy required however restricts the amount of data to only three subwords and will correct up to two subwords in error[4.30].

This code is not perfect so it possesses the additional ability to detect errors of greater length than it can correct. If there are three sub-words in error, it is often the case that this double error correcting code will be able to detect the error. This is due to the generated error syndrome not belonging to any of the vector subspace of possible syndromes associated with a particular double or single error. In this case the code will work as a triple error detecting code rather than attempting to correct two bits which are not in error.

3.4:7 Reed Solomon coding in High Integrity Recording Media [4.31,4.32,4.33]

The modulation code used in high integrity Media (such as the 8/10 modulation, used in R-DAT), causes error propagation. In this case a single bit error in the output could after demodulation become a bursty error of up to 8 bits in the replay signal. Reed Solomon over $GF(2^8)$ was therefore an obvious choice for the error correction coding to be used.

The signal processing and error correction for the Digital Audio tape has been undertaken in relation to digital recording for consumer electronics as part of the application of R-DAT to digital audio . The method of error correction employed in R-DAT is a doubly-encoded Reed Solomon code over $GF(2^8)$. A data frame consists of 128 blocks, which include all the sampled data during a half rotation of 15 m.sec [4.31].

Error correction is undertaken by two concatenated Reed Solomon codes referred to as the inner and the outer codes. These codes regard the data as a series of blocks of length 8 bits. The inner code, C1, is a (32,28) R.S.C. having a minimum distance of 5 so enabling the correction of two erroneous blocks. The outer code, C2, with a four block interleaving in the horizontal direction is a (32,26) RS code with a minimum distance of 7 so enabling the correction of three erroneous blocks. The concatenation of these two codes is shown in figure 4.4. Due to the high packing of R-DAT there is a tendency for long burst errors, so a high ability for both error detection and correction is required in the digital audio systems.

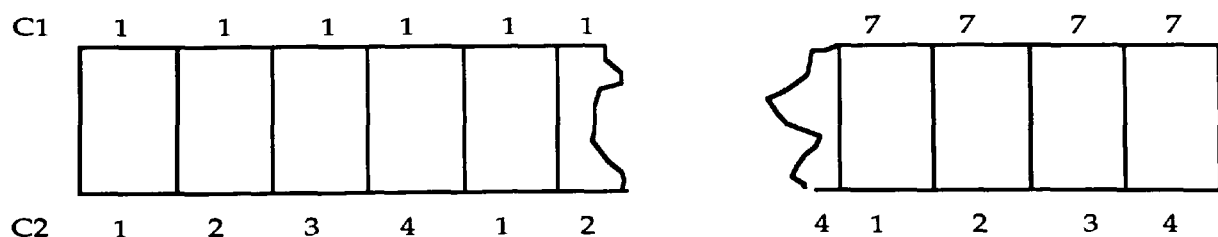


Figure 4.4 Section of recorded tape from R-DAT showing the Code Concatenation

This method of encoding is similar to that used in the optical disk, being an extension of optical disc encoding. In addition there is a model of the error decision process in the flow charts for both the inner C1 code and outer C2 code, in Figure 4.5 and 4.6 at the end of this chapter.

4.5 Convolutional Codes[4.34,4.35,4.36]

4.5:1 Introduction

These are an important but totally distinct group of Error correcting codes. The distinction of Convolutional codes is due to the fact that the algorithm for producing an encoded bitstream is totally different to either block or cyclic code construction which were discussed in the previous sections. The ability to correct errors in Convolutional coding does not come from additional redundancy at the end of each data word but in the dependence of the present output state to the previous output states.

Part of the importance of these codes is due to the large amount of work which is currently being undertaken to find a method of combining Error Correction Code with Modulation Codes. The aim of this research is to combine the redundancy which occurs in both Modulation and Error Correction to give a single approach to the problem Channel and Error Control encoding. This is dealt with in greater detail in Section 5.8.

In the simplest form a convolutional code will take a single bit of input data and map it onto two bits of output data using a polynomials of order three to represent the shift registers in the coder. For this type of encoder it can be said that:

k is the number of input bits = 1

n is the number of output bits = 2

$N-1$ is the number of dependent blocks = 2

k/n is the code rate

There exist both random and burst error correcting convolutional codes, though in this section only random error correcting burst codes will be discussed.

4.5:2 Encoding of a Convolutional Code

Like all error correcting codes convolutional codes make use of generating polynomials over which they are defined. In convolutional codes these polynomials are a representation of the shift register which develops the code. This is shown in Figure 4.7 for a three stage shift register with three output bits and a single input.

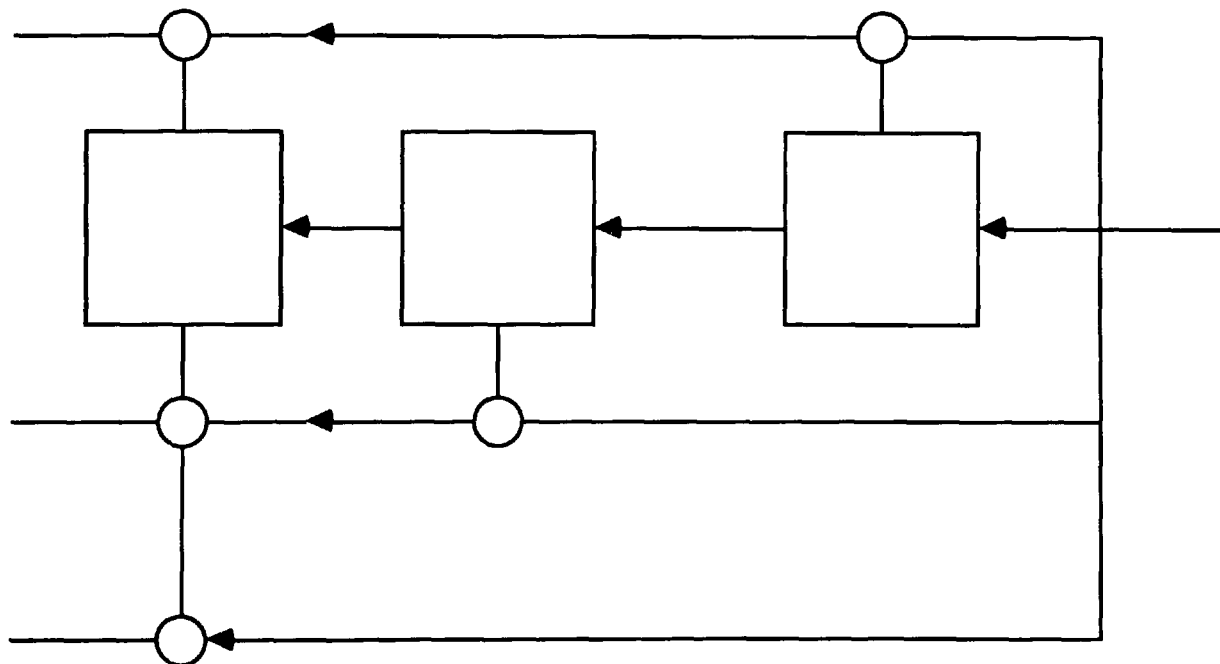


Figure 4.7 Convolutional encoder for code given by the polynomials, X^3+X^2+1
 X^3+1 & X^3+X+1 [4.32]

In this diagram the bits enter the shift register from the right, the elements in the boxes are then shifted one space to the left with the bit on the left being lost. The encoded sequence then leaves by the three output lines on the left this gives a data

to code rate of 33.3%. The polynomials define the feed back through the loops. These polynomials have to be coprime, as ones which are not will lead to catastrophic error propagation, (CEP). CEP is caused when two distinct input sequences merge to give the same output sequence. In this case, it is not possible to derive which of the two input sequences the received output sequence was derived.

An example of a data stream and its encoding by this method of error correction coding is given below:

Shift registers: $P_1 = X^3 + X^2 + 1$

$$P_2 = X^3 + X + 1$$

$$P_3 = X^3 + 1$$

Data sequence: $1 \setminus 0 \setminus 0 \setminus 1 \setminus 0 \setminus 1 \setminus 1 (0 \setminus 0 \setminus 0)$

Encoded Sequence: $111 \setminus 010 \setminus 100 \setminus 000 \setminus 010 \setminus 011 \setminus 010 \setminus (110 \setminus 011 \setminus 111)$

The last n terms are always zero to clear the shift register

4.5.3 Catastrophic Error Propagation

A method of checking for CEP is the, Encoder State Diagram, this shows the values of the states and hence the output sequence to any given input sequence. This is shown in Figure 4.8A. For the convolutional encoder given by $X^2 + 1$ and $X + 1$ these equations are not coprime as $(X + 1)^2$ is equal to $X^2 + 1$. This can be compared to the encoder state diagram for $X^2 + X + 1$ and $X^2 + 1$ which is shown in Figure 4.8B.

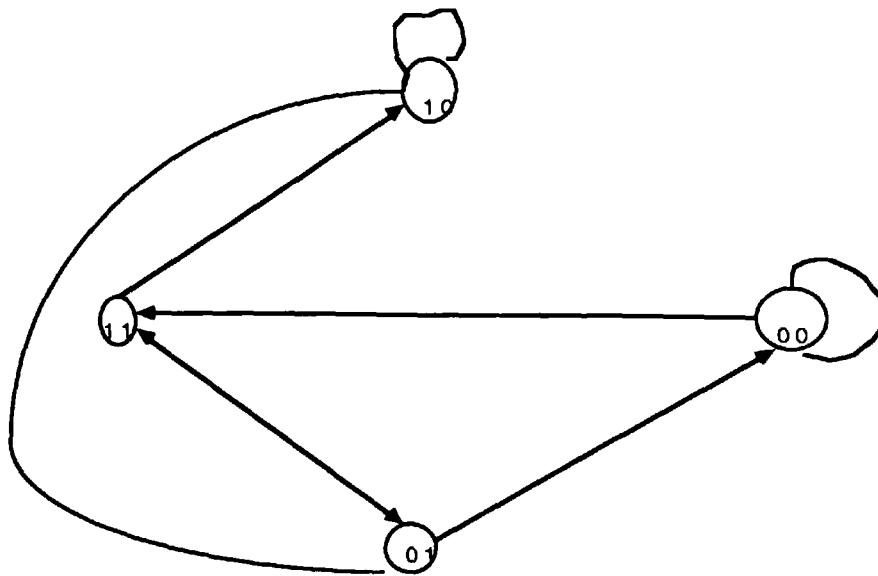


Figure 4.8A Encoder State Diagram for a code with CEP

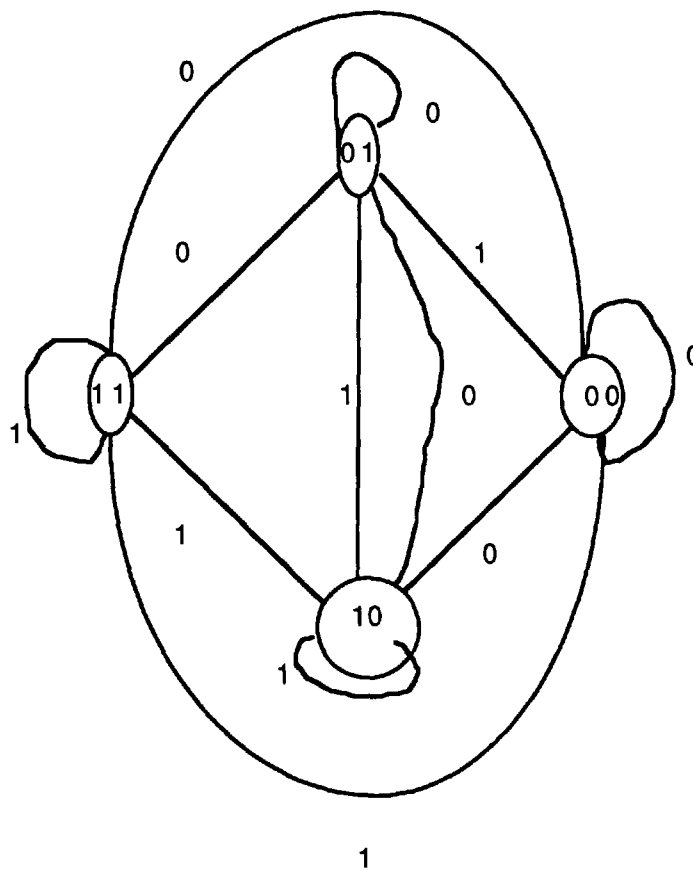


Figure 4.8B Encoder state diagram for Convolutional code without CEP.[4.34]

The numbers at the nodes represent the states in the blocks, and the numbers on the loops represent input bits to the block diagram.

4.5:4 Maximum likelihood and the Viterbi algorithm

As Convolutional code maps a data stream into a matrix, it is often necessary to denote the output bits by a series of vectors. The maximum likelihood estimator is a method of gaining as close a match to a generating data sequence as possible from a received encoded signal. If a decoder is said to find \bar{u} an estimation of the original data sequence u and \bar{v} an estimation of the sequence v sent down the channel from the received signal r . Then

$$P(E|r) = P(\bar{v} \neq v | r) \quad (4.21)$$

$$P(E) = \sum_r P(E|r)P(r) \quad (4.22)$$

To minimise $P(E)$ it is necessary to minimise $P(E|r)$, hence minimisation of $P(\bar{v} \neq v | r)$ which is equivalent of maximising the function $P(v | r)$ using the standard log-likelihood function[4.37].

However it is necessary to have a fast and relatively simple decoder for these convolutional codes with the plain maximum likelihood decoder, for a data sequence of length L generated by a N_j stage feedback loop, there are then as many as 2^{LN} possible paths through the set of received.

This leads to a method of being able to disregard non optimal paths as soon as they become non-optimal. This algorithm was discovered by A J Viterbi in 1967 it simply states that:

- 1) For each of the 2^v stored paths compute the distance between the received frame and the 2^N branches.

- 2) For each of the 2^v nodes construct the 2^N paths which terminate at that

point. Select and store the best.

This can be represented by updating a table after each received message frame, containing;

the end state;

the start state;

output;

the distance between the output and the received frame;

the accumulated distance to the start state and the sum of the last two values.

It is this final value which will determine whether a path should be kept. The lower the value then the better the path. In cases of a tie then the choice is made arbitrarily.

4.6 Coding Enhancements

4.6:1 Interleaving[4.38]

The Reed Solomon code can now be enhanced by interleaving the individual code sub-words so that successive sub-words in the same track on the media are not from the same codeword. This is illustrated below with subwords from three separate codewords. These are interleaved to gain a more efficient structure for combating a burst error of the type shown below in Figure 4.9 [4.38].



Sub-word from code word 1



Sub-word from codeword 2



Sub-word from codeword 3



Sub-word in error

Figure 4.9 Interleaving to combat long burst error

Interleaving can be applied to any type of code. As in the above example interleaving is most effective when it uses data which has already been separated into sub-words or bytes for the encoding, such as the Reed Solomon or B-adjacent codes.

4.6:2 Cross-interleaving [4.39,4.40,4.33]

Cross-Interleaving is used in both of the high integrity Media of R-DAT and Optical disks. In this case two codes are interleaved across each other as shown in the Figure 4.11. Similar methods have been used in the transmission of signals such as the the Teledion System developed for the Canadian Broadcasting Company, which is shown below in Figure 4.10.

Cross-interleaved data can be modelled as being stored in the form of an array. The codes enmesh perpendicular to each other, the Interleaving occurs with bits down the strip of recorded data. So given that the large bursty type errors occur during the read-write process the cross-interleaving will be far more effective than the more simple interleaving techniques. Furthermore, Cross Interleaving gives protection against random single errors.

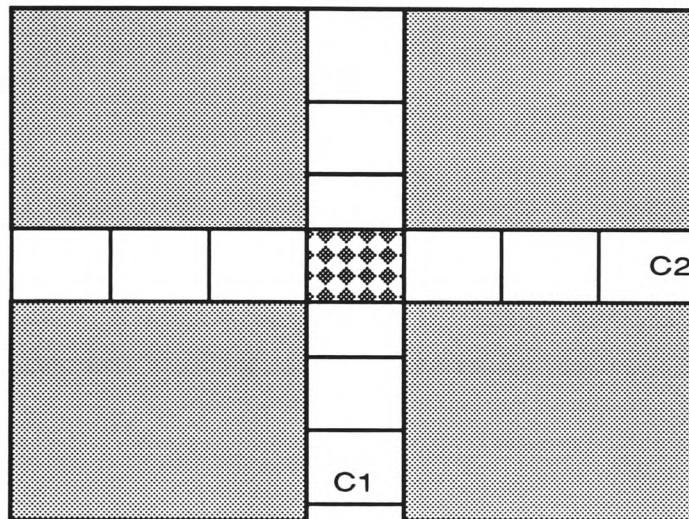


Figure 4.10 TELIDON SYSTEM OF INTERLEAVING

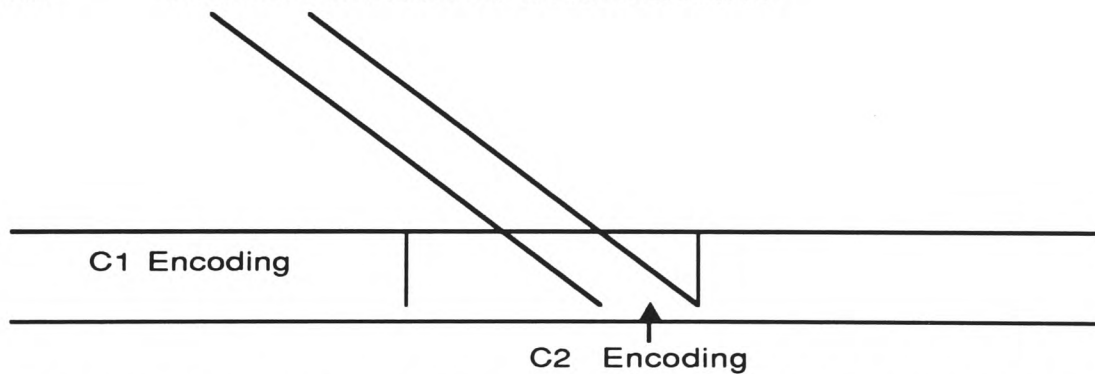


Figure 4.11 CROSS INTERLEAVING OF REED SOLOMON CODES IN HIGH INTEGRITY MEDIAS

4.6.3 Concatenated Codes

It has been shown that it is possible to combine two different types of Error Correcting code into one algorithm. In the use of two Reed Solomon codes in the High Integrity Recording Media[4.33]. However, it is possible to use two different types of Error Correction code and Concatenate these two codes together. For example a Convolutional code maybe used as an inner code together with an outer code with byte correcting properties [4.41].

In such a system the convolutional coding is applied to the data first. The coded sequence is then re-encoded by the outer Reed Solomon Coding. It has been shown that this coding method satisfies the Gilbert-V~~as~~hamov lower bound [4.42].

An additional possibility is to encode the raw data using (7,4,1) Hamming code and then to further encode by the use of Reed Solomon code encoding the bits over $GF(2^7)$. So using one Hamming codeword as a data byte for the Reed Solomon code.[4.43].

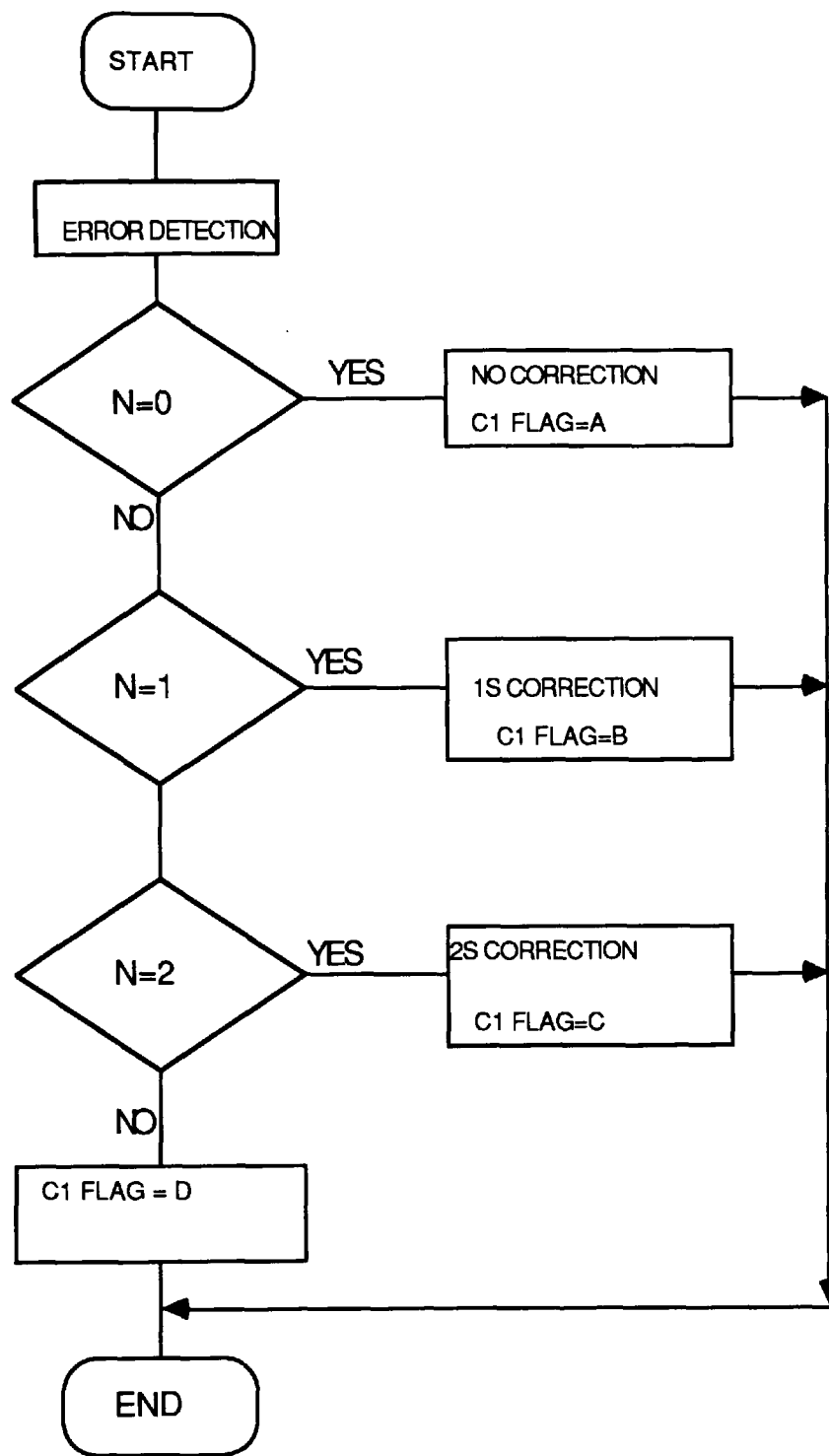


Figure 4.5 Flow Chart for C1 Decoding

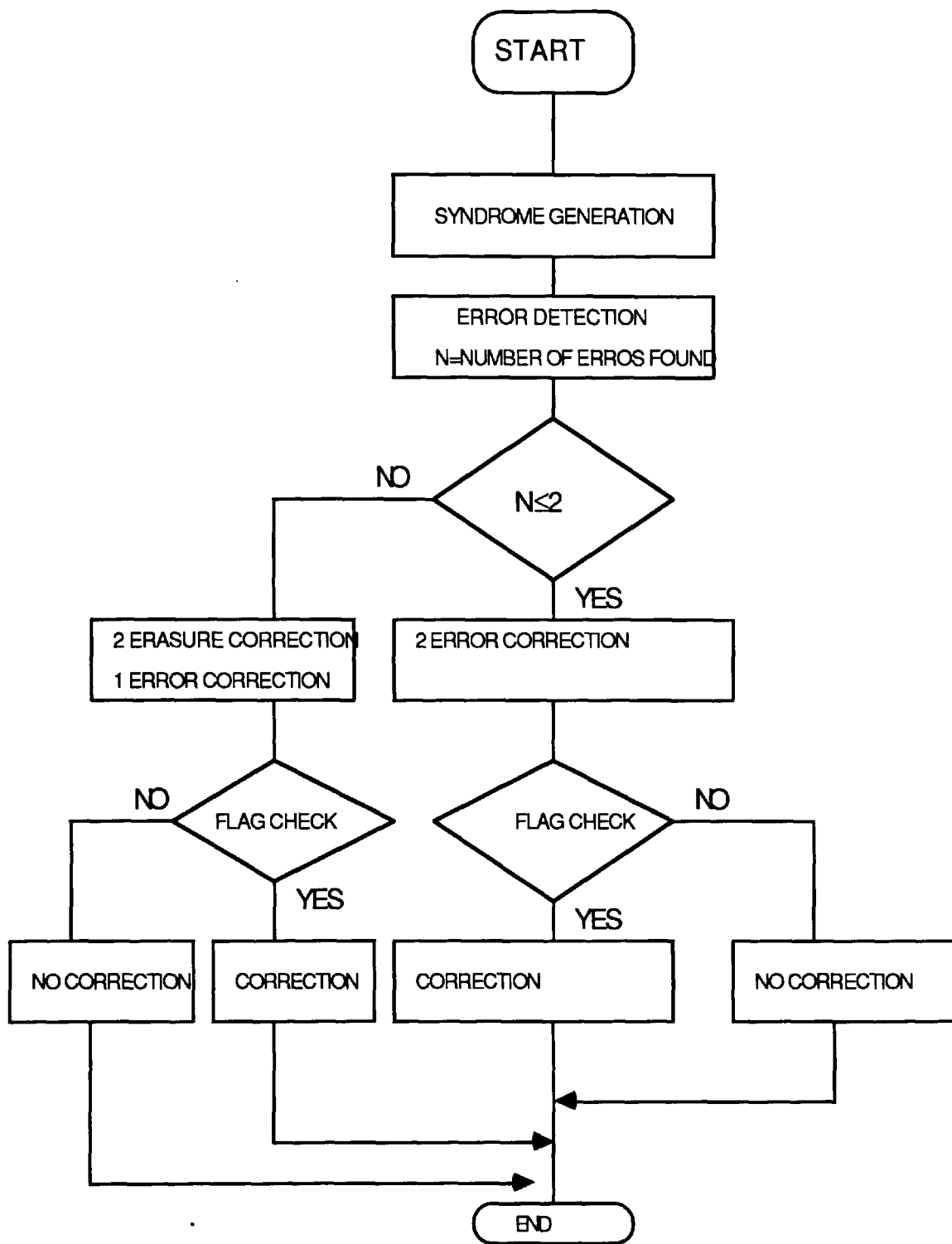


Figure 4.6 Flow Chart for C2 Decoding

REFERENCES

- 4.1) **R.Hill** "Error Correcting Codes", Chapter 1, Oxford Applied Mathematics and Computing Science Series Clarendon Press Oxford 1986
- 4.2) **R.R.Varashamov** " Estimate of the Number of Signals in Error Correcting Codes"Doklady A.N.S.S.S.R. 117 No.5 pp 739-741
- 4.3) **T.R.N.Rao and E.Fujiwara** "Error Control Coding For Computer Systems", Chapter 6 "Codes for Mass Memories" McGraw-Hill International 1991 pp 299-348 1990
- 4.4) **A.M.Patel and S.J.Hong** "Optimal Rectangular Code for High Density Magnetic Tapes" IBM Journal of Research and Development Vol.18 (Nov 1974) pp 579-588
- 4.5) **A.M.Patel** "Adaptive Cross Parity (AXP) Code for High Density Magnetic Tapes" IBM Journal of Research and Development Vol. 29 (Nov 1985) pp 546-562
- 4.6) **R.R.Varashamov and G.M. TenegoI'ts** "Correction Code for single asymmetric Errors" Avtomatikai Telemekhanika Vol.26 No. 6
- 4.7) **S.D.Constantin and T.R.N.Rao**"On the Theory of Binary Asymmetric Error Correcting Codes" Information and Control Vol. 40 1979 pp 20-46
- 4.8) **R.Hill** "Error Correcting Codes", Chapter 2, Oxford Applied Mathematics and Computing Science Series Clarendon Press Oxford 1986
- 4.9) **J.Watkinson** "Coding for Digital Audio" Chapter 4 pp 79-140 Focal Press 1990
- 4.10) **P Sweeney** "Error Control Coding", Chapter 2 Prentice Hall International 1991 pp 28-31
- 4.11) **W.W. Peterson and E.J. Weldon** "Error Correcting Codes" Chapter 5 , MIT Press 1961 pp 117-120
- 4.12) **R.W. Hamming** "Error Detecting and Error Correcting Codes", Bell System technical Journal Apr 1950 pp 147-160
- 4.13) **R.Hill** "Error Correcting Codes", Chapter 3, Oxford Applied Mathematics and Computing Science Series Clarendon Press Oxford 1986
- 4.14) **R.Hill** "Error Correcting Codes", Chapter 1, Oxford Applied Mathematics and Computing Science Series Clarendon Press Oxford 1986
- 4.15) **W.W. Peterson and E.J. Weldon** "Error Correcting Codes" Chapter 9 , MIT Press 1961 pp 269-310
- 4.16) **I.S. Reed** "A Class of Multiple Error Correcting Codes and the Decoding Scheme" IRE Trans. PGIT-4 1954 pp 38-49

- 4.17) **R.C. Bose and D.K.Ray-Chaudhuri** "On a Class of Error Correcting Binary Group Codes", Information and control Vol-3 1960 pp 68-79
- 4.18) **J.Adaamak** "Foundations of coding" Chapter 9 " Reed Muller Codes: Weak Codes with Easy Decoding" J. Wiley and Sons 1991 pp 137-159
- 4.19) **B Middleton and T.Jack-Kee** "Dropouts and their Effects on Error Rates in Digital Magnetic Tape Recording System"IRE Conference on Video and Data Recording Southampton Apr 1982 pp 43--49
- 4.20) **W.W. Peterson and E.J. Weldon** "Error Correcting Codes" Chapter 8 MIT Press 1961 pp 206-268,
- 4.21) **W.W. Peterson and E.J. Weldon** "Error Correcting Codes" Appendix C2 MIT Press 1961 pp 476-492,
- 4.22) **P Sweeney** "Error Control Coding", chapter 4 Section 4.23 pp 105 Prentice Hall International 1991
- 4.23) **I.S. Reed and G. Solomon** "Polynomial Codes over Certain Finite Fields" Journal Society Industrial Applications Maths Vol-8 No. 2 June 1960 pp 300-304
- 4.24) **J-P. Boly and W.J. van Gils** "Codes for combined Symbol and Digit Error Control" IEEE Trans. on Information Theory Vol. IT-34 No. 5 Sept 1988 pp 1286-1307
- 4.25) **P. Sweeney** "Error Control Coding" , Chapter 6 Section 6.4 pp 136-138 Prentice Hall International 1991
- 4.26) **P Fire** "A Class of Multiple error correcting codes for Non-Independent errors" Sylvania Reconnaissance Lab report RSL-E-2 1959
- 4.27) **J.Watkinson**" Coding for Digital Audio" Chapter 4 pp 79-140 Focal Press 1990
- 4.28) **D.C. Bossen** "B-adjacent Error Correction" IBM J-Res Dev 14 pp 402-408 1970
- 4.29) **J.Watkinson** "Coding for Digital Audio" Chapter 4 pp 79-140 Focal Press 1990
- 4.30) **P. Sweeney** "Error Control Coding" , Chapter 4 Section 4.22 pp 101-104 Prentice Hall International 1991
- 4.31) **K.Hayashi, T.Arai, T.Noguchi, H.Okamoto and M.Kobayashi** "Error correction for R-Dat and its evaluation". IEEE ICASSP 1.3 Tokyo1986
- 4.32) **T.Arai, K.Okamoto, M.Kobayashi and T.Takeuchi** "High Capability Error Correction LSI for CD-Player and CD-ROM" IEEE Trans. on Consumer Electronics Vol CE-30 N0.3 August 1984 pp 353-359
- 4.33) **L.B. Vries and K.Odaka** "CIRC The Error Correcting code for the Compact Disc Digital Audio System" 67th AES Convention Oct 1980 pp 178-186
- 4.34) **P Sweeney** "Error Control Coding" , Chapter 5 pp 110-133 Prentice Hall International 1991

- 4.35) **G.C.Clarke and J.B. Cain** "Error Correction Coding for Digital Communications" Chapter 6 "Convolutional Code Structure and Viterbi Decoding" pp 227-267 1981 Plenum Press
- 4.36) **K.S. Shamugan** "Digital and Audio Communication Systems" Section 9.6 "Convolutional Codes" pp 478-486 J.Wiley and Sons 1979
- 4.37) **G.P. Beaumont** "Intermediate Level Statistics" Chapman and Hall 1980
- 4.38) **J.Watkinson** "Coding for Digital Audio" Chapter 4 pp 79-140 Focal Press 1990
- 4.39) **J.Watkinson** "The Art of Digital Audio" Section 7.15 pp 241 Focal Press 1988
- 4.40) **B.C. Mortimer M.J. Moore and M.Sablatash** "The Design of a high performance error-correcting Coding Scheme for the Canadian Broadcast Teledion System Based on Reed-Solomon codes." IEEE Transactions on Communications Vol COM-35 No.11 Nov 1987 pp 1113- 1123
- 4.41) **L-N Lee** "Concatenated coding system employing a unit memory convolutional code and a Byte Orientated decoding Algorithm" IEEE Trans. on Communications Vol. Com-25 No. 10 Oct 1977 pp 1064-1074
- 4.42) **J. Justensen C. Thomassen and V.V. Zylabov** " Concatenated codes with Convolutional Inner Codes" IEEE Trans. on Information Theory Vol. 34 No. 5 Sept 1988 pp 1217-1225
- 4.43) **W.W. Peterson and E.J. Weldon** "Error Correcting Codes" Section 5.10 "Concanted codes", MIT Press 1961 pp139-141

Chapter 5

Modulation Codes

5.1 Introduction

5.1:1 Introduction

The purpose of this chapter is to describe and demonstrate the various types of Modulation code, used in data storage, and to illustrate their specifications.

A Modulation Code is the method by which a stream of binary data digits is converted into a digital waveform which is then recorded on to the medium. In the digital waveform a one represents the presence of a transition in a timing window and a zero that no transition is present. It has been shown in Chapter 2 that Modulation conventionally occurs after the error correction bits have been added to the data.

One of the main assumptions in Chapter 2 was data is a stationary random variable so that the replay pulses would be independent along the tape. However this does not account for the block structures imposed in most recordings; these are added to the data to simplify the error correction and replay processes.

The first section of this chapter deals with the background of Modulation codes, the reasons why they are used in digital data recording and the parameters over which modulation codes are defined. The main groups of modulation codes are then discussed with examples of these types of code.

The two modulation coding schemes, Manchester and Miller, whose use in digital data recording are examined in the latter parts of this thesis are then introduced. Further codes common in both fixed head, rotary head and thin film data recording are then discussed, including those block codes which are used in high integrity digital data recording of R-DAT and Optical Disc.

In the final Section of this chapter several codes which combine Modulation with error correction will be discussed. These use the redundancy inherent in Modulation coding to trap and correct errors.

5.2 Modulation Coding Principles

5.2:1 The Rationale of Modulation Coding[5.1,5.2]

The purposes of Modulation coding are:

- 1) the reduction of the amount of DC in the Recorded signal. This is done by the eradication of the repetition of the same Modulation bit for a whole Series of 1's or 0's. Codes which have a high amount of DC are prone to clockloss errors such as dropout and jitter;
- 2) the use of Modulation codes may also help in the error detection by signalling a code violation for certain errors. This is due to the fact that the run length limited codes (RLL) have limits on the permitted number of consecutive 0's between each pair of 1's. This will then permit the detection of some errors in the recording process. It is important to note that in a RLL the normal use of

1's and 0's to represent data is suspended. A 1 represents a transition and a 0 an empty timing window;

3) certain types of modulation coding also permit the system to reclock automatically. For example in the Manchester code there must be a transition in the middle of each data cell, where each data cell consists of two modulation bits;

4) to allow the Power Spectral Density of the signal to be matched to the frequency response of the channel, as the maximum frequency of the signal can be increased to fill the channel bandwidth so not allowing any wideband noise to dominate the signal;

5) to combat pulse crowding. In high density recording the error mechanism ISI becomes a major problem. The choice of modulation code can then be made to combat pulse crowding. Those codes which have a high code rate, approaching unity for certain group codes, avoid the close packing of transitions. Due to this reduction in the packing density the group codes will be far less prone to ISI.

5.2:2 Parameters for Modulation Coding[5.3]

To be able to compare modulation codes it is necessary to have a set of parameters over which these codes are defined. These parameters apply to all modulation codes from Manchester modulation to the Group codes used in modern high integrity media. These are given below:

1) the Window Margin. The tolerance of the transition to bitshift is given by the window margin, T_w . The larger this value of T_w then the better the resistance to bitshift;

2) the Minimum Transition Distance. T_{min} this relates to the fundamental wavelength λ_{min} that is possible to record at a fixed tape speed. This is one of the major factors which determine the packing density. In addition spacing loss which is caused by dirt on the tape or poor tape quality is a function of λ_{min} and hence T_{min} is related to the reliability of the system;

3) the maximum inter transition distance T_{max} . To allow for self clocking, the phase should be correctable at the point of transition. In order to facilitate this T_{max} should be finite and as small as possible;

4) clock recovery. After a long burst error or at the start of the tape the decoding circuit must be reset on to the correct clock pulse from the tape as quickly as possible. This is known as clock recovery and requires the following parameters:

- i) the value of T_{max} or the ratio of T_{max}/T_{min} to be as small as possible;
- ii) a high probability of a transition in a timing window is desirable;

5) DC or Low Frequency Content. DC free code is essential in rotary head recorders as these cannot pass the DC component. In addition DC free codes

have other beneficial characteristics:

- i) simpler decoding circuits;
- ii) a high pass filter can now be introduced into the replay electronics to reduce the sensitivity of the media to asperities;
- iii) the low frequency portion of the channel band can be utilised for different purposes, such as servo control;

6) The Constraint Length. The constrained length L_C , is the length of prior Modulation bits which effect the present Modulation bit. The chance of error propagation is proportional to L_C .

5.3 Types of Modulation Code

There are many different algorithms for Modulation coding, these fall into four main types:

Run Length Limited Codes;

Non Return To Zero Codes;

Group Codes;

These code types will now be considered together with an individual treatment of some of the simpler codes of each type.

5.3:1 Run Length Limited Codes[5.4, 5.5, 5.6, 5.7, 5.8, 5.9]

RLL codes have set limits for the amount of allowable DC between each consecutive pair of pulses. To find the length of these runs of zeros the modulated signal is divided into a series of sections with each section containing a single pulse. The shortest of these sections is of length $d+1$ and the longest of length $k+1$ [5.8]. This is illustrated by the example below:

000101001001010100101;

0001 01 001 01 01 01 001 01;

where $k+1=4$ and the $d+1=2$.

These codes are denoted by the set of variables which control the constraints of the code $(d,k;m;n;r)$ these codes have;

d is the minimum number of pulse free modulation bits between each pair of pulses, these are used to prevent pulse crowding;

k is the maximum number of pulse free modulation bits between each pair of pulses, these provide a means of self-clocking;

m is the number of data bits;

n is the number of code bits, where $n>m$;

$r=1$ implies a fixed length code;

$r>1$ implies a variable length and fixed pulse code.

The manner and the degree whereby these constraints on the code effect the performance of the modulation code against errors can be shown by:

Code Rate, $R=m/n$. The higher the Code Rate the better the resistance to timing jitter;

Minimum Transition Interval $M= (m/n)(d+1)$. High Minimum Transition Intervals lead to a high read back SNR;

Pulse Gap Ratio $P=(k+1)/(d+1)$. Codes with a high Pulse Gap Ratio have high shift, poor self clocking and a large DC component.

Therefore the ideal RLL code for high linear density and good data reliability will have high R , high M , and low P . A conflict of interests arises due to the fact that M and P are proportional to each other.

RLL codes have an error detecting ability. This is due to the selection of a codeword of length n , to represent a data set of length m . Thus 2^m words will make up the constrained codewords from the original unconstrained set of 2^n . This leads to a code rate greater than one. This code rate will give redundancy since if $n>m$ there will be a considerably greater number of unconstrained codewords of length n than the original set of 2^m constrained codewords. So therefore if an error breaks the runlength constraints of the code it will be detected as it is not a member of the set of allowable constrained codewords.

The Asymptotic Information Rate or Capacity of a channel, using fixed length codes is defined by Shannon [5.7] in terms of the entropy. For a set of symbols X with possible values $\{x_1, x_2, \dots, x_n\}$ and associated probabilities $\{p_1, p_2, \dots, p_n\}$; the Entropy is defined as:

$$H(p_1, \dots, p_n) = -\sum_{j=1}^n p_j \log_2 p_j$$

This gives the capacity of a memoryless channel. However RLL codes can be modelled as a Markov process as they require information about the previous bits to obey the runlength constraints. This gives the following maximum capacities for the runlength constraints:

k	d=0	d=1	d=2	d=3	d=4	d=5	d=6
1	.6942						
2	.8791	.4057					
3	.9468	.5515	.2878				
4	.9752	.6174	.4057	.2232			
5	.9881	.6509	.4650	.3218	.1823		
6	.9971	.6690	.4979	.3746	.2669	.1542	
7	.9986	.6793	.5174	.4057	.3142	.2281	.1335
8	.9993	.6853	.5293	.4251	.3432	.2709	.1993
9	.9996	.6888	.5369	.4376	.3620	.2979	.2382

Table 5.1 Maximum capacity of Run Length Limited Sequences[5.9]

The Power Density Spectra of RLL codes have been investigated [5.10] and the results for several of the more common RLL codes used in digital recording are given, these include the Miller code and the (2,7) code used by IBM and Xerox.

5.3:2 Non Return to Zero Codes (N.R.Z.)

The premise of all N.R.Z. codes is that the code will never stay at zero. The two most common types of N.R.Z. code are the standard N.R.Z. and the modified N.R.Z.1. [5.11] whose algorithms for construction are given below.

(a) NRZ: Change direction of saturation at mid bit time only if the present bit is not equivalent to the previous one;

(b) NRZ1: Change mid bit only if the bit is a 1.

These Modulation codes do very little to remove the inherent DC from the encoded message. Compared to other more complex forms of NRZ Modulation such as Miller and Manchester.

5.3:3 Group Codes[5.11, 5.12]

Group Modulation Codes have been in existence for many years. A method of generating these codes was first developed by in 1970[5.11]. All codes which do not use a single type of transition to represent a particular data bit throughout whole length of the code can be regarded as group codes. However within this thesis a tighter definition of the term 'Group Codes' is used:

a Group Code is a modulation code where the encoding algorithm regards the data not as an infinite stream of individual data bits but as being grouped into a series of data

vectors. Each data vector is of equal finite length. These data vectors are then mapped onto other vectors of modulated transitions all, of which are also of equal finite length

Group Codes are becoming increasingly important and are used in both R-DAT and Optical Disk. The first of these mediums uses an 8/10 Modulation code, taking 8 data bits to 10 modulated bits. The second uses Eight to Fourteen Modulation (EFM) which takes 8 data bits to 14 modulation bits,[5.12]. Both mediums use Group Codes with Reed Solomon Error Correction over $GF(2^8)$ where the data is divided into sub-words, of length 8 before encoding.

Two examples of simple group coding algorithms are given below. These are the Rice Code and the Octal Coded Binary. These examples cover the encoding algorithms and the modulated waveforms of the two codes.

5.3.3 a) Rice Code[5.11]

The first six bits of the binary data sequence are initially viewed:

if these first six bits are the sequence 010101. Then the system encodes these bits as shown in Figure 5.1. Otherwise the last two bits are returned to the data bitstream;

if the remaining bits commence with 11. Then all four are encoded as shown in Figure 5.1. Otherwise the last two bits are returned to the data bitstream;

the remaining two bits are encoded as shown in Figure 5. 1.

This system is repeated with the next six bits in the data stream, including any bits which were returned earlier.

Figure 5.1 shows all the possible recording wave forms for this system;

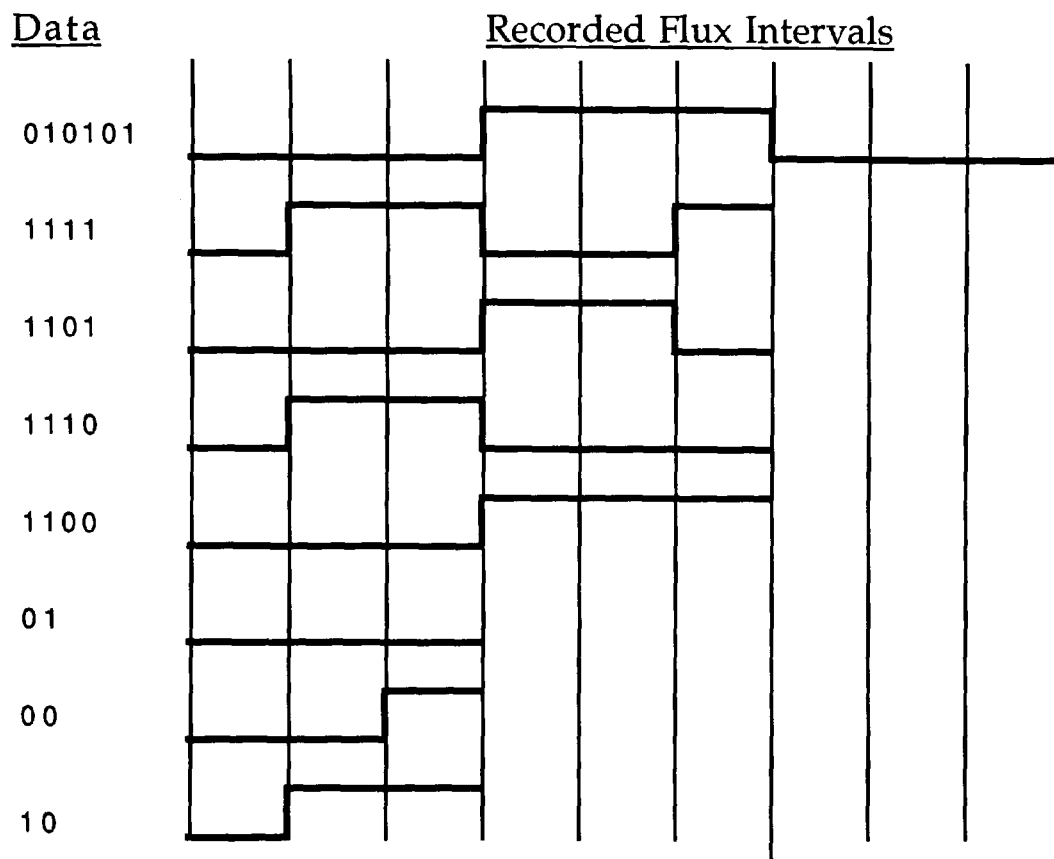


Figure 5.1 Rice Encoding Diagram[5.11]

This code is of considerable interest in computer data storage. It provides a performance comparable to NRZ1 code, while allowing some self clocking. However the encoding strategy is comparatively complex to the non-group codes such as the NRZ1.

5.3:3b)Octal Coded Binary [5.11]

In this code the data bits are coded in groups of three. The possible modulated waveform sets are shown in the Figure 5.2 .

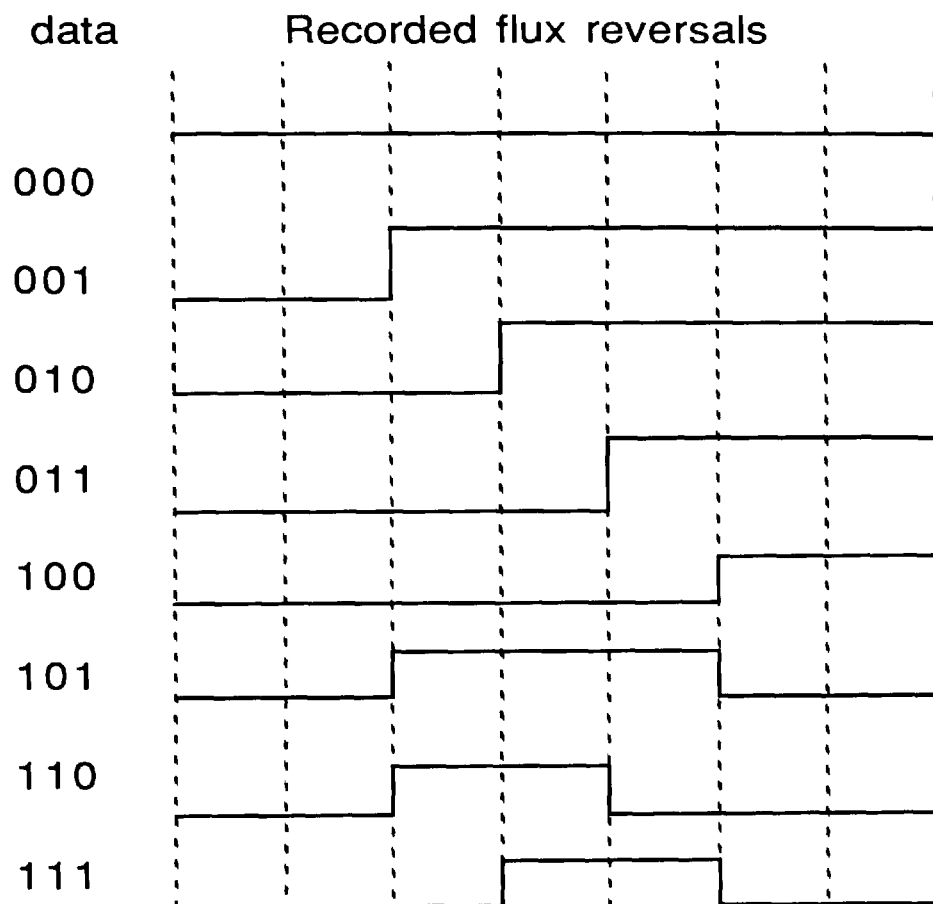


Figure 5.2 Octal coded Binary Modulation.

5.4 Modulation Codes Considered in this Thesis

5.4:1 Manchester Modulation or FM [5.11, 5.1]

The Manchester code transmits:

a received 1 as positive pulse for the first half of the data bit cell and a negative pulse for the second half;

the 0 is conveyed as the same pulse sequence but with reverse polarity.

As there is at least one transition in each data bit cell the receiver clock timing can be extracted from the waveform in the presence of long strings of data 0's and 1's. The bandwidth required for Manchester Modulation is twice that of a standard bipolar signal[5.11]. The Manchester Modulation forms a (0,1) RLL code. A Section of binary data with m code bits will have a maximum of $2m$ modulated transitions.

The RLL properties of the Manchester code are therefore:

Code Rate; $R=m/n = 1/2$. This is a low code rate so this code will be perceptible to timing jitter;

Minimum Transition Interval $M=(m/n)(d+1)= 1/2$. This low minimum transition interval will lead to a low read back SNR;

Pulse Gap Ratio $P=(k+1)/(d+1)=2$. This is relatively low so will lead to a high

self clocking ability and a small DC component.

5.4:2 The Miller Code or MFM [5.11.5.13]

The algorithm for this code is comparatively simple:

a transition occurs at the mid data bit time of the 1;

a transition occurs at the end of a data bit time between consecutive 0's .

The attractions of Miller coding for magnetic recording are that:

it has a small power for frequencies near DC;

it needs approximately half the bandwidth used for Manchester coding;

the code is simple to implement as shown above;

in addition it is less susceptible to the problems of ISI as an error mechanism;

in data recording, this is due to the increase in the minimum distance between pulses.

Miller code can also be regarded as a (1,3) RLL code which maps a data bit onto two Modulation bits. This allows some method of comparison between the Manchester and Miller Modulation codes:

Code rate $R=m/n=1/2$:

Minimum Transfer interval $M=(m/n)(d+1)=1.5$;

$P=(k+1)/(d+1) = 5/3$ ratio.

Comparing the figures above for the Miller code to those of the Manchester code given in 5.4:1 the Miller code has the same rate as the Manchester code with an improved Minimum Transfer Interval and Pulse Gap Ratio.

5.5 Enhancements to the Miller code

5.5:1 Miller² code or M²FM[5.14]

This code was devised by J.W. Miller[5.14] as an alternative to the original Miller code. The Miller² code was designed in such a way as to remove the DC content. DC leads to a low minimum signal rate which tends to manifest itself as a baseline shift in either the read back waveform or its derivative.

Coding rules for Miller² coding

The data stream is broken into sequences of data of three possible types:

- (a) any number of consecutive 1's;
- (b) a pair of 0's separated by either no ones or an odd number of 1's;
- (c) a single 0 followed by an even number of 1's (terminated by a 0 which is not counted as part of the sequence).

The sequence types (a) and (b) are encoded in the same way as the normal Miller code while those sequences of type (c) have the transition for the final 1 inhibited. This can be shown in Figure 5.3.

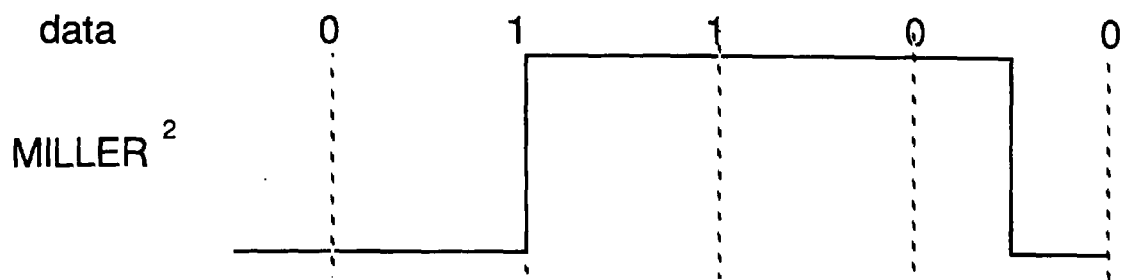


Figure 5.3 Miller² Encoding

The coding rules for Miller² coding is given by:

- a transition at the mid bit point of a 1;
- a transition at the end bit between consecutive 0's;
- suppression of the transition for the last of any even number of 1's.

The runlength constraints of the Miller² code are (2,5) giving the following set of results:

Code rate $R=m/n=1/2$;

Minimum Transfer Interval $M=(m/n)(d+1)= 1.5$;

Pulse Gap ratio $P=(k+1)/(d+1)= 6/3$.

The Miller² code combines the Minimum Transfer Interval of the Miller code with the superior Pulse Gap Ratio of the Manchester code.

5.5:2 Zero Modulation[5.15]

The third member of the Miller family of codes is Zero Modulation. The Code Rate of Zero Modulation is $1/2$. The code has a minimum runlength, $d = 2$ and a maximum runlength, $k = 4$. In addition to the runlength constraints Zero Modulation also considers the accumulated charge on any digit. This accrued charge is bounded by $\pm c$ units. For the Zero Modulation code this limit is set at ± 3 units. The coding rules for the encoding and decoding algorithms for Zero Modulation are given below:

The Algorithm requires the following Notation

d_0 is the data bit to be encoded;

a_0b_0 are the ZM digits to which the bit is recorded;

$a_{-1}b_{-1}$ are the ZM digits to which the previous bit is recorded;

a_1b_1 are the ZM digits to which the following bit is recorded;

$d_{-1} d_{+1}$ are the adjacent data bits, both with respective ZM digits;

$P(A)$ is the look-ahead one-sequence parity bit. This is the modulo-2 count of ones in the binary sequence up to the next zero;

$P(B)$ is the look-back one sequence parity bit. This is the accumulated modulo-2 count of all zeros from the start of the data up to and including d_0 .

To encode the data there are a set of conditions which must be followed:

Encoding Algorithm

Data	Modulation	
d_0	$a_0 b_0$	Condition
0	10	$d_{-1}=0$ or $d_{-1} = 1$ and $a_{-1} b_{-1} = 00$
0	00	$d_{-1} = 1$ and $a_{-1} b_{-1} \neq 00$
1	10	$d_{-1}=0$ and $P(A) = 0$ or $d_{-1} = 1$ and $a_{-1}b_{-1} = 00$
1	00	$d_{-1} = 1$ and $a_{-1}b_{-1} = 10$
1	01	Otherwise

Decoding Algorithm

Modulation	Data	
$a_0 b_0$	d_0	Condition
10	1	$a_1b_1 = 00$
10	0	$a_1b_1 \neq 00$
00	1	$a_{-1}b_{-1} = 01$
00	0	$a_{-1}b_{-1} \neq 01$
01	1	none

Due to the complexity of this Modulation code which uses both look-ahead and look-back algorithms and a parity check bit for codes with only a limited memory, Zero Modulation possesses a powerful check capability for bit error detection, in addition to the ability to synchronise the code after drop-outs.

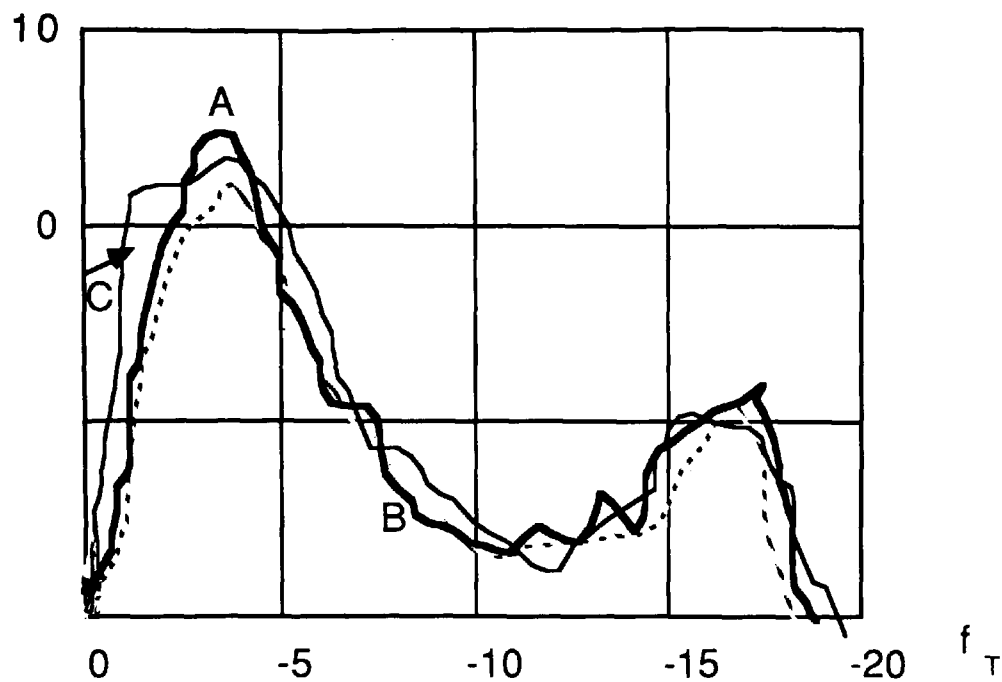
5.5:3 Comparison of the Miller Codes

The power spectral densities of the three Miller type Modulation codes were plotted against frequency[5.16]. These plots are given below in figure 5.4 in which the different codes are represented by a different line.

Code Type	d	k	R	M	P
Manchester Modulation	0	1	0.5	1	2
Miller Modulation	2	4	0.5	1.5	5/3
Miller ² Modulation	2	5	0.5	1.5	2
Zero Modulation	2	4	0.5	1.5	5/3

Table 5.2 Comparison of the Runlength Constraints of the Miller Type Codes

As can be seen from Figure 5.4. The power spectral densities of the three different Miller type codes are remarkably similar. Indeed the Power Spectral Density of the three principal codes in the Miller family all have the same primary defect of non-zero energy at DC.



A= Miller code (bold line)

B= Zero Modulation (dotted line)

C=Miller² code (plain line)

Figure 5.4 PSD of three Miller like codes (5.18)[5.16]

5.6 Modulation Code Used in Fixed Head Recording[5.17]

5.6:1 Codes in Fixed Head Recording

There are several different modulation codes used in fixed head recorders. The choice of code is dependent upon the desired data rate and the runlength coefficients required. Zero Modulation and Miller² which were discussed in Section 5.5:2 are widely used. In addition to these codes another member of the Miller family is used, this is a modified form of Zero Modulation[5.18] which has runlength constraints (1,4) in place of (2,4) of unmodified ZM.

5.6:2 The 3PM code[5.19]

This code is used in the IBM high-performance floppy disc files. The runlength constraints of this code are (2,7), which allow a better minimum spacing of three times the fundamental distance compared with two for Miller code. The Code Rate is maintained at 1/2, hence the fundamental distance is half the length of a data bit cell. The recorded transition density is lower than the Miller code by a factor of 1.5. The code uses variable length blocks to encode the data. It is similar in that respect to the examples given for block coding in Section 5.3;3.

As can be seen from the table below, encoding is undertaken by partitioning the data into blocks of two, three or four bits. To decode, the process is reversed and the data is divided into blocks of 4, 6 or 8 bits. Alternatively, a decoding algorithm is also available in which the error propagation is limited by the use of a preamble and a postamble of a known number of bits.

Data	Code Word
10	0100
11	1000
000	000100
010	100100
011	001000
0010	00100100
0011	00001000

Table 5.3 Encoding Scheme for (2,7) Code

5.6:3 Jacoby and Kost's (1,7) code [5.20]

The Code Rate $R=2/3$ in this Modulation code, it has improved properties compared with the Miller and 3PM codes including:

an increase of 33% in the data rate, the Minimum Transition and the Fundamental Distance of recording compared with the Miller Modulation code;

compared with the 3PM code an increase in the Fundamental Distance of 33%. However T_{\min} is reduced by 11% and the ratio of T_{\max} / T_{\min} has been increased by 50%.

The code can be represented either as a fixed length block code, which takes data words of length 2 bits, mapping these onto Modulation codevectors of length 3, with at least a single transition but no consecutive transitions. However this method of encoding leads to conflict over the runlength constraints, as code words which end with a transition may be followed by codewords which begin with a transition. This can be shown in Table 5.4:

Serial Number	Data Word	Code Word
0	0 0	1 0 1
1	0 1	1 0 0
2	1 0	0 0 1
3	1 1	0 1 0

Table 5.4 Basic Encoding Table for (2/3) Code

The Serial Number in Table 5.4 refers to the decimal representation of the code. Taking the data words in pairs derive an alphabet of 16 words of length 4 as shown in table 5.5.

Serial Number	DATA BITS	CODE BITS
0 0	0 0 0 0	1 0 1 1 0 1
0 1	0 0 0 1	1 0 1 1 0 0
0 2	0 0 1 0	1 0 1 0 0 1
0 3	0 0 1 1	1 0 1 0 1 0
1 0	0 1 0 0	1 0 0 1 0 1
1 1	0 1 0 1	1 0 0 1 0 0
1 2	0 1 1 0	1 0 0 0 0 1
1 3	0 1 1 1	1 0 0 0 1 0
2 0	1 0 0 0	0 0 1 1 0 1
2 1	1 0 0 1	0 0 1 1 0 0
2 2	1 0 1 0	0 0 1 0 0 1
2 3	1 0 1 1	0 0 1 0 1 0
3 0	1 1 0 0	0 1 0 1 0 1
3 1	1 1 0 1	0 1 0 1 0 0
3 2	1 1 1 0	0 1 0 0 0 1
3 3	1 1 1 1	0 1 0 0 1 0

Table 5.5 Encoding table for two data words

However it can be seen from this table that the codewords represented by the Serial Numbers 00,01,20 and 2,1, violate the runlength constraints and so must be modified to satisfy these bounds. The altered codewords then give the following set of codewords.

Serial Number	DATA WORD	CODE WORD
0 0	0 0 0 0	1 0 1 0 0 0
0 1	0 0 0 1	1 0 0 0 0 0
2 0	1 0 0 0	0 0 1 0 0 0
2 1	1 0 0 1	0 1 0 0 0 0

Table 5.6 Additional coding table.

The merged data words in Table 5.6 lead to a complex decoder requiring full word look ahead[5.20].

5.7 Modulation Codes in High Integrity Recording

5.7:1 The Advantages of Group Codes for High Integrity Media

The principle application of group modulation codes is in high integrity media. Both the optical disc and the R-DAT tape use a group code based on eight data bits to accommodate the Reed Solomon coding over $GF(2^8)$.

The two types of Modulation code are very similar as both take a block of 8 data bits from the error correction code and map it on to a longer set of transitions. This maximises the ability of the Reed Solomon code to correct an entire data subword. Such high code rates (0.8 for the 8/10 Modulation code) will result in a the entire block being corrupted by a single shift error in the output prior to demodulation. However this type of error can be fully compensated by the Reed Solomon code.

EFM was devised before 8/10 Modulation but the operating mode of both codes are similar as both are high density block codes. Accordingly this Section concentrates upon EFM, even though it is used in optical rather than magnetic recording.

5.7:2 EFM the Modulation Code in Optical Recording [5.21, 5.22, 5.23]

Modulation codes are also used in optical discs which were discussed in Section 3.5. In the recent past Modulation codes have been applied more widely to optical recording. Digital magnetic fixed head recorders use the 3PM (2,7) code[5.19] while the optical disc uses an EFM code. The EFM code was designed to fulfil the following criteria:

(i) low probability of readout error at high information densities;

(ii) resistance to imperfect optics, caused by either defocussing or disc skewing about the optical axis. These will cause degradation in the phase and amplitude characteristics, so the Modulation system should have low sensitivity to tolerances along the optical light path. This implies a Modulation code with a high minimum distance T_{\min} ;

(iii) good clocking characteristics. The bit clock is used as a method of synchronisation for the motor control of the player and the digital data. This is generated by the readout signal from the data pit edges. The signal must therefore attempt to

minimise T_{\max} the maximum distance between transitions;

(iv) high resistance to dirt and scratches on the surface of the cover of the disc. These cause a change in the envelope of the readout signal which produces low frequency-noise. These can be removed by digital filtering provided that there is no significant low-frequency component of the signal. In addition the low frequency component leads to interference in the servo system causing instabilities;

(v) good resistance to error propagation. This is achieved by mapping the demodulated block of eight bits onto a subword of Cross Interleaved Reed Solomon Code, used in the error correction system.

EFM is a group code based on a block of 8 data bits. It has a minimum distance of 3 channel bits ($=1.5$ data bits) with a sampling window of 1 channel bit ($=0.5$ data bit), and a maximum distance of 11 channel bits.

There are 267 14 bit patterns which satisfy the two limits on the run length $T_{\min} = 3$ and $T_{\max} = 11$ [5.21]. As only 256 are required for the eight binary bits from the Reed Solomon code so the additional 11 are discarded. Those discarded exhibit the worst DC characteristics.

Table 5.7 shows part of the conversion between the eight bit data block and the modulated group of eleven.

Decimal Word	Data Word	Modulation Word
100	01100100	01000100100010
101	01100101	00000000100010
102	01100110	01000000100100
103	01100111	00100100100010
104	01101000	01001001000010
105	01101001	10000001000010
106	01101010	10010001000010
107	01101011	10001001000010
108	01101100	01000001000010
109	01101101	00000010000010
110	01101110	00010001000010

Table 5.7 EFM Modulation

To prevent a violation of the constraint on $T_{\min}=3$ at least 2 merging bits must be added between modulated codewords. Though to allow flexibility three merging bits are added. The merging bits used are either 000, 100, 010, or 001. The last three non-zero codes are shown below as cases A, B and C respectively. The use of these merging bits in EFM can be shown in Figure 5.5.

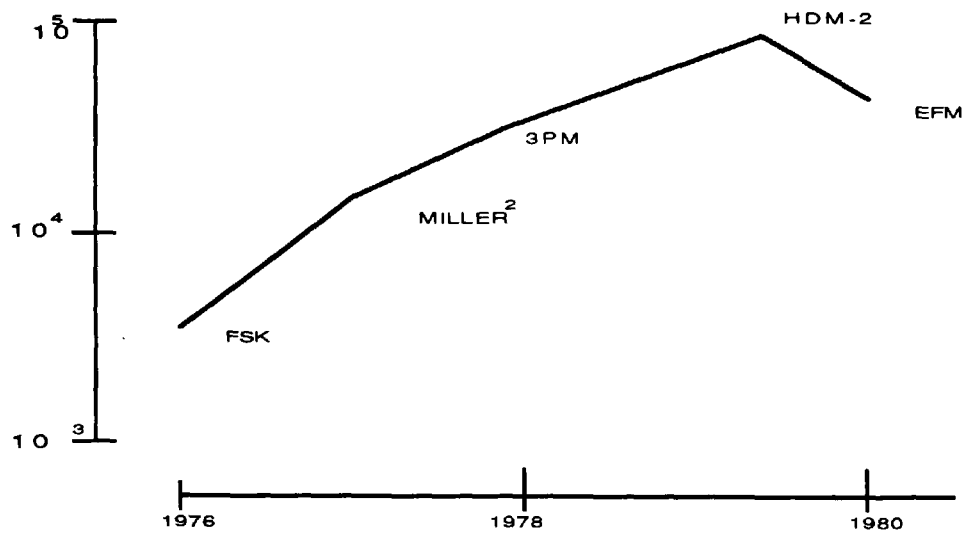


Figure 5.6 Packing density for Modulation codes in optical recording.[5.3]

5.7:2 Alternative Modulation for Optical Disc.[5.24, 5.30, 5.31]

Although EFM is now the standard modulation code used in optical recording media, there are many other codes which satisfy the (2,10) RLL characteristics of EFM while maintaining better data rates. These in the main are different types of block sliding code, the table below presents the results of the experiments undertaken by French and Wolf.

(d,k)	Capacity	m/n	DR	Encoder States	Splitting States	Comments
(2,10)	0.5418	8/17	1.41	1	n/a	EFM
(2,7)	0.5174	2/4, 3/6, 4/8	1.50	1	n/a	Variable Length
(2,7)	0.5174	1/2	1.50	7	3	Sliding Block
(2,8)	0.5283	1/2	1.50	5	3	Sliding Block
(2,9)	0.5418	2/4	1.50	3	1	Sliding Block
(2,10)	0.5418	4/8	1.50	3	1	Sliding Block

Table 5.8 Comparison of Modulation codes used in Optical Recording [5.24]

5.7.3 8/10 Modulation code in R-DAT [5.25, 5.26]

The recording code used in the R-DAT recorders is an optimised 8/10 Modulation code. 8/10 codes were first used in digital video recording to give a DC-free code with a wide detection window. The code rate of 0.8 for the 8/10 Modulation is very high though it is not unique; since there exists a 4/5 Modulation code which is used in communications for Fibre Distributed Digital Interfaces, (FDDI)[5.27]. Indeed there are codes such as the 8/9 codes which have a higher code rate, though the maximum runlength for these codes is large, $11.6T$ to $18.7T$ [5.28], where T is the length of the data window.

The 8/10 Modulation code is a group code designed to fit with Reed Solomon error correction coding. Its design is particular to the priorities of the R-DAT recording system. These criteria are given below:

- (i) a large timing window. This is necessary due to the low Signal to Noise ratio caused by the narrow track pitch and crosstalk;
- (ii) the code should be DC free and possess only a small low frequency spectrum. This enables the recorder to eliminate low frequency crosstalk noise;
- (iii) the maximum transition, T_{\max} , should be as low as possible. In addition the ratio T_{\max}/T_{\min} should be as low as possible. These criteria allow easy erasure and cause less crosstalk noise.

placed in the memory until the next conversion.

The DC figure in this algorithm refers to the transitions shown in table 5.10. These DC codewords were used to minimise the Digital Sum Variation, DSV which is shown as +1 or -1 in Table 5.10.

	Q' = -1			Q' = 1		
Dataword	Codeword	DC	Q	Codeword	DC	Q
00010000	1101010010	0	1	1101010010	0	-1
00010001	0100010010	2	-1	1100010010	-2	-1
00010010	0101010010	0	-1	0101010010	0	1
00010011	0101110010	0	1	0101110010	0	-1
00010100	1101110001	2	1	0101110001	-2	1

Table 5.10 Part of the conversion table for 8/10 Modulation[6.26]

The Q is generated by :

$$Q = (Q' + DC) * (-1)^P$$

Where:

Q is the DSV for the present codeword;

Q' is the DSV for the previous codeword;

DC is the DC content of the codeword;

P is the parity of each bit of the codeword.

The optimal modulation code which fulfils all these requirements is an 8/10 Modulation code where $T_{\max} = 4T_{\min}$. This system gives a total of 271 employable codewords. This code can be compared with other codes discussed previously in this chapter in the Table 5.9.

T_w	T_{\min}	T_{\max}	Sync T_{\max}	DSV	Note
0.5T	0.5T	T	T	∞	Manchester
0.5T	T	2T	2.5T	∞	Miller
0.5T	T	2.5T	3T	3	Miller ²
0.8T	0.8T	2.4T	3.2T	∞	4/5 FDDI
0.8T	0.8T	4T	4.8T	3	8/10 type 1
0.8T	0.8T	3.2T	3.2T	3	8/10 type 1

Table 5.9 Parameters of some commonly used Modulation codes

To generate the 8/10 code it is necessary to follow the procedures outlined below. In the generation algorithm the DSV for the present codeword is referred to as Q and the DSV for the previous codeword is referred to as Q':

- i) a dataword is converted to a $Q'=-1$ codeword;
- ii) the DC content of the codeword is found, the codeword is then labelled as being either $DC=0$ or $DC=+2$;
- iii) the Q' which is generated from the previous codeword. This is then checked, if $Q'=+1$ and $DC=+2$ then the first bit of the codeword is then inverted;
- iv) Q is calculated from the DC content of the codeword and Q'. This is then

$a = (A + CZ + Y(C \oplus F (G + H)))$	
$b = (A(B + DE) + A(B + C))$	
$c = (AC + A(D + E) + BDE)$	
$d = (A(C + BDE) + CDE + CZ + (AB \oplus FGHY))$	
$e = ((AB + D)E + ABCDE + YF(G + H))$	
$f = ((AE(C + B \oplus D)) + ((D + CZ) \oplus F(G + H)))$	
$g = (FG + Y + (B + C))$	
$h = (FGH + FY)$	$Y = (A(B + C)DE)$
$i = (H + FG + FY)$	$Z = (ADEF(G + H))$
$j = FG + FY$	

Table 5.11 Conversion table for 8/10 Modulation[5.26]

In this conversion table the upper case letters represent bits in the data word and the lower case letters represent bits in the modulated word.

Although the 8/10 Modulation code seen above is incorporated into the R-DAT recording system, other 8/10 rate Modulation codes have been developed. There exist codes with better modulation constraints than that those of the codes derived above. A Generic Constraint Graph[5.29] has been developed to generate these codes. This graph generates a constrained (d,k,c) modulation code, where c is the upper limit on the DSV. This code has constraints given by $(0,3,5/2)$ this code minimises the product of the maximum runlength and the digital sum variance, given by kc , for an 8/10 rate modulation code.

5.8 Combined Error correction and Modulation Codes

5.8:1 Basic Principles of Combined Coding

In Chapter 2 it was stated that modulation codes did have an effect on the output error rate. However the facility of modulation codes to reduce the error rate is not maximised in a conventional recording system.

Codes may be devised which possess both the runlength qualities inherent in modulation coding and the Hamming distance properties of error correcting codes. This idea of combined E.C.C. and modulation is not new [5.32]. Combined coding allows an increase in possible data density on the media while maintaining the linear density of the recording. It was illustrated in Chapter 2 that modern day recording methods have practically reached the limit of allowable linear recording density, due to the effects of ISI, crosstalk, jitter and density dependent noise.

The conventional RLL modulation codes used in magnetic recording discussed in Sections 5.3 -5.7 have very poor Hamming distance properties. Indeed the sliding block codes discussed in section 5.7:2, magnify the effect of the channel errors by error propagation[5.33]. This can be seen in a plot of Channel Bit Error Rate against Viterbi decoded Bit Error Rate, the two codes used in this experiment were the Miller code and a Sliding Block Code.[5.34]

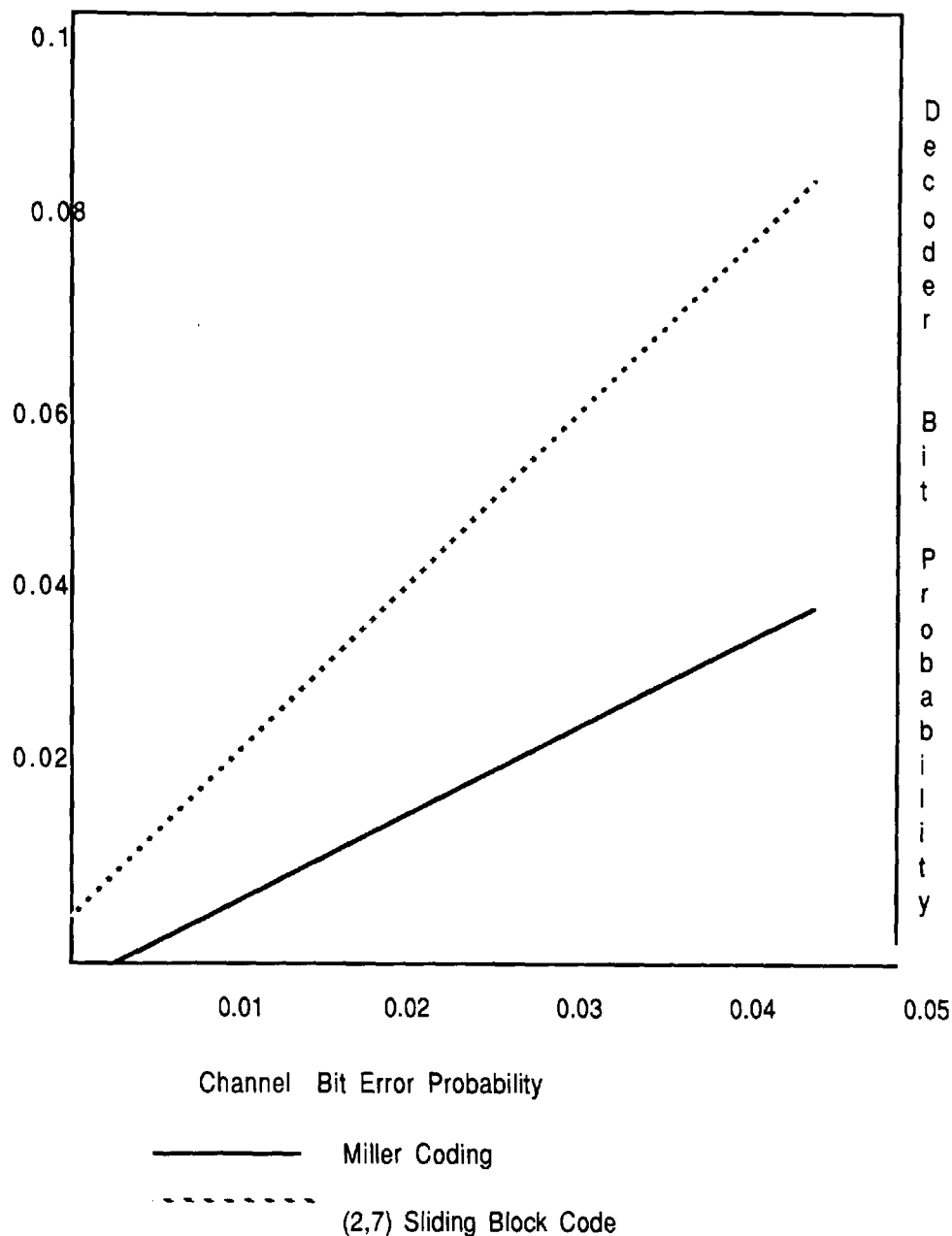


Figure 5.5 Error Rates for Two Standard Modulation codes[5.34]

5.8:2 Combined ECC and Runlength Limited Codes [5.35,5.36]

Error correction and modulation codes may be combined by preserving the Hamming distance of the code while imposing runlength constraints[5.35]. This can be achieved by bit stuffing; i.e. by adding an extra bit or series of 0 bits on to the codeword. This however does not improve the Hamming distance although it does give a simple method of modulating the code sequence[5.36]. This will not

lead to error propagation due to the choice of Modulation code.

In Section 4.1:2 it was stated that words with odd or even Hamming weight will always have a Hamming distance of at least 2. So allowing error detection using the parity of the Hamming weight. This leads to a simple case of runlength constrained error detecting block codes. The combined error detection and modulation code consists of runlength limited blocks of either odd or even weight. These codes have a minimum Hamming distance of 2 and so will be able to detect a single error.

5.8:3 Error Detection Using Runlength Constraints.[5.37,5.9]

The error control capability of a modulation code may be imposed by the choice of runlength constraints. There are three possible methods by which this may be done.

Method 1

This relies on the upper limit for the runlength, k to be twice that of the lower limit, d . So within this code the majority of binary symmetric errors will violate the runlength constraints.

The (1,2) (3,6) and (7,14) codes are typical examples of the $(d,2d)$ code. The error correcting capability of these codes is shown in the following example which uses a (3,6) code:

Original Modulation Vector 1 0001 00001 0000001 00001 0001

Received Modulation Vector 1 0001 000000000001 00001 01 01

Errors can be detected by blocks which do not fit the bounds on the runlength constraints. The third and fifth blocks do not fit the limits.

The first error is found as block length exceeds $k+1$ and the second error as the block length is less than $d+1$.

To use the (2,4) code will entail a reduction in the capacity of the channel to approximately 20% of the level in the industry standard (2,7) code given by Franzcek[5.9] as described in Section 5.6:2.

Method 2

The same principle is used to detect and correct the errors, however the runlengths have a wider range. In these codes the set of allowable runlengths is given by X . So for a pair of allowable runlengths $x_i, x_j \in X$ iff $x_i + x_j + 1 \notin X$. The allowable runlengths can be found from $(k, \dots, d, 2(d+1), \dots, 4(d+1))$, a numerical example of such a code is the (2,3,4,10,11,12,13,14,15,16,17,18,19,20) where the numbers in the brackets represent allowable runlengths. This gives the same error detection ability as the code above as it uses the same principle of allowable runlengths to correct errors.

Method 3

This uses a RLL code with no limitations on either d or k . However consecutive runlengths are only permissible if they obey the following criterion:

$$x_i, x_j \in X \text{ for which } x_i + x_j + 1 \notin X.$$

It is possible to construct a code (1,4) which is 77% efficient for a (1,4) code.

The runlength constraints of these codes only allow the correction of a single error and so all the codes will be susceptible to bit slip induced by double errors. In addition the Shift Channel Error, which was found by Howell[5.38] to be the most common in high density recording, will not be corrected by these codes, all of these codes depend upon a proportionally large increase or decrease in the runlength to detect errors. Whereas Shift Channel Errors only translates the position of the pulse by a single timing window. The majority of shift errors will therefore not be detected by this type of code.

5.8.4 Trellis Codes[5.33,5.34,5.39]

Many combined error correcting and modulation codes can be developed from Trellis diagrams. These are a series of nodes representing the states linked by arcs representing the input bit and the data sequence for the recorder[5.33]. The diagram below Figure 5.6 shows a trellis diagram for a code with just two states. The code is a rate 1/3 code with runlength constraints (1,3).

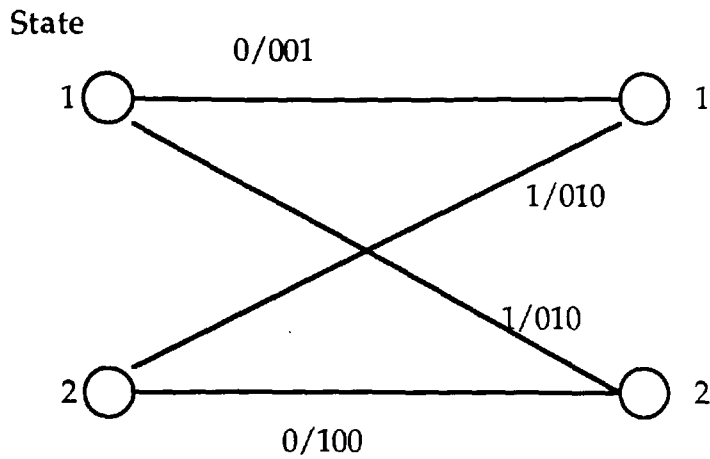


Figure 5.6 2 State Trellis code with free distance 4 [5.34]

Although the code given above is a comparatively simple one there are many applications for trellis codes in combined modulation and error control coding. Combined trellis codes may be mapped onto Convolutional Codes to enable the trellis codes to be synthesised[5.38].

To be able to understand the complexities of trellis codes, it is first important to define several terms and notations:

Terms

R code rate when k information bits are mapped onto n code bits the code rate is given by k/n ,

b Minimum runlength d in the above modulation,

l Maximum runlength k in the above modulation,

C upper bound on accumulated charge,

d_{rem} Minimum number of code differences between any two paths which have diverged from some point and will later remerge.

d_{\min} Minimum number of code differences between any two paths which have diverged from some point. Remergence is not necessary.

These terms can now be used to define a combined error correcting and modulation code. The following trellis diagram is for a rate 3/5 code with $v=2$, $d_{\min} = 3$ and $(b,l)=(0,4)$

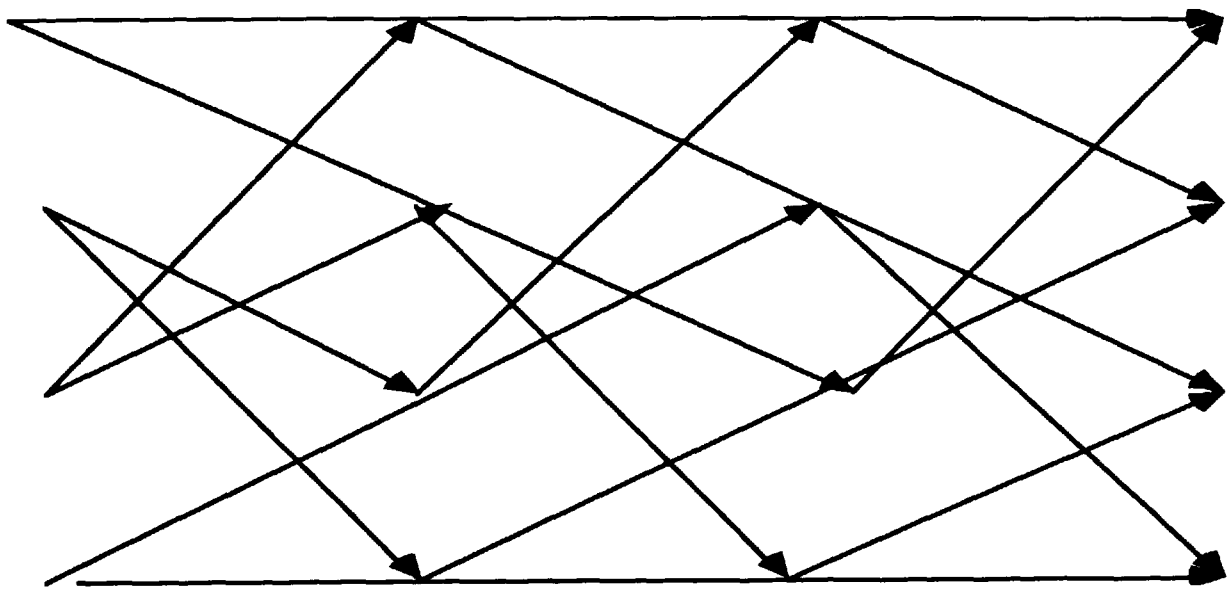


Figure 5.7 Trellis Diagram for Combined Modulation code[5.39]

An example of Trellis coding is the rate 3/8 code given by Ytrehus[5.40]. This uses the 32 (1,3) constrained codewords of length 8 and a 4 state encoder to give a constrained code of free Hamming distance 3. The codewords are given a decimal notation this shown in Table 5.12.

1	10101000	9	10010010	17	01010001	25	00101001
2	10101010	10	10010001	18	01001000	26	00100100
3	10101001	11	10001000	19	01001010	27	00100101
4	10100100	12	10001010	20	01001001	28	00100010
5	10100101	13	10001001	21	01000100	29	00010100
6	10010100	14	01010100	22	00101000	30	00010101
7	10010100	15	01010101	23	00101000	31	00010010
8	10010101	16	01010010	24	00101010	32	00010001

Table 6.11 Decimal Representation of 32 (1,3) Constrained Codewords of Length 8

The "into" state selection is done by the initial word which is drawn from $GF(2^3)$. To find the state the decimal representation of the binary word is divided by 2 and then 1 is added to the integer part of the state. The word is then selected to bridge the gap between the states. This mapping between the states is given in Table 5.13.

	Into(S_1)	Into(S_2)	Into(S_3)	Into(S_4)
Outof(S_1)	1,9	7,12	5,6	8,13
Outof(S_2)	4,11	2,21	15,19	3,10
Outof(S_3)	14,18	16,26	22,24	17,27
Outof(S_4)	23,31	28,29	25,32	20,30

Table 5.13 Conversion Table Between Different States for the (1,3) constrained code

This code is a non catastrophic (1,3) code with Code Rate $3/8$ and free Hamming distance 3. So the code will be able to correct single errors and prevent error propagation.

REFERENCES

- 5.1) **J.Watkinson** "The Art of Digital Audio" Section 6.6 "Simple Codes" Focal Press London and Boston 1988 pp176-179
- 5.2) **P.Peebles** "Digital Communications" Wiley International Series
- 5.3) **T.T. Doi** " Channel Codes for Digital Audio Recordings", Presented at the 70th AES Conference NY Oct 30-Nov2 1981
- 5.4) **T.Horiguchi and K.Morita** "An Optimisation of Modulation Codes in Digital Recording" IEEE Trans on Magnetics, Vol.MAG-12 No.6 Nov 1976 pp 740-742
- 5.5) **T.Tamura M.Tsutsimil H.Aoi et al** "A coding Method in Digital Magnetic Recording" IEEE Trans. on Magnetics Vol. Mag-8 No.3 Sept 1972 pp 612-613
- 5.6) **P.A. Franaaszek** " Sequence State Methods for Run-Length Coding" IBM Journal of Research and Development 1970 pp376-383
- 5.7) **C.E.Shannon** " A Mathematical theory of Communication" Bell Systems. Tech. J.Vol - 27 pp 379-423 July 1948
- 5.8) **E.Zhevai and J.K.Wolf** "On Runlength Codes" IEEE Trans on Information Theory Vol. 34 No.1 Jan 1988 pp 45-54
- 5.9) **K.Schouhammer Immink** "Coding techniques for digital recorders, Chapter 5, Run Length Limited Sequences " pp 81-161 Prentice Hall International 1991
- 5.10) **A.Gallopoullous, C. Heegard and P.H.Seigel** "The Power Spectrum of Runlength Limited Codes" IEEE Trans on Communications Vol. 37 no. 9 Sept 1989 pp 906-917
- 5.11) **N.D. Mackintosh** "The Choice of Recording Code", The Radio and Electronic Engineer, Vol. 50 No. 4, April 1980 pp 177-193
- 5.12) **J. Watkinson** " Coding For Digital Recording Chapter 3 Channel code Section 3.4 Group Codes" pp 59-61 Focal Press 1990
- 5.13) **A.Miller** "U.S. Patent 3,108,261" October 22 1963
- 5.14) **J.Mallinson and J.W. Miller** " Optimal codes for digital magnetic recording", Radio and Electronic Engineer Vol. 47 No. 4, Feb. 1963 pp 172-178
- 5.15) **A.Patel** "Zero Modulation Encoding in Magnetic Recording" IBM Journal of Research and Development Vol July 1975 pp366-378
- 5.16) **D.A. Lindholm** "Power Spectra of Channel Codes for Digital Magnetic Recording" IEEE Trans on Magnetics, MAG-14 No.5 Sept 1978 pp 321-323

- 5.17) **C.D. Mee and E.D. Daniel** Editors "Magnetic Recording" Mc Graw-Hill Volume II "Computer Data Storage" A.M. Patel Chapter 5 'Signal and Error Control Coding' Section 5.3 Modulation Codes pp 245-259
- 5.18) **N.K. Ouchi** "Apparatus for encoding and Decoding Data in a Modified Zero-Modulation Data Code" U.S.Patent 3,995,264 1973
- 5.19) **P.A.Franaszek** "Runlength Limited Variable Length Coding with error Propagation Limitation" U.S. Patent 3,689,899 1972
- 5.20) **G.V. Jacoby and R.Kost** "Binary rate two thirds code with Full Look-Ahead " IEEE Transactions on Magnetics Vol. MAG-20 No.5 pp 709-711
- 5.21) **H. Ogawa and K. Shouhammer Immink** "EFM - The Modulation code for the Compact Disc Digital Audio System", Premier Audio Engineering Society Conference June 1982 pp 117-124
- 5.22) **K. Schouhammer Immink and U. Gross** "Optimisation of low frequency properties of Eight to Fourteen Modulation", The Radio and Electronic Engineer Vol 53 No. 2 Feb. 1983 pp 63-66
- 5.23) **K. Schouhammer Immink** "Coding techniques for digital recorders, Chapter 2 Compact disc a design case" pp 15-32 Prentice Hall International 1991
- 5.24) **C.French and J.K.Wolf** "Alternative Modulation codes for Compact Disc", IEEE Transactions on Consumer Electronics, Vol. 34 No. 4 November 1988 pp 908-911
- 5.25) **T.Furukawa M.Ozaki and K.Tanaka** "On a DC Free Block Modulation Code" IEEE Transactions on Magnetics MAG-20 No 5 September 1984 pp 878-880
- 5.26) **S.Fukuda, T.Kojima, Y.Shipuku and K.Odaka** "8/10 Modulation Codes for Digital Recording", IEEE Trans on Magnetics MAG-22 1986 pp 1194-1196
- 5.27) **S. Joshi** " Making the LAN Connection with a Fiber Optic Standard" Computer Design Sept 1985 pp
- 5.28) **H.Yoshida T. Shimada and Y.Hashimoto** "8-9 Block code : A DC-Free Channel Code for Digital Magnetic Recording" SMPTE Journal Vol.92 No. 9 Sept. 1983 pp 918-922
- 5.29) **L. J. Fredrickson** "A $(d,k,c)=(0,3,5/2)$ Rate 8/10 Modulation Code" IEEE Trans on Magnetics MAG-26 1990 pp 2318-2320
- 5.30) **C.M.J. Van Uijen and C.P.M.J.Baggen** "Performance for a class of Channel codes for assymetric Optical Recording Channels" 7th VADR Conf, York pp 29-32
- 5.31) **D.T.Tang and L.R.Bahl** " Block Codes for a Class of Constrained Noiseless Channels" Information and Control Vol.17 1970 pp 436-461
- 5.32) **J.K. Wolf and G.Ungerboeck** "Trellis coding for Partial Response Channels" IEEE Trans on Communications Vol.Comm-34 Aug. 1986 pp 765-773

- 5.33) **P.Lee and J.K.Wolf** "Combined Error Correction/Modulation Codes" IEEE Trans on Magnetics" IEEE Trans on Magnetics Vol. MAG-23 No.5 Sept 1987 pp 3681-3683
- 5.34) **Y.Lin and J.K.Wolf** "Combined ECC/RLL Codes" IEEE Trans on Magnetics Vol. MAG-24 No.6 Nov. 1988 pp 2527-2529
- 5.35) **H.C. Ferreira, J.F.Hope and A.L.Nel** "Binary Rate Four Eighths Runlength Constrained Error Correcting Magnetic Recording Code" IEEE Trans on Magnetics Vol. MAG-22 Sept 1986 pp 1197-1199
- 5.36) **C.A.French** "Distance Preserving Runlength Limited Codes" IEEE Transactions on Magnetics Vol. MAG-25 No.5 Sept 1989 pp 4093-4095
- 5.37) **H.C. Ferreira and S.Lin** "Of Integer Compositions and Error Correcting (d,k) Block Codes"Submitted to IEEE Transactions on Information Theory
- 5.38) **T.D. Howell** " Analysis of Correctable Errors in the IBM 3380 disk file" IBM J.Res. Dev. Vol 28 1984 pp 206-211.
- 5.39) **H.C Ferreira** "The Synthesis of Magnetic Recording Trellis Codes with Good Hamming Distance" IEEE Trans on Magnetics Vol MAG--23 No.5 Sept 1985 pp 1356-1358
- 5.40) **O.Ytrehus** "A Rate 3/8 (1,3) Constrained code with free Hamming Distance 3" Reports in Informatics, Report N0. 38, Department of Informatics University of Bergen, September 1989

Chapter 6

Theoretical Results

6.1 Introduction

This chapter sets out the theoretical and statistical results concerning the effects of the two primary Non-clockloss error mechanisms. This includes a discussion of the PDF of ISI induced bitshift and its convolution with Noise induced bitshift. Plots were drawn of the effect of combining these two primary non-clockloss error mechanisms on the output error rate.

Two different PDFs for noise induced bitshift were examined the first given by Katz and Campbell[6.1] and the second by Ryley and Loze[6.2]. The differing effects of these distributions on the error rate were compared.

The correlation of these error mechanisms pulse to pulse is investigated to discover how this effects the error statistics. This is to examine the widespread theory that the effects of Non-clockloss errors can be regarded as being broadly independent pulse-to-pulse.

The effects of different demodulation schemes for the Manchester Modulation code were examined to find their effect upon the output Error Rate of the code. This leads to a considerable reduction in the Error Rate of the code. The phenomena of aliasing was also investigated, this leads to additional errors by the Error Correction code falsely correcting bits not in error due to the error syndrome.

The effect of a bitflip is demonstrated in the case of the Ferreira combined modulation and error correcting code, this resulted in the complete loss of the signal from the channel. A method to prevent bitflips in these combined codes is also discussed; these codes have a lower ratio of minimum to maximum runlengths to the standard Ferreira combined coding scheme.

6.2 Noise Induced Bitshift

6.2:1 The PDF of Noise Induced Bitshift

Two different models for the PDF of Noise induced bitshift are considered here:

The first that of the Katz and Campbell[6.1]. This model assumes that there is a low packing density so giving an isolated pulse. The isolation leads to a negligible amount of ISI induced bitshift. The model assumes that the signal is distorted by zero mean Gaussian Noise source $n(t)$ with mean square value σ_0^2 . The bitshift from the centre of the timing window t_0 is given by τ .

Therefore at a zero crossing the detector output is:

$$s'(t_0 + \tau) = -n'(t_0 + \tau) \quad (6.1)$$

Where:

s' is the differentiated signal;

n' is the differentiated noise;

τ is the amount of bitshift.

If the SNR is sufficiently large then the amount of bitshift will be small. In addition the noise has only a small bandwidth and is therefore slowly varying. This will lead via the first term of the Taylor expansion to a figure for the bitshift of :

$$\tau = \frac{-n'(t_0)}{s''(t_0)} \quad (6.2)$$

Eq. (6.2) then gives the PDF of the bitshift which is shown in the Function given below.

$$F(\tau) = \frac{\mu}{[\sigma_i^2 2\pi]} \exp \frac{[-\mu^2 \tau^2]}{[2\sigma_i^2]} \quad (6.3)$$

Where:

μ is the second differential of the signal;

σ_i^2 is the mean squared value of the differentiated noise;

τ is the amount of bitshift

In addition to the model for the noise induced bitshift given in Eq. (6.3) there is also a model given by Ryley and Loze[6.2], it assumes that the bandwidth of the noise is constrained by the recording channel, so that the signal and noise have comparable bandwidths. Expanding the signal about the centre of the timing window gives:

$$\tau s''(t_0) = -n'(t_0) - \tau n''(t_0) \quad (6.4)$$

Rearranging gives :

$$\tau = \frac{-n'(t_0)}{[\mu + n''(t_0)]} \quad (6.5)$$

Where:

$n''(t_0)$ is the second differential of noise with respect to time

This leads to the following equation for the probability density function of bitshift:

$$F(\tau) = \frac{\sigma_1 \sigma_2}{\pi(\sigma_1^2 + \tau^2 \sigma_2^2)} \exp \left\{ \frac{\mu^2 \tau^2}{2(\sigma_1^2 + \tau^2 \sigma_2^2)} \right\} \times \left(\exp \left\{ -\frac{\mu^2 \sigma_1^2}{2\sigma_2^2(\sigma_1^2 + \tau^2 \sigma_2^2)} \right\} \right. \\ \left. + \frac{\mu \sigma_1}{\sigma_2} \left\{ \frac{\pi}{2(\sigma_1^2 + \tau^2 \sigma_2^2)} \right\}^{1/2} \operatorname{erf} \left\{ \frac{\mu \sigma_1}{\sigma_2 [2(\sigma_1^2 + \tau^2 \sigma_2^2)]^{1/2}} \right\} \right)$$

Where:

σ_2^2 is the mean squared value of the second derivative of the noise

The above gives the PDF for the bitshift as defined by Ryley and Loze it is symmetric about the centre of the timing window.

The PDF of Ryley and Loze varies from that given by Katz and Campbell in the use of the term σ_2 . The relationship between σ_0 , σ_1 and σ_2 is given overleaf:

$$\sigma_0^2 = \int_{-\infty}^{\infty} W(f) df$$

$$\sigma_1^2 = \int_{-\infty}^{\infty} 4\pi^2 f^2 W(f) df \quad (6.7)$$

$$\sigma_2^2 = \int_{-\infty}^{\infty} 16\pi^4 f^4 W(f) df$$

Where:

$W(f)$ is the PSD of the noise at the input to the differentiator

6.2:2 Comparison of the Noise PDFs

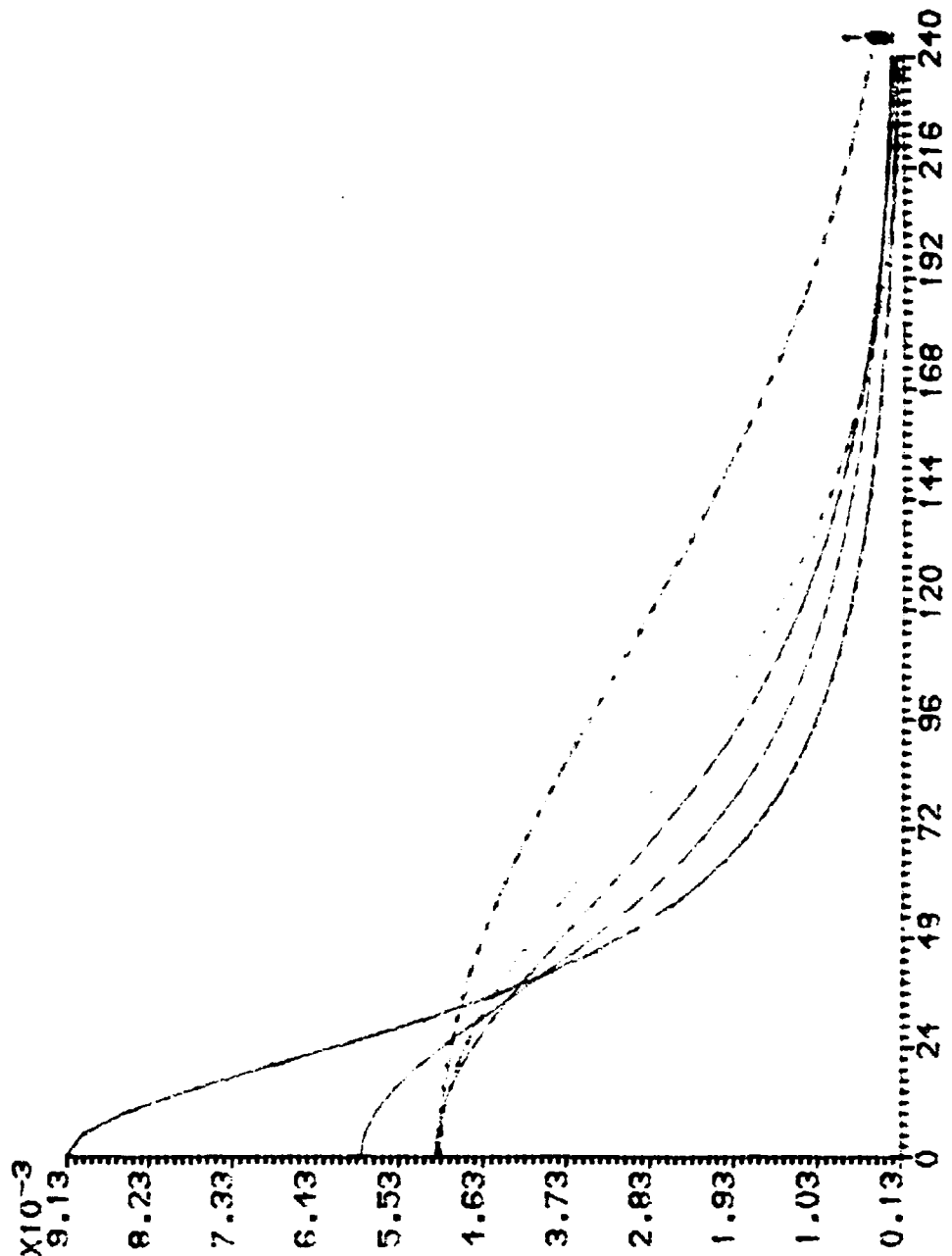
There is a relationship between the two PDFs (Eq. 6.3 and Eq. 6.6). The relationship can be expressed through a variable A . This variable is a normalisation factor which is defined as being the ratio of σ_2 to σ_1 . The higher the value of A then the greater the difference between the two PDFs. This is shown below in Figure 6.1 which plots the PDF given by Ryley and Loze against τ for different values of A .

The density functions with the lower variance with respect to τ have a higher value of A . If the value of A is zero, i.e. when σ_2 is equal to zero then the Ryley and Loze model collapses to that given by Katz and Campbell. This is shown in Figure 6.1 as the plot whose PDF at $\tau = 0$ is the least.

In addition the effects of the differences in the two PDFs can be modelled in Figures 6.2 and 6.3. These plot the log of the induced bit error rate against the nominal signal to noise ratio in decibels. The effect of the Katz and Campbell PDF for noise induced bitshift, is shown in Figure 6.2. The effect of the Ryley and Loze PDF for noise induced bitshift is shown in Figure 6.3 .

The SNR referenced along the top of the graph, is taken from the tape as being $10\log_{10} V_0^2/\sigma_0^2$. These plots illustrate of how optimistic the PDF postulated by Katz and Campbell for noise induced bitshift error was compared to the PDF derived by Ryley and Loze.

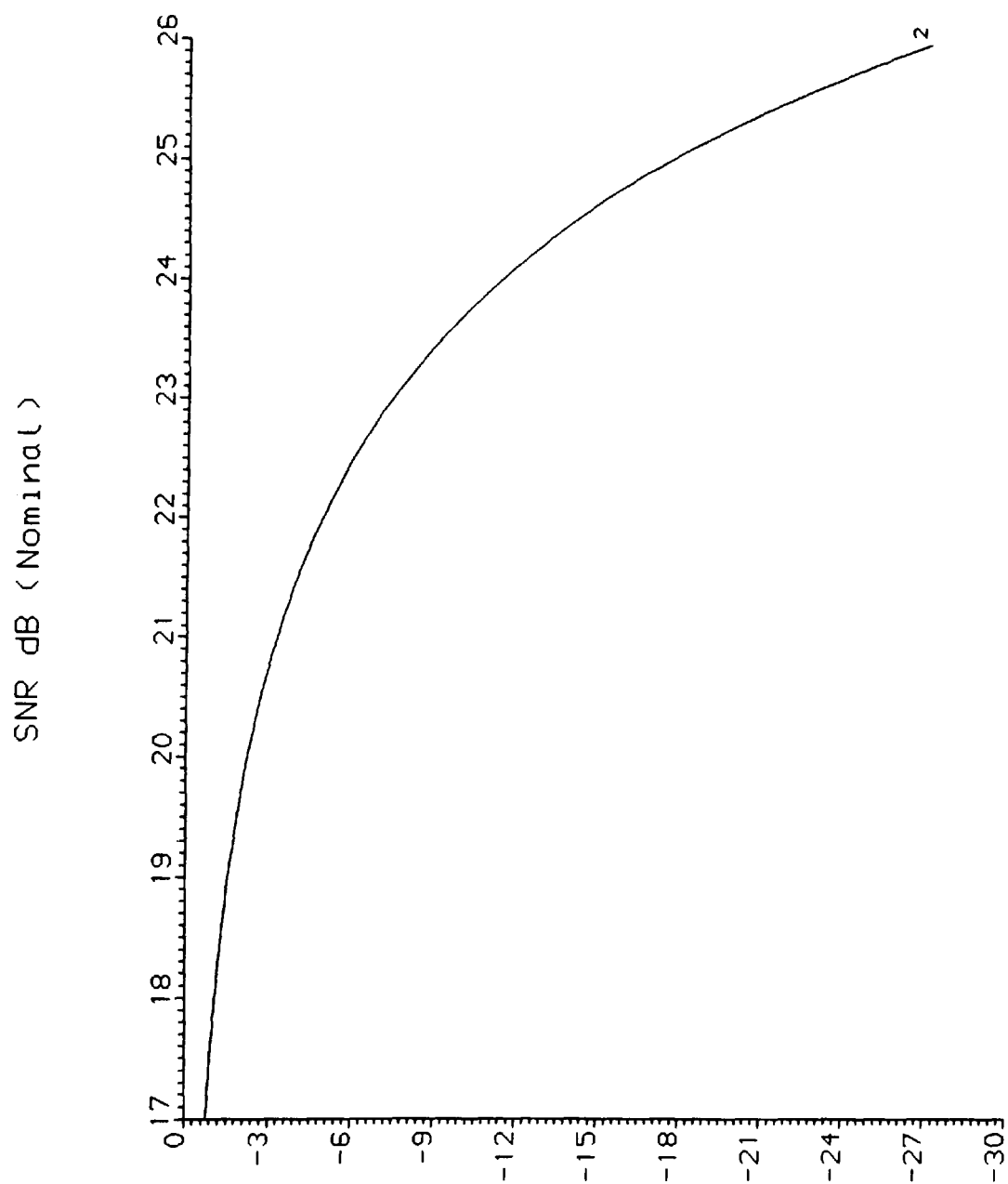
COLOURED NOISE



1	$A=0$
2	$A=0.25$
3	$A=0.5$
4	$A=1$
5	$A=2$

Figure 6.1 Ryley and Hoze Noise Distribution as A varies

Log of the Error Rate against SNR for Katz and Campbell



Katz and Campbell Error Rate

- | | |
|---|--|
| 1 | |
| 2 | Katz and Campbell Noise Induced Bitshift |

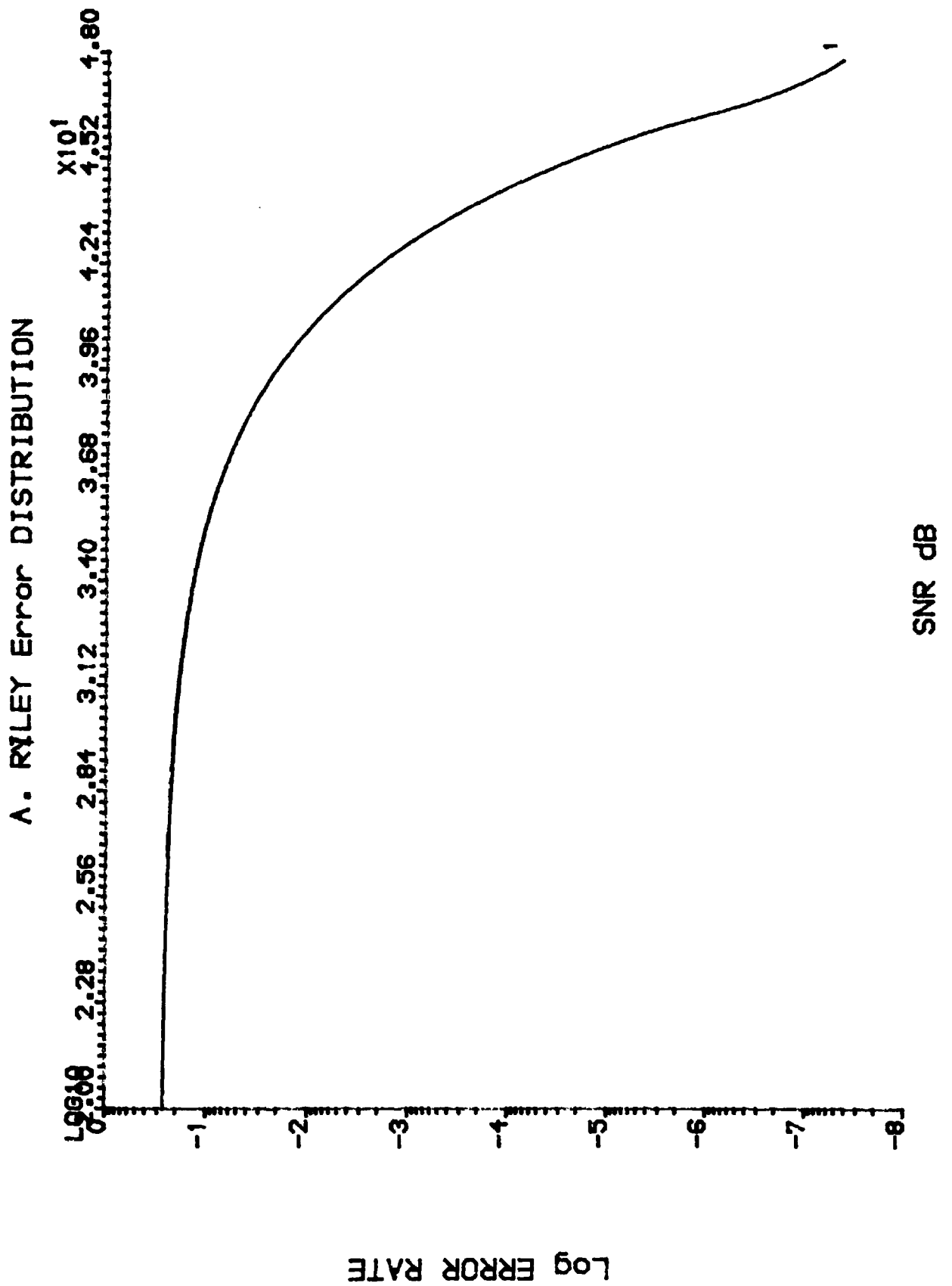


Figure 6.3 Ryley and hoze Error Rate against SNR.

6.2:3 The Effect of Windows on the Noise Induced Error Rate

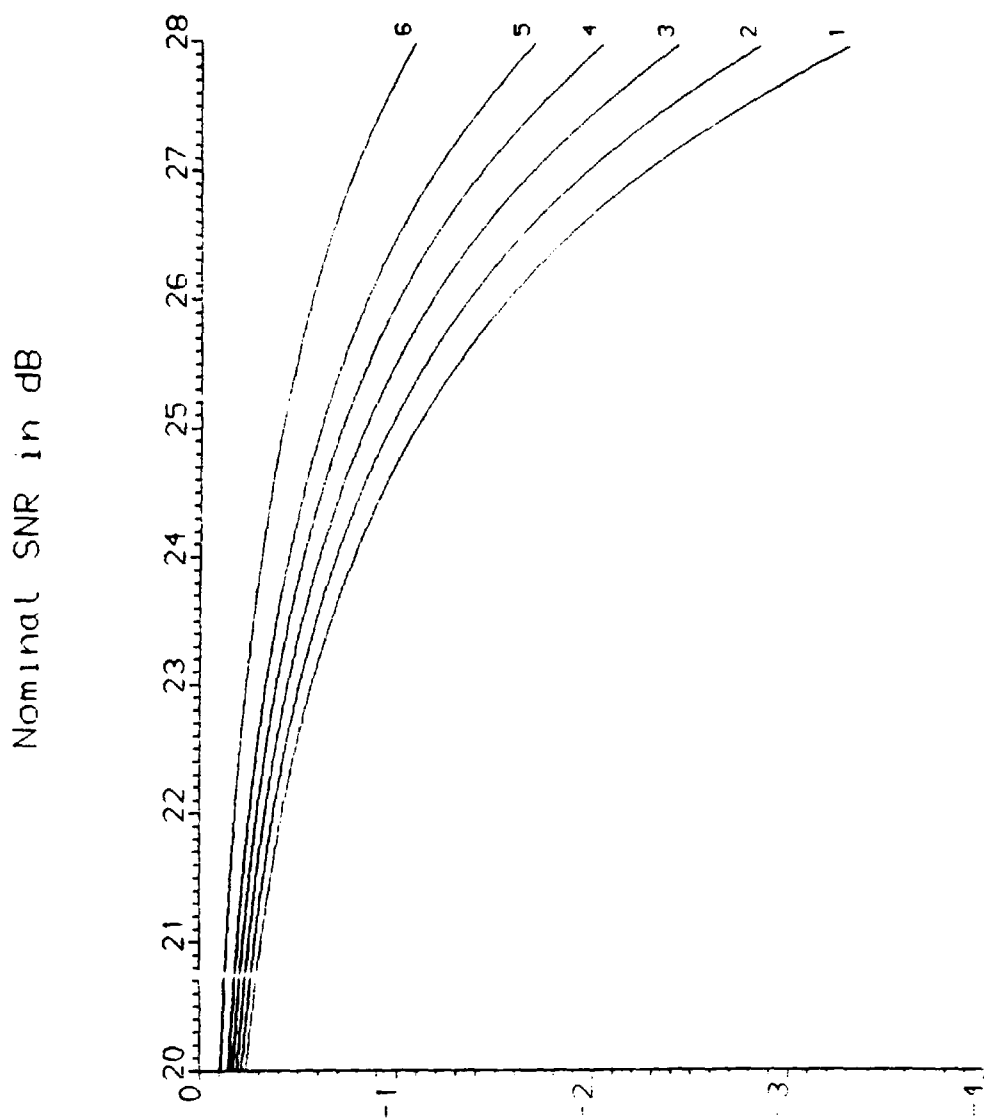
A symmetric Lorenzian type replay pulse is assumed. Hence with zero mean noise the timing window is centred upon the peak of the pulse. If a non-symmetrical pulse had been assumed then the Optimum Timing Window would not be symmetrical about the peak of the pulse. Similarly if any DC bias was present then the peak of the replay pulse would be shifted in the timing window.

Guard bands are used to prevent a Shift Channel Error, though there is a penalty due to the reduction in the functional size of the timing window. Figure 6.4 and 6.5 plot log error rate against SNR for the Katz and Campbell PDF. Figure 6.4 is for noise only while Figure 6.5 includes the additional errors due to ISI. The increasing proportion of the timing window used by the guard band is illustrated by the different plots. Figures 6.4 and 6.5 show that a timing window with guard bands which are sufficiently large to compensate for all shift channel errors will lead to a decrease in the effective SNR. This will lead to a timing window which is reduced to give no viable read out signal.

However for recordings with low SNR a large guard band is required to prevent shift errors. Therefore the optimum size of the guard band depends upon:

- the shape of the replay pulse;
- the raw error rate;
- the pdf of bitshift;
- the density of recording;
- the minimum run length of the modulation code.

Figure 6.4 The effect of guard bands on Error Rate (Noise Only)



Plot showing the error rate for different guard band

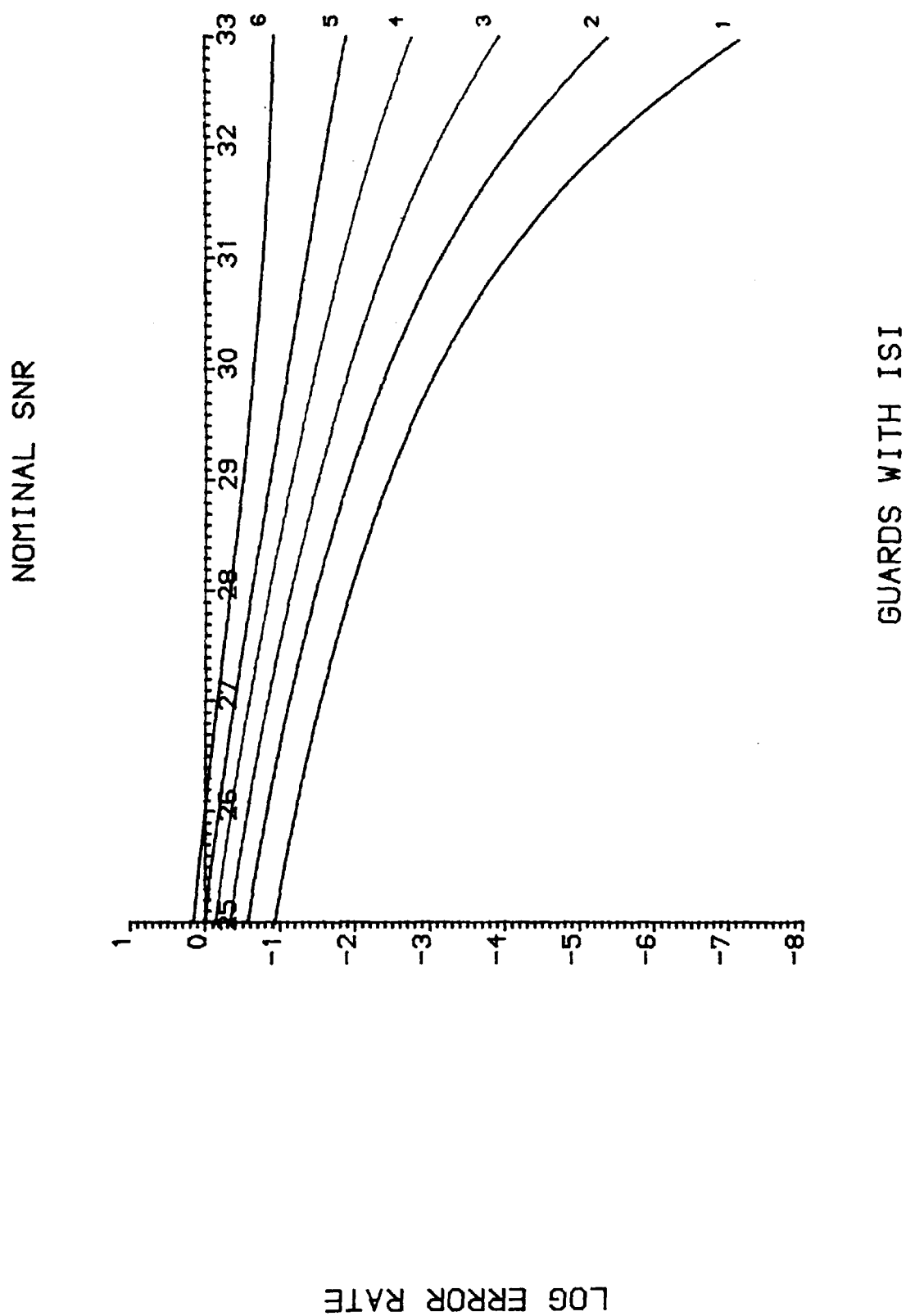


Figure 6.5 The effect of guard bands on Error Rate (Noise and ISI)

6.2:4 The Autocorrelation of Noise

One of the primary assumptions about noise induced bitshift was that it was independent pulse to pulse. To examine this hypothesis it is necessary to find the autocorrelation function of noise induced bitshift. However as the ACF is normalised then only the ACF of the channel noise and its derivatives need be found.

The Autocorrelation of the channel noise was found by taking the inverse Fourier Transform of the channel noise and its first and second derivatives. The Autocorrelation is the standardised Autocovariance of these distributions. The Autocorrelation function is given by the inverse Fourier Transform of the PDS of the noise. The choice of the tape noise is made to give a good approximation of the bandlimited noise channel while retaining a mathematically simple Autocorrelation function. The Power Density Spectrum of the channel noise is given by $W(f)$ [6.3]. From the inverse Fourier transform of the PDS it is possible to drive the autocorrelation function of the noise, which is given below;

$$\text{ACF } n_0(\phi) = \sqrt{2} \propto e^{-((\pi\beta\phi)^2 / 4)} \quad (6.8)$$

Where:

\propto is the constant of proportionality

β is the bandwidth

ϕ is the time difference for the correlation

Eq. (6.8) gives the Autocorrelation function of the undifferentiated channel Noise given by n_0 however it is also important to examine the Autocorrelation function of the differentiated channel Noise rather than the Autocorrelation of the channel Noise itself. The PDS of the differentiated noise is given by $4\pi^2 f^2 W(f)$. This then yields the Autocorrelation function;

$$ACF_{n_0'}(\phi) = -(2\pi) \propto 2(\beta\pi)^2 \beta e^{-(\pi\beta\phi)} [1-2(\beta\pi)^2 \phi^2] \quad (6.9)$$

Finally the ACF of n_0'' is given by the inverse Fourier transform of its PDS. The PDS is equal to $16\pi^4 f^4 W(f)$. This gives the Autocorrelation function of the second differential of the noise to be:

$$ACF_{n_0''}(\phi) = (-6(\beta\pi)^2 + 24(\beta\pi)^4 \phi^2 + 8(\beta\pi)^6 \phi^4) k \quad (6.10)$$

Where;

$$k \text{ is } -(2\pi) \propto 2(\beta\pi)^2 \beta e^{-(\pi\beta\phi)}.$$

However to give the correct reading all the above have to be normalised at $\phi=0$ to give the correct Autocorrelation functions.

$$\begin{aligned} ACF_{n_0}(\phi) &= e^{-(\pi\beta\phi)} \\ ACF_{n_0'}(\phi) &= [1-2(\beta\pi)^2 \phi^2] e^{-(\pi\beta\phi)} \\ ACF_{n_0''}(\phi) &= (1 - 4(\beta\pi)^2 \phi^2 - \frac{4}{3}(\beta\pi)^4 \phi^4) e^{-(\pi\beta\phi)} \end{aligned} \quad (6.11)$$

The two graphs Figure 6.6 and Figure 6.7 give plots of the Autocorrelation functions for the first and second derivatives of the channel noise. Figure 6.6 is for the autocorrelation of the first differential of the noise, while Figure 6.7 is for the autocorrelation of the second differential of the noise.

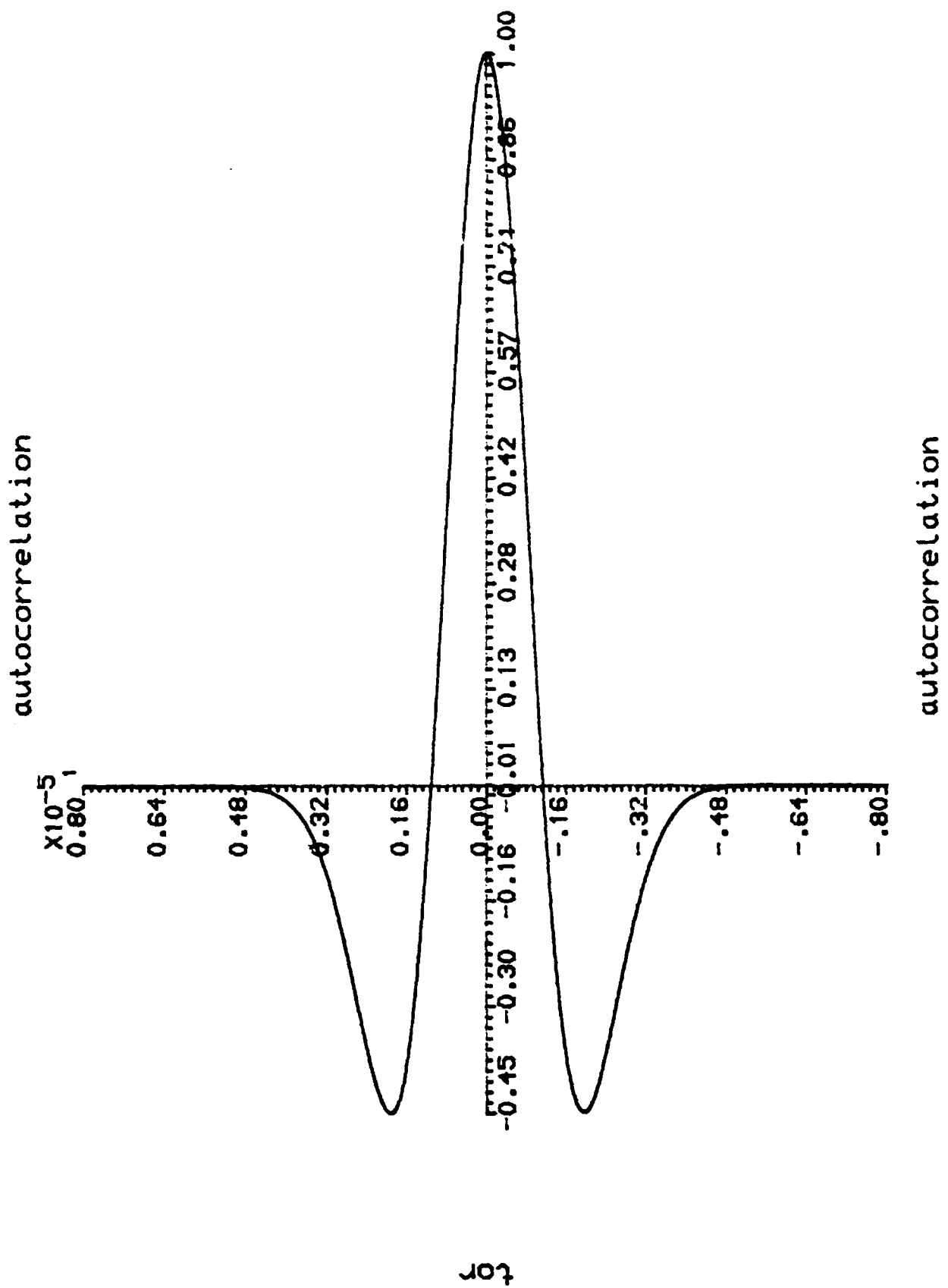


Figure 6.6 Autocorrelation of $\rho'(\phi)$

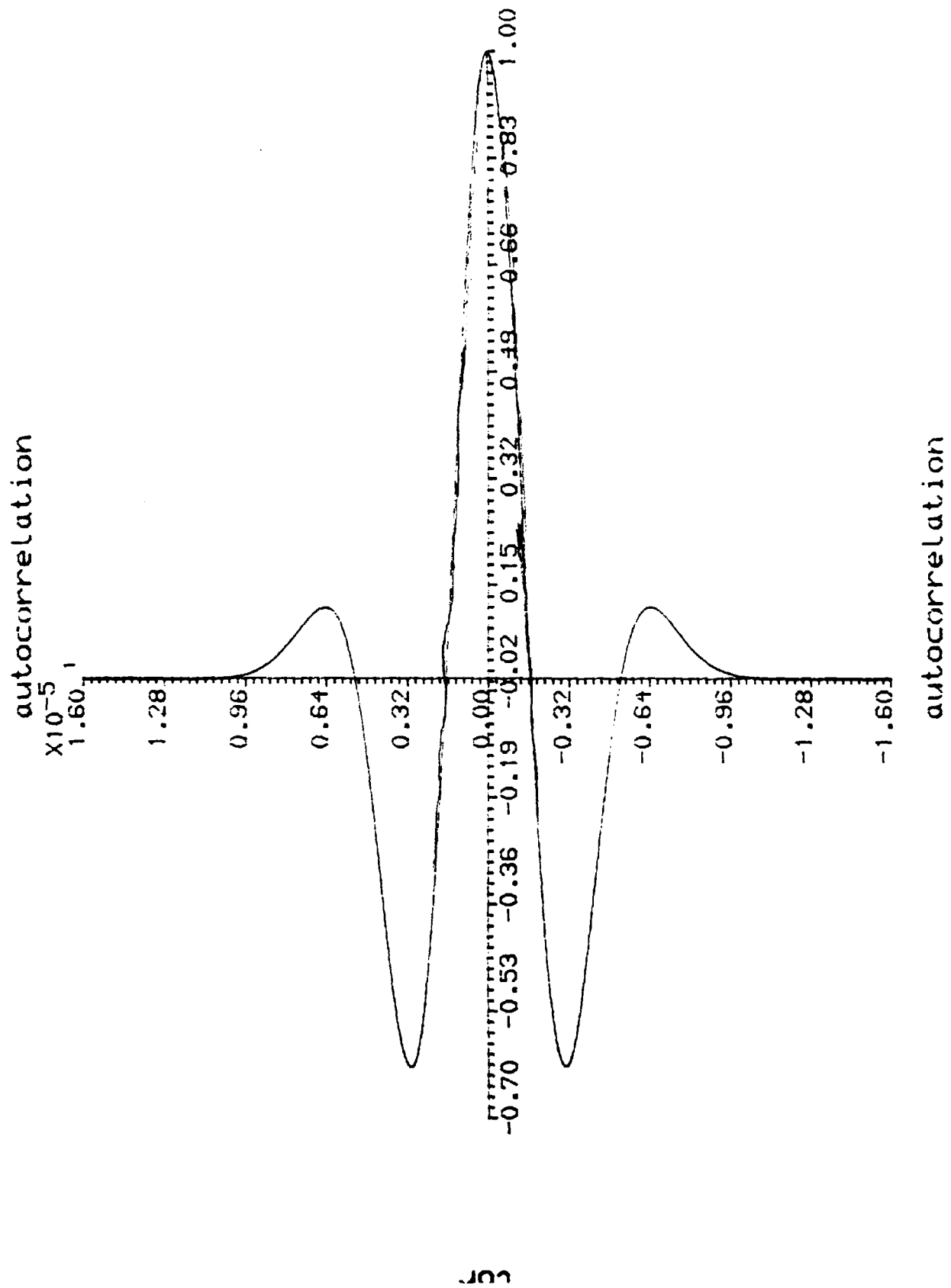


Figure 67. Autocorrelation of $n''_0(\phi)$

The time difference over which the level of Autocorrelation is measured is shown by the figure for which runs along the vertical axis. The auto-correlation is measured horizontally. The bandwidth is set to the value of 100 KHz, which is a typical value for a recording channel[6.3]. In addition the tape speed at 15ips is the same for the two plots. This enables a comparison to be made between the two functions as the Autocorrelation pulse to pulse is dependent upon both the bandwidth and the tape speed.

The effects of altering the bandwidth and the tape speed can be shown in Table 6.1 which compares different bandwidths and tape speeds against the value of noise autocorrelation pulse to pulse for the first five fundamental distances to the pulse under observation.

	1st FD	2nd FD	3rd FD	4th FD	5th FD
<u>Bandwidth =50 KHz</u>					
Speed =20ips	0.145	-0.617	-0.229	0.103	0.078
Speed =30 ips	0.562	-0.247	-0.617	-0.413	-0.666
Speed =40 ips	0.741	0.145	-0.402	-0.617	-0.497
<u>Bandwidth=100 KHz</u>					
Speed =20ips	-0.617	0.103	0.0187	0.0	0.0
Speed =30 ips	-0.247	-0.413	0.103	0.053	0.0049
Speed =40 ips	0.145	-0.617	-0.229	0.103	0.078
<u>Bandwidth =200 KHz</u>					
Speed =20ips	0.103	0.0002	0.0	0.0	0.0
Speed =30 ips	-0.413	0.053	0.0002	0.0	0.0
Speed =40 ips	-0.617	0.103	0.0187	0.0	0.0
<u>Bandwidth=300 KHz</u>					
Speed =20ips	0.0187	0.0	0.0	0.0	0.0
Speed =30 ips	0.103	0.0002	0.0	0.0	0.0
Speed =40 ips	-0.229	0.0187	0.0	0.0	0.0

Table 6.1 Level of Autocorrelation for first five fundamental distances

From the table it can be seen that to avoid autocorrelation a wide band recording channel is desirable, in addition the lower the tape speed the better the resistance of the recording channel to the effects of noise Autocorrelation. The recording density is kept constant for these experiments but it will also play a major part in reducing the correlation pulse to pulse. As at a recording density of 20% of that

used to tabulate the fundamental distance would be five times that used for the table.

6.3 Theoretical results involving ISI

6.3:1 The Effect of ISI on Magnetic Recording

As has been shown ISI is the cumulative effect of the interaction of the tails of the various replay pulses [6.4]. As an error mechanism ISI's importance is due to its relationship to additive noise. The combined effect of these two mechanisms leads to a higher error rate than produced by Noise alone disregarding the effects of ISI.

The amount of bitshift due to ISI is found by selecting a value of ISI induced bitshift from an array together with its probability. The noise model was the same as used for the previous results which did not involve any ISI. The probability of error due to the combined effect of ISI and noise was modelled by multiplying the probability of ISI induced bitshift by that of the level of noise required to create an error compared to the edge of the timing window.

Plots were made for the combined, noise and ISI induced bitshift error rate, these are shown in Figures 6.8, 6.9, and 6.10. These plots can then be compared to those for the Noise only error rates.

It is important to note that ISI causes more errors at high packing density than at low ones, this is illustrated by Figure 6.9 and 6.10 which are for ISI with the same noise source with different packing densities.

The recording can be regarded as a series of pulses with their peaks in the centre of each of the timing windows. ISI can be thought of as a translation of the peak of the recording pulse. Although the peak stays within the timing window it is now closer to one side than the other.

Noise can cause a further shift in the position of the pulse, as the distribution for Noise induced bitshift is assumed to be symmetric and of zero mean this results in an increase in the error rate. The distribution for the noise induced bitshift is Gaussian the probability of bitshift is higher in the region which is between $\beta - \alpha$ and α standard deviations from the centre of the distribution than in the region between α and $\beta + \alpha$ standard deviations from the centre of the distribution than in the region. This difference in the sizes of the two regions can be expressed mathematically as:

$$\begin{aligned}
 & 2 \int_{\alpha}^{\infty} f(\tau) d\tau < \int_{\beta+\alpha}^{\infty} f(\tau) d\tau + \int_{\beta-\alpha}^{\infty} f(\tau) d\tau \\
 & \int_{\alpha}^{\infty} f(\tau) d\tau - \int_{\beta+\alpha}^{\infty} f(\tau) d\tau < \int_{\beta-\alpha}^{\infty} f(\tau) d\tau - \int_{\alpha}^{\infty} f(\tau) d\tau \quad (6.12) \\
 & \int_{\alpha}^{\beta+\alpha} f(\tau) d\tau < \int_{\beta-\alpha}^{\alpha} f(\tau) d\tau
 \end{aligned}$$

Where :

$f(\tau)$ is the PDF of the noise;

α is the half timing window width divided by the deviation of the noise;

β is the ISI induced bitshift divided by the deviation of the noise.

The first line of Eq. (6.12) states that the probability of a noise induced bitshift error is less than that of an error when noise is convolved with ISI. The second line separates bitshifts, to give the shift whose probability is increased by the action of the ISI on the right hand side of the inequality and the shift whose probability is decreased by the action of the ISI on the left.

6.3:2 Formulation of ISI

The amount of ISI present in the recording system is dependent upon several variables. These included:

- the shape of the recording pulse;
- the packing density of the recording;
- the choice of the Modulation code.

The pulse shape used in this thesis is the Lorentzian pulse. This gives a figure for the ISI of:

$$ISI = \sum_{j=-\infty}^{\infty} w_j \frac{V}{(1+(j\mu/T_{50})^2)} \quad (6.13)$$

If a differential detection system is used such as the gated crossover then it is first necessary to differentiate the original value of ISI given in Eq.(6.13). In this case the

following figure is of interest:

Differentiating the above gives

$$ISI = \sum_{j=-\infty}^{\infty} w_j \frac{-2Vi\mu}{((T_{50})^2 + (j\mu)^2)^2} \quad (6.14)$$

Where:

$w_j \in \{-1,0,1\}$ the polarity of the j^{th} pulse

j is the index value;

V is the peak voltage;

T_{50} is the half pulse width;

μ is the length of the timing window.

To obtain the horizontal bitshift the vertical shift at the point of zero crossing Eq.(6.14) must be divided by $s''(t_0)$ the second differential of the pulse at its peak.

Now:

$$s''(t_0) = -2V/(T_{50})^2. \quad (6.15)$$

So the ISI induced bitshift is given by:

$$ISI = \sum_{j=-\infty}^{\infty} w_j \frac{i\mu}{(1 + (i\mu/T_{50})^2)^2} \quad (6.16)$$

As with Noise it is important to distinguish between the ISI induced bitshift and the figure given for the vertical level of ISI. In the case of the ISI, the induced bitshift is the amount which the zero crossing of the differentiated pulse is translated along the horizontal axis. However the figure given for the ISI is the

weighted sum of the tails of the neighbouring pulses.

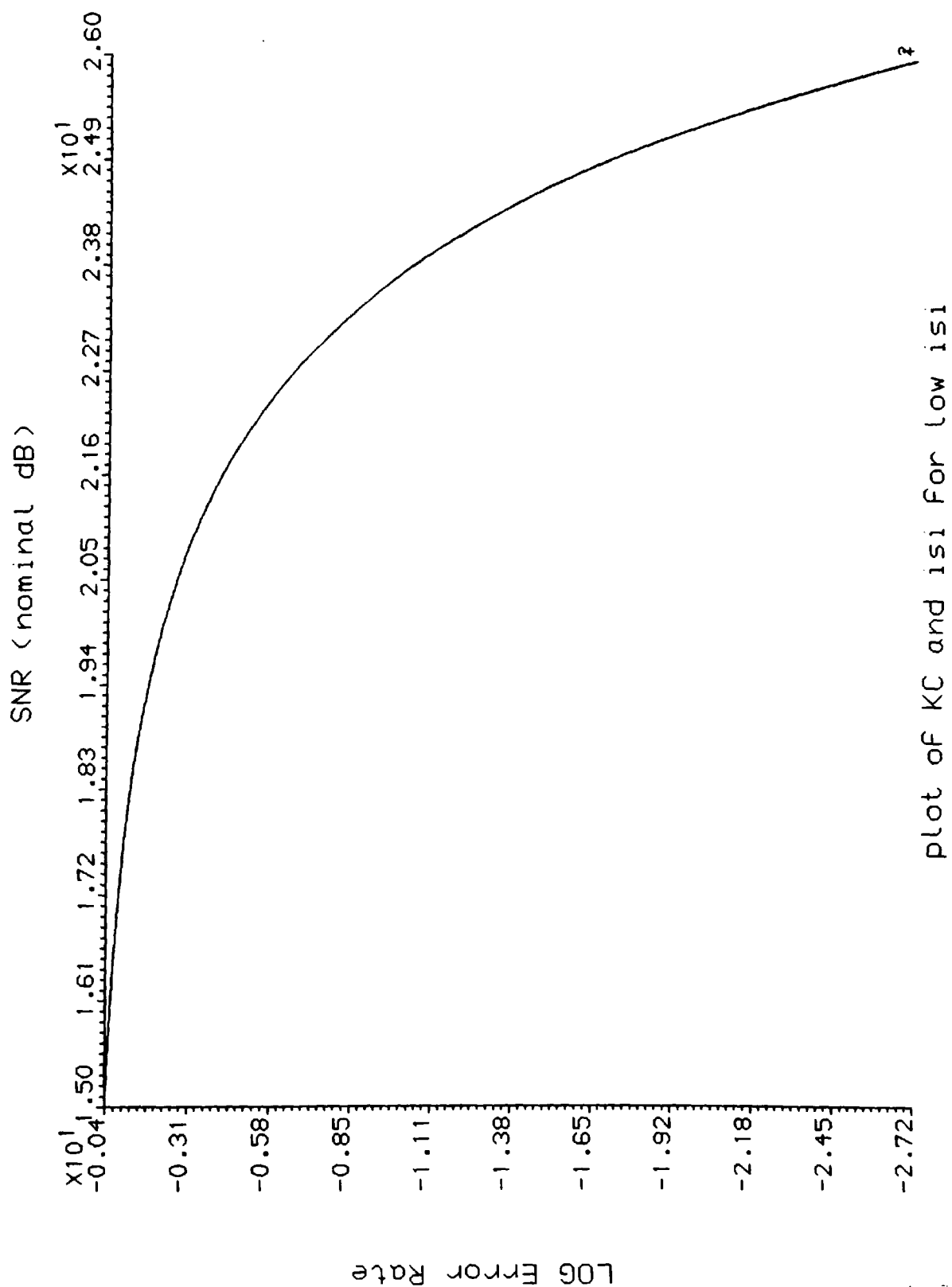


Figure 6.8 Katz Campbell Noise Induced Bitshift with ISI
(Low Density)

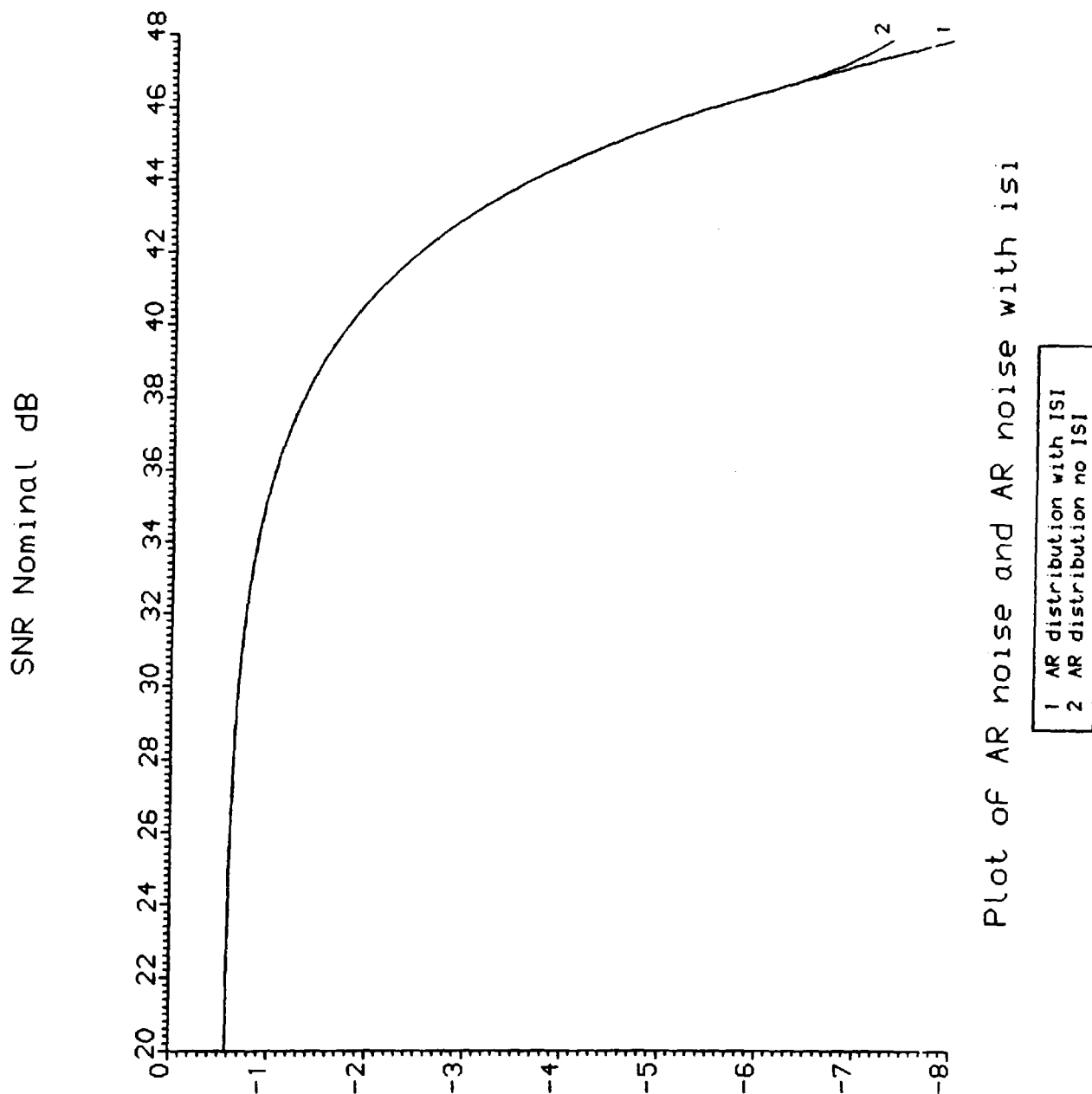


Figure 6-9 Ryley and Loze Bitshift Distribution with ISI (LOW DENSITY)

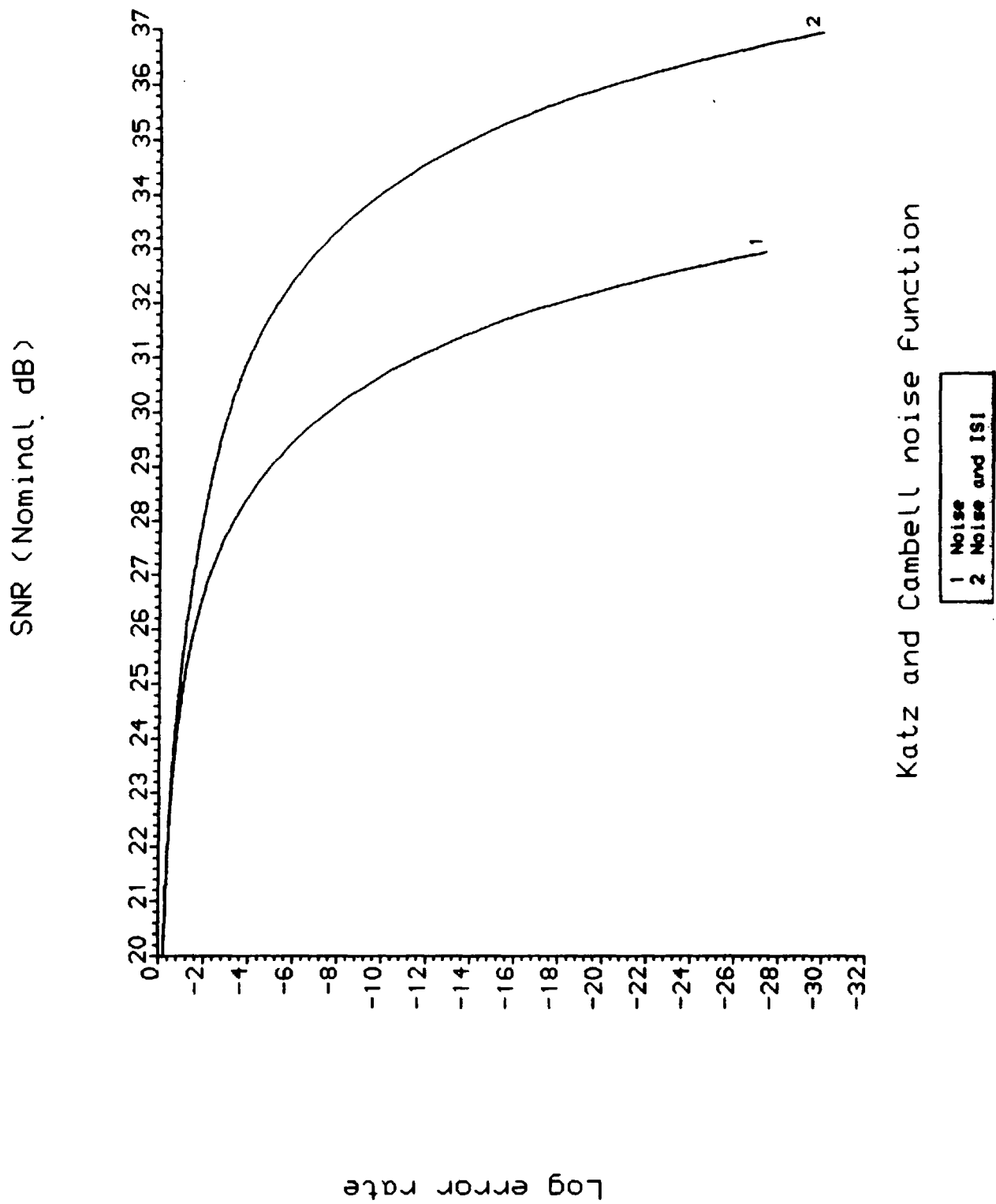


Figure 6-10 Katz and Campbell Bitshift with ISI (High Density)

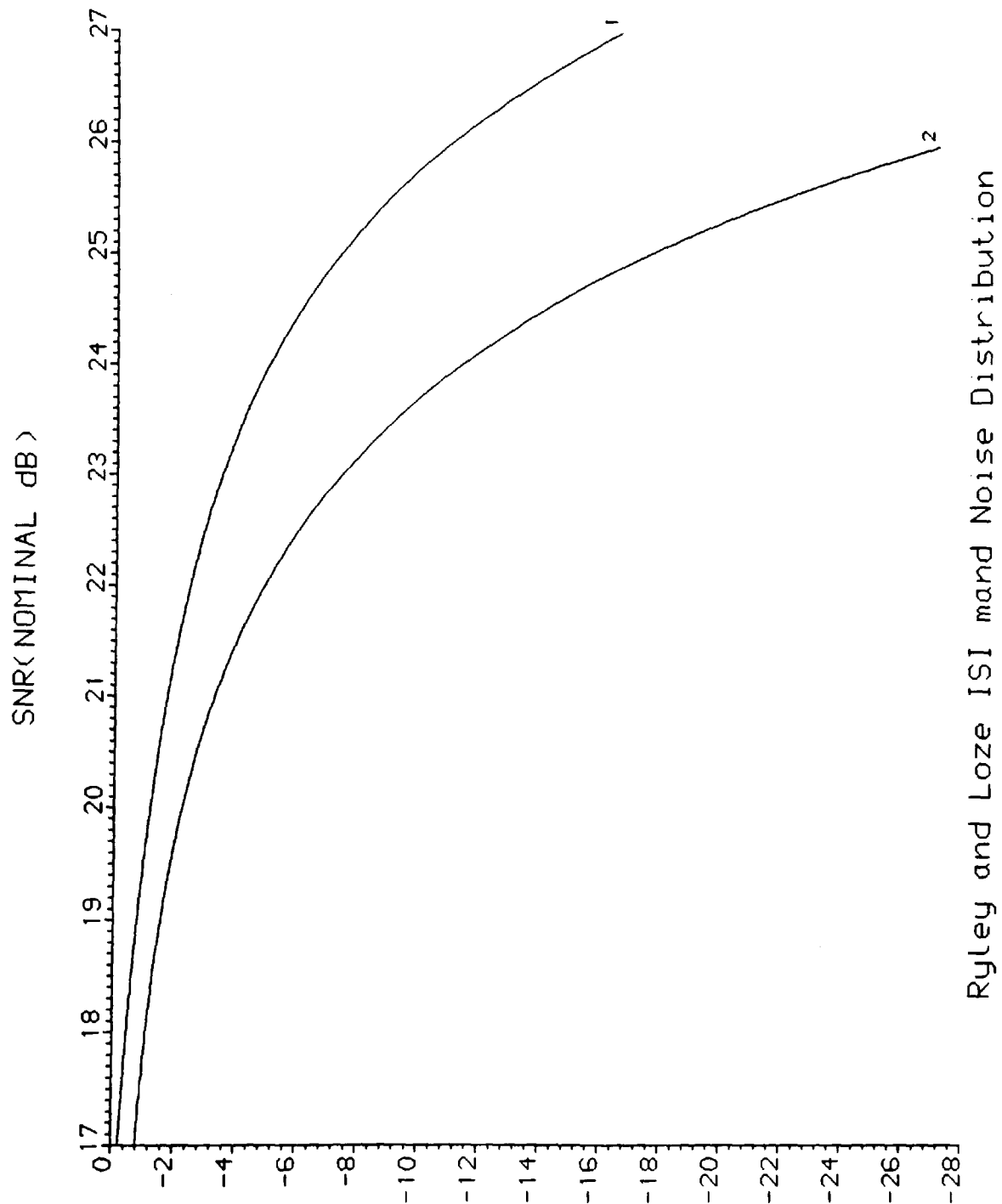


Figure 6-11 Ryley and Loze Bitshift Distribution with ISI
Log Error Rate (High Density)

The packing density effects the level of ISI in two ways:

by reducing the fundamental recording distance and hence the size of the timing window. This leads to an increase in the number of errors. As bitshift errors may now be caused by lower values of ISI and noise;

the high packing densities mean that a greater number of bits become involved in the process of ISI. It has been shown by Loze[6.3], that for each Kilo Bit per Inch, KBPI of modulation pulses an additional transition effects the ISI.

The higher the data density the greater the number of individual spikes of ISI whose height will also increase so giving a greater bitshift effect. This can be best illustrated by comparing the ISI at 4 KBPI and that at 16 KBPI:

at the recording density of 16 KBPI we may assume a maximum of 16 effective transitions either side of the point of reference. This leads to 2^{32} possible values of ISI. In addition the neighbouring pulse to the one in question will be only a quarter of the distance from the pulse of interest than is the case for the lower density;

for the lower recording density the ISI will be made up of 4 possible values either side of the timing window so giving a maximum of 2^8 possible points of ISI. In addition these values are equal to the 4th, 8th, 12th and 16th values

for the ISI at 16 KBPI.

6.3.3 The Effect of the choice of Modulation code on ISI

The readout SNR is affected by:

- i) the minimum runlength of 0's between each transition in the Modulation code;
- ii) Code Rate, which effects the size of timing window for a fixed data rate.

Codes with a high minimum runlength and Code Rate tend to have better characteristics in combating ISI as there is a greater separation between transitions. This can be illustrated below by a comparison of Miller and Manchester Modulation codes.

Modulation codes effect the amount of ISI. Pulses which are equidistant from the pulse in question to either combine or cancel with those which are opposite. If both pulses have the same parity they cancel, but if the parity is opposite they will combine. This is a direct result of the symmetry of the assumed Lorentzian pulse.

If a non symmetrical pulse had been used, such as the more accurate description of the replay pulse given by Middleton, Miles and Noyau[6.5], complete combination or cancellation of opposite pulses would not occur. Hence it would be more difficult to obtain a model for the ISI.

Consider first the effect of the structure of the Modulation codes on the level of ISI at very low packing densities. Only the effect of the pulses from the neighbouring

timing windows is measurable as the wider spread pulses give too low a reading. This leads to the following probability densities for the two Modulation codes being examined.

Manchester

$P(\text{ISI from 1 Fundamental distance away}) = 1/3;$

$P(\text{ISI from 2 Fundamental distances away}) = 1/6;$

$P(\text{No ISI assuming symmetry}) = 1/2.$

Miller

$P(\text{ISI from 1 Fundamental distance away}) = 0;$

$P(\text{ISI from 2 Fundamental distances away}) = 1/3;$

$P(\text{ISI from 3 Fundamental distances away}) = 1/6 ;$

$P(\text{ISI from 3 Fundamental distances away interfering with that of 2 Fundamental distances away}) = 1/6;$

$P(\text{No ISI assuming symmetry}) = 1/3.$

Although this is not a complete list of all possible values of ISI for high density recording, it does give an effective measure of how the ISI from the important neighbouring pulses is affected by the choice of modulation code. For a practical data rate and tape speed the effect of ISI is far greater than this simplified model. Note that at typical recording densities ISI from the fundamental distance is 10 times greater than that from three fundamental distance.

From these figures it is easy to see that ISI will be a greater problem in recordings using Manchester Modulation code. This is due to the lower value of the minimum run length consistent with the theoretical work done on the run length dependencies of errors in Modulation codes[6.6].

6.3:4 Correlation of ISI

Noise has been shown to be correlated pulse to pulse for both Modulation codes. Therefore an inspection of the correlation level of ISI between two adjacent pulses was undertaken. The correlation of the values of ISI was measured for the two different modulation codes.

The correlation between the ISI on successive pulse for Miller encoding for the Miller code was found by partitioning the values of ISI into several different sets. This was done by using the runlength between the two adjacent transitions as a marker. Data was thus partitioned into three groups. The correlation was found by using the Minitab[6.7] statistical software package.

For both modulation codes it was found that the ISI was correlated, for a fixed runlength size. The following results were gained for Miller Modulation code and similar results were obtained for Manchester Modulation.

Miller code Correlation of ISI on successive Replay Pulses

Gap of 2 fundamental distance gives a correlation = - 0.126;

Gap of 3 fundamental distance gives a correlation = 0.349;

Gap of 4 fundamental distance gives a correlation = -0.83.

However for both modulation codes when the total amount of ISI was compared the correlation was found to be less than 10% and so could be ignored. This was due to the correlation on the different runlengths cancelling each other out to leave only a small value for the correlation over all runlengths. This suggests that the correlation of ISI may be ignored.

6.4 Modulation Coding Results

6.4:1 Introduction

All modulation codes add redundancy, therefore the error rate may be reduced by careful demodulating of the received pulses. In this case the Miller and Manchester Modulation codes are studied.

6.4:2 A Method of decoding Manchester Modulation Codes.

In Manchester Modulation is that there is a transition in the centre of every data bit cell. This central transition is present in the data frame regardless of the actual data bit. This is designed to give good clocking from the replay pulse.

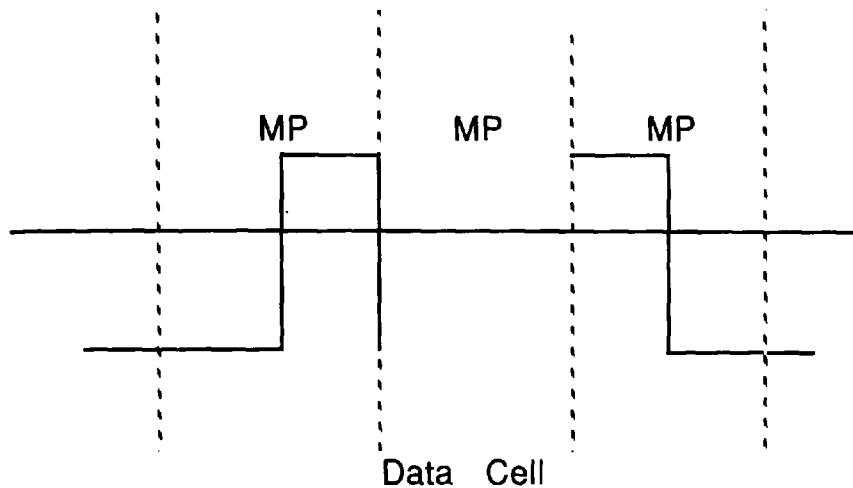


Figure 6.11 Section of Manchester Code Missing a Transition at a Mid Point

In which:

MP is the mid point of a data cell where a transition must be present;

EP is the end point of a data cell where a transition may be present ;

the dotted lines represent the limits of the data cells.

However in an asymmetric channel it is possible, to use a maximum likelihood based decoder, to reconstruct the pulses provided that there is no loss of clocking. In addition in the event of a clockloss error, Manchester code has one of the best abilities to regain clocking.

Maximum Likelihood Decoder with No Clock Loss

In an asymmetric channel, ($1 \rightarrow 0$) it is possible to reconstruct the modulated sequence. If there is no transition in the mid point of a timing window, then the following algorithm is put into effect to determine the missing bit and so to enable the recovery of the data.

- i) check the two neighbouring transitions to the erroneous data bit;
- ii) if one or both of these transitions are non-zero then store the orientation;
- iii) the missing bit will be the opposite of the stored orientation;
- iv) this will enable the mid bit to found as can be shown below.

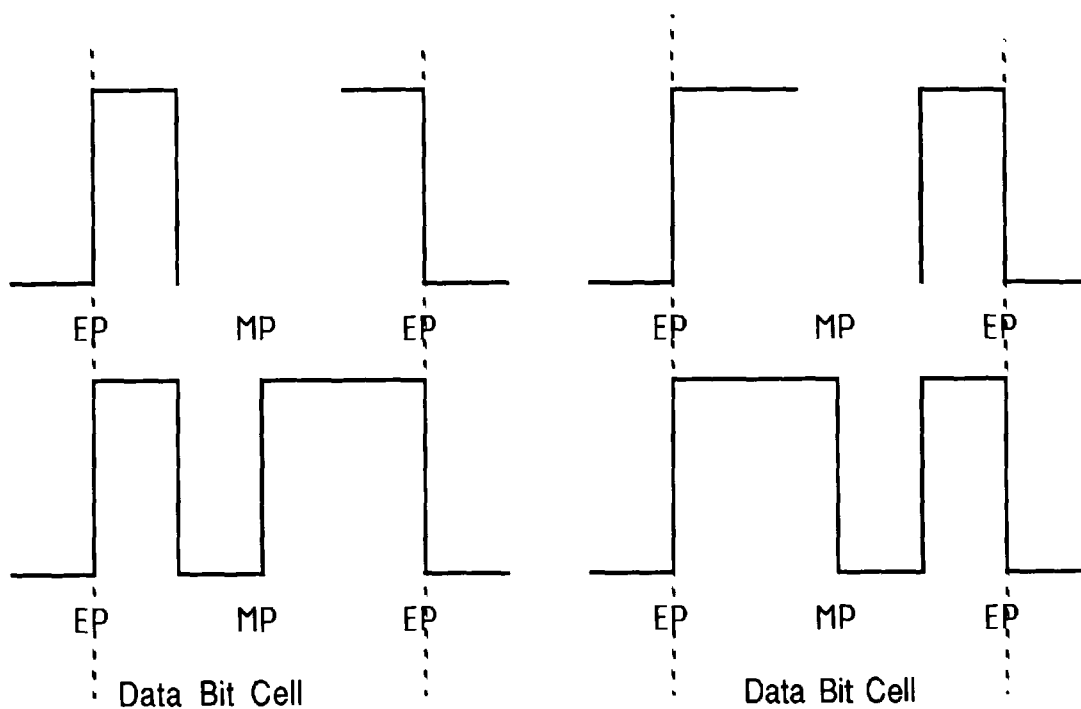


Figure 6.12 Reconstruction of Manchester coding from one error

If the two immediate neighbours are both zero:

- i) inspect the next two transitions;
- ii) if their parity is the same, the missing transition is the opposite.

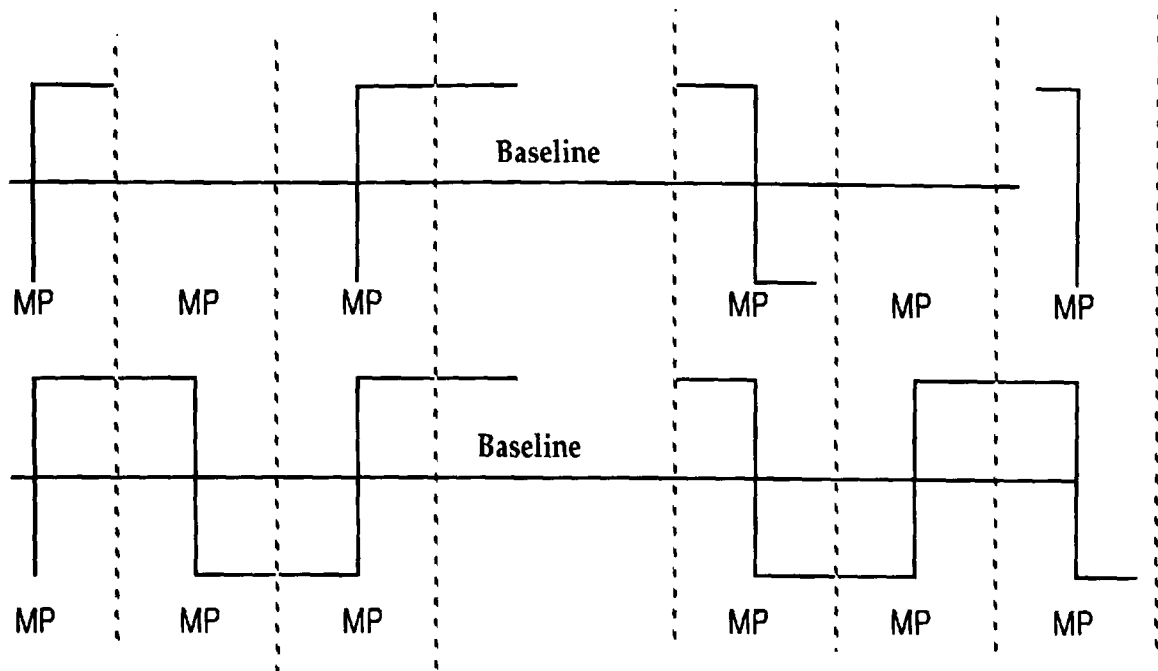


Figure 6. 13 Manchester Recovery with No Adjacent Windows

If there is a difference between the orientation of the two transitions in the centre of the next window then it is impossible to tell the orientation of the missing data bit. As there must have been an additional redundant clocking bit in one of the missing transitions.

This method results in a reduction of the uniform error rate by removing all singleton errors.

6.4:3 Miller Decoding

The encoding algorithm for the Miller code is not as simple as the Manchester code as there is not a single Modulation character for each data cell.

Miller codes represent a data 0 in two different possible ways:

- i) no transition in the data window this is represented as a double blank, 00;

ii) a transition at the end of the data window this can be represented by 01.

The Miller code represents a data 1 as:

a transmission in the centre of the data window.

When a transmission is shifted from its correct transition window no transitions are received in the data cell due to the use of guard bands.

This gives Table 6.2 for the data transmission and reception. Once again a uniform error probability is assumed to enable reasonable figures to be drawn in Table 6.2.

The effect of Autocorrelation on errors is ignored.


		Transmitted Modulation			
			1 0	01	00
R e c e i v e d	10		$1-\epsilon$	0	0
	01		0	$(1-\epsilon)/2$	0
	00		ϵ	$\epsilon/2$	$1/2$

Table 6.2 Probability Table for Miller Code Reception and Transmission

Hence given that a 00 is received it is more likely that a 0 was transmitted than a 1.

$$P(\text{data one}/00) = 2e/(1+3e)$$

where:

e is the original error rate.

This means that the algorithm to decode miller code reduces to:

if there is a transition in the first bit cell then demodulate as a data 1;

if there is no transition in the first bit cell then demodulate as a data 0.

Hence the second bit is redundant and hence we required to know only whether there is a transition in the first timing window of the data cell.

6.5 Error Correcting Codes

6.5:1 Aliasing

In Chapter 4 a perfect code was defined. This is a code whose error syndromes totally fill their vector space: i.e. the code is optimal for the given codeword length and error correction over that certain field. However this does lead to the problem that these codes are unable to distinguish errors of length half their Hamming distance or longer. For these longer errors such codes will attempt to correct for an error having the same syndrome but having been generated by a number of errors which is less than or equal to the maximum correctable length. This will lead to additional errors being generated by the error correcting code and hence to an increase in the error rate.

False error correction is called "aliasing" and may be cured by using a code which corrects more bits in the word. Assuming a known uniform data bit error probability. The amount of redundancy required can be determined by the binomial expansion of the error rate. In this case it is beneficial to use longer codewords which allow a greater percentage of correction for the same percentage of redundancy.

6.5:2 Bit and Byte Error Rates for Non-Linear codes

It has been stated in Chapter 4 that the Reed Solomon code is a byte error correction code. This is a code which will correct a subword of several bits in length if there is a fixed bit error probability, a conversion from bit error rate to byte error rate follows from the binomial expansion.

This figure is given by the following equation:

$$P_{\beta} = 1 - (1 - P_e)^b \quad (6.17)$$

Where:

P_{β} = Byte error rate

P_e = Bit error rate

b = Byte length

From this it is possible to draw conclusions about this byte error rate by altering the conditions upon which it is dependent. These include:

i) subword length for a given bit error probability the byte error probability is dependent upon the length of the code;

ii) bit Error Rate. For a constant subword length the byte error rate is dependent upon the bit error rate.

From the above it is equally possible to make predictions about the byte error rate for any given bit error rate probability provided that the bit error rate is constant.

However as has been shown earlier there is no uniform bit error rate due to the autocorrelation pulse to pulse of noise. However this does give a good estimate for the byte error rate given a bit error rate even with the noise autocorrelation.

6.5:3 Results involving RLL constrained Error Correction

In Section 5.9:4 a set of codes using RLL constraints to trap errors was described[6.8]. Such codes trap and correct certain errors. However at high error rates these codes will lead to error propagation due to bit slip. Bit slip is caused by a successive loss of transitions and may occur in a short dropout or due to a highly correlated noise source. A bit slip error is as severe as a large dropout because it results in a translation of the the received data relative to the input data of one place. Table 6.3 illustrates the (2,4) error detecting code. This code will have runlengths of 5 to 9 which are shown in the first half of the table for a singleton error. The second half of the table shows all the possible runlengths for a double error.

	2	3	4	
2	5	6	7	Single Error Runlengths
3	6	7	8	
4	7	8	9	
5	8	9	10	Double Error Runlengths
6	9	10	11	
7	10	11	12	
8	11	12	13	
9	12	13	14	

Table 6.3 Runlengths for a (1→0) ASC for the Ferreria (2,4) code

As can be seen from the shaded double runlengths in Table 6.3 it is possible to have a double error with the same runlength as a single error this will lead to the original double error being corrected as a single error leading to bitflip.

We now show that the Ferreira codes can be extracted from a single error correcting code to one which will correct double errors. This is achieved by modifying the constraints of the code, the elimination of double errors will resolve the problem of bitflip.

To prevent bitflip it is necessary to be able to distinguish between double errors and single errors in the output of the code. To do this the following inequality has to be satisfied for the minimum, n and the maximum, d runlengths involved in the code.

$$3d+2>2n+1 \quad (6.18)$$

Giving:

$$(3d+1)/2 > n \quad (6.19)$$

This can then be maximised for n with respect to d to give:

$$n=3d/2 \quad (6.20)$$

This then gives an upper bound on n in terms of d for a code which will detect double errors in addition to the single errors in the code given by Ferreira. To lose three transmissions in succession is highly unlikely and to be able to compensate for this would require:

$$4d+3 > 3n+2 \quad (6.21)$$

This then gives the maximum value for n in terms of d as:

$$n = 4d/3 \quad (6.21)$$

This bound on the runlengths is too narrow to be of any use as a modulation code. The information rate for such a code would be almost zero. For example the maximum information rate of a (6,8) code would be 0.1993 of the unconstrained channel [6.4].

	2	3	
2	5	6	Single Error Runlengths
3	6	7	
5	8	9	Double Error Runlengths
6	9	10	
7	10	11	

Table 6.4 Double Error Correcting Runlength Limited Family 1 code.

Table 6.4 shows all the runlengths of a (2,3) code which can correct successive erroneous bits.

This type of code can be extended to include longer runlengths in the same manner as before. An example of this type of code is the (2,3,,12,18) Modulation code which allows a wide variety of runlengths and in addition the ability to detect double errors when they occur.

It is also possible to use constrained modulation codes. In these codes those sequences of runlengths which allow a double error to be decoded as a single one are inadmissible. Here taking the (1,4) code discussed by Ferreira as the third family of codes. Once all those runlengths which lead to error propagation have been removed, there are only 29 possible modulation combinations are allowable, from the original 64.

	1	2	3	4	
1	3	4	5	6	Single Error Runlengths
2	4	5	6	7	
3	5	6	7	8	
4	6	7	8	9	
5	7	8	9	10	Double Error Runlengths
6	8	9	10	11	
7	9	10	11	12	
8	10	11	12	13	
9	11	12	13	14	

Table 6.5 Double Error Correcting Constrained Runlength Family 3 Code

The final method of double error detection using runlength limited sequences is to combine the idea of a $(d, 2d)$ bound on the runlength with the deletion of sequences that would lead to bit-slip error induction. This concatenated code proves to give a high data rate and in addition possesses the ability to detect multiple errors.

An algebraic description of the allowable runlengths is given below.

A pair of consecutive runlengths $a, b \dots (d, 2d)$ are allowable if $a+b+2 > 4d+1$.

This code is illustrated in Table 6.6

	2	3	4	
2	5	6	7	Single Errors
3	6	7	8	
4	7	8	9	
5	8	9	10	Double Errors
6	9	10	11	
7	10	11	12	
8	11	12	13	
9	12	13	14	

Table 6.6 (2,4) code showing Suppressed Runlengths:

In the above code those runlengths which lead to a bit slip error caused by a double error runlength in the shaded area are suppressed. This gives a comparatively wide range of runlengths while preventing the bit slip error from occurring. In this case any elements from the set of the series of runlengths given by,

$$2,2,2: 2,2,3 \ 2,3,2 \ \& \ 3,2,2.$$

Will be prevented from occurring so as to allow a high Information Rate in the channel.

References

- 6.1) E.Katz and T.Campbell** "Effect of Bitshift Distribution on Error Rate in Magnetic Recording" IEEE Transactions on Magnetics Vol MAG-13 No.3 May 1979 pp 1050-1053
- 6.2) A.Ryley and M.K.Loze** "Bitshift Errors due to Wideband Noise in Digital Magnetic Recording" Eight International Conference on Video Audio and Data Recording Birmingham England 24-26 April 1990 pp 15-19
- 6.3) M.K. Loze** "Computer aided Modelling of Digital Magnetic Recording Systems" Ph.d. Thesis 1987 Manchester Polytechnic
- 6.4) K.A.S Immink** " Coding Techniques for Digital Recorders" Prentice Hall International 1991 Chapter 1 p 5
- 6.5) T.T. Doi** "Channel Codes for Digital Audio Recordings" , Presented at the 70th AES Conference NY Oct 30 - Nov 2 1981
- 6.6) B.K.Middleton, J.J. Miles and R.H.Noyau** "Magnetic Properties of Materials, Transition Shapes and Digital Recording Properties" IEEE Trans on Magnetics Vol MAG-26 No.5 Sept 1990 pp 2128-2130
- 6.7) B.L. Ryan B.L.Joiner and T.A. Ryan** "Minitab Handbook Second Edition" Duxbury Press 1985
- 6.8) H.C. Ferreira and S.Lin** "Of Integer Compositions and Error Correcting (d,k) Block Codes"Submitted to IEEE Transactions on Information Theory

Chapter 7

Computer Simulation Design and Implementation

7.1 Introduction

In this chapter an outline of the Monte-Carlo simulation of the recording process is given. These experiments compare and contrast two different types of modulation and of error correcting codes.

The two modulation schemes used are the unmodified Miller code, MFM and Manchester code, FM. The purpose of the experiments is to compare the error propagation of these modulation codes to show how the different run length limitations effect both ISI and the correlation of noise induced bitshift errors.

Two different error correction codes were used: the linear BCH code which encodes the data as a series of single bits, and the nonlinear Reed Solomon code groups the data into subwords. These are then combined to form a long codeword.

The correlation pulse to pulse of the tape noise which was measured earlier in Section 6.2:4 was used to gain a more accurate model of the noise induced bitshift. The effect of the correlated noise was compared to that of the binomial distribution for bitshift errors induced by noise which is independent pulse to pulse. This was done to test the effect on the distribution of replay errors of noise correlation for the different types of modulation and error correction code.

Next the effectiveness of the Ferreira combined modulation and error control codes was measured against an independent noise source. The ability of these codes to combat non clockloss errors was compared to that of the codes with tighter bounds on the runlength limitations. These were discussed in Section 6.5:3. and were designed to prevent a bitflip which is caused by a double error.

7.2 Computer Simulation Design

7.2:1 Wiener filter design of the coefficients of the filter for the Autocorrelation

A theoretical model of the correlation pulse to pulse of the tape noise source was calculated in Section 6.2:4. This gave a more accurate model for the distribution of the noise induced bitshift errors in a codevector. The distribution of bitshift errors in a codevector is an influencing factor in the choice of both modulation and error correction codes. To simulate the effect of noise correlation it is necessary to use a feedback loop for the previous values of the noise source. This gives an output noise source which is correlated from sample to sample as each sample is taken at the peak of the replay pulse then the noise is correlated pulse to pulse. The output is a weighted sum of the previous outputs and the present input. The coefficients of this feed back loop are found by using a Wiener filter.

The feedback loop gives the level of noise autocorrelation over multiples of the fundamental distance. This can be regarded as an optimal nonrecursive estimator for the values of the filter to give the correlation of noise induced bitshift pulse to pulse. A four stage feedback loop was used as a compromise between a more accurate filter with many terms and the simplicity of mathematical calculation of a filter with less terms.

To obtain the necessary feedback loop coefficients to give the correct value of the noise correlation pulse to pulse it is necessary to multiply the inverse of the correlation matrix by the correlation for the noise from the different timing windows.. This tabulates the level of correlation for the successive fundamental distances. This correlation matrix is given by, \mathbf{a} :

$$\mathbf{a} = \begin{bmatrix} 1 & X_i Y_{i-1} & X_i Y_{i-2} & X_i Y_{i-3} & X_i Y_{i-4} \\ X_i Y_{i-1} & 1 & Y_{i-1} Y_{i-2} & Y_{i-1} Y_{i-3} & Y_{i-1} Y_{i-4} \\ X_i Y_{i-2} & Y_{i-1} Y_{i-2} & 1 & Y_{i-2} Y_{i-3} & Y_{i-2} Y_{i-4} \\ X_i Y_{i-3} & Y_{i-1} Y_{i-3} & Y_{i-2} Y_{i-3} & 1 & Y_{i-3} Y_{i-4} \\ X_i Y_{i-4} & Y_{i-1} Y_{i-4} & Y_{i-2} Y_{i-4} & Y_{i-3} Y_{i-4} & 1 \end{bmatrix}$$

and the correlation between the different data windows provides the coefficients of the column vector \mathbf{x} , this is shown below:

$$\mathbf{x} = \begin{bmatrix} X_i Y_i \\ Y_i Y_{i-1} \\ Y_i Y_{i-2} \\ Y_i Y_{i-3} \\ Y_i Y_{i-4} \end{bmatrix}$$

Where:

X_i is the present input to the feedback loop:

Y_i is the present output of the feedback loop;

Y_{i-k} is the output of the the feedback loop k stages previously;

$Y_{i-k} Y_{i-j}$ is the correlation between the output k and j stages previously:

This filter is shown below in Figure 7.1 . The Wiener Filter[7.1,7.2] is used to filter the uncorrelated input noise in a weighted sum with previous outputs, to give a correlated sample. It can be regarded as a recursive shift register. In this case a four stage shift register would be used.

$$a := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -.47537 & -.12668 & .0932 \\ 0 & -.47537 & 1 & -.47537 & -.12668 \\ 0 & -.12668 & -.47537 & 1 & -.47537 \\ 0 & .0932 & -.12668 & -.47537 & 1 \end{bmatrix}$$

$$x := \begin{bmatrix} .86557 \\ -.47537 \\ -.12668 \\ .0932 \\ .00863 \end{bmatrix}$$

$$a^{-1} \cdot x = \begin{bmatrix} 0.866 \\ -0.953 \\ -0.904 \\ -0.601 \\ -0.303 \end{bmatrix} .$$

Figure 7.1 Weiner filter results for autocorrelation of ISI.

These results were validated using the Minitab[7.3] statistical software package to test the correlation of the output levels of the Weiner Filter.

7.2.2 Design of the Model of BCH Error Correcting

BCH codes are linear as they treat the input data as a series of single bits for which the error syndromes may be combined by superposition. This implies that they will correct the error in the code vector wherever it occurs, provided that the total number of errors in any single code vector is less than half the Hamming distance of the code.

As a consequence of this property linear BCH codes which correct a large number of terms have many possible error syndromes. For example the (31,16,3) Error Correcting code has nearly 5000 possible error syndromes which have to be checked to determine the error. To simulate all the different codes of varying Hamming distance is impracticable. The amount of calculation of errors from the given syndrome involved for each code and the computer time it would take would limit the number of codes to be tested very severely. The alternative method of creating a large database of errors and their associated syndromes would have demanded a large amount of computing time to generate.

However it is possible to use the linearity of the BCH code to develop a simpler test for errors. Thus for the (31,15,3) code mentioned above, it is possible to simulate this code by using only 31 information bits. The data is assumed to be derived from a stationary random process, it is then possible to infer that the redundancy in the codeword will also be random. Therefore no actual coding need take place, as the data and redundancy can be generated from the same source.

Worst case patterns are allowed to occur solely due to the randomness of the bit generator. The BCH error correction codes do not generate more worse case patterns than those which are already present in the data. This is true for all BCH codes with the exception of those codes with a very high correcting rate such as the (7,1,3) code. This represents data as either a string of seven encoded 1's for a data 1 and a string of seven encoded 0's for a data 0. This require so much redundancy and have such a poor information rate that they are of little use in a practical case.

To best understand the methods used for the simulation a full description of the algorithm will be given:

- 1 generate a series of random binary digits. Copy this series and put a copy into memory for later comparison;
- 2 modulate these digits ;
- 3 add error through a Gaussian noise channel. This can be either correlated or uncorrelated pulse to pulse In addition the error probability can be increased by convolving the pdf of the noise induced bitshift with that for the ISI induced bitshift;
- 4 demodulate the resulting stream of replay pulses;
- 5 compare the continuous demodulated stream from part 4 with the original in the computer memory. Select all the points at which the received and generated streams differ, mark these as a 1 in the error stream and the correct bits as a zero in the error stream;

- 6 break the continuous error stream into words of the correct length for the given code pattern. This enables the errors to be assessed against the varying length of the codewords;
- 7 count the total number of errors and the number which occur during the transmission of each code vector. This value for the number of bit errors in a word is set into an array which is governed by the amount of errors which occur in each codeword. So for the example at the end of the data stream array[x] will contain the number of codewords with x errors in;
- 8 print out the array generated above to give the number of words with that amount of errors in;
- 9 the effects of BCH error correcting code can be found by the errors which will be corrected by codes of different Hamming distances. The effect of additional data corruption due to aliasing can be shown. This as stated in Section 6.5:2 effects the code by falsely correcting bits which are not in error in the data sequence.

The above model gives a simple though thorough method of simulating the effect of BCH code and the two Modulation codes. It has been shown earlier that complete simulation of the encoding/ decoding process for long BCH codewords with a large amount of redundancy take a large amount of computer time as well as involving a large number of complex manipulations. The lack of any need to use computer time to divide the data sequence by the polynomial brings an improvement in the run time of the programme..

A principal benefit of this algorithm is the flexibility in the length of correctable errors. As has been shown earlier a code which can correct only three errors confronted with a piece of data with four errors in it, will 'correct' those three bits which when in error give the same syndrome as the four bits in error. This leads to additional errors due to aliasing; these may be found by adding the number of bits the code can correct to the original number of error bits. A code which can correct 3 bits in error which has 5 bits in error in one codevector would result in an error of 8 bits in the codevector.

7.2:3 Design of the model of Reed Solomon Error Correction coding

Unlike the BCH codes in Section 7.2:2 the Reed Solomon codes are not truly linear. They consist of a series of data blocks called subwords which are taken from an extended binary Galois field. These subwords can be of varying length which is dependent upon the length of the codeword. Though for the 256 subword Reed Solomon code used in high integrity recording there is a direct analogy between the eight bit computer subword and the subword in the code. These codes were dealt with in section 4.4:7. These codes correct an entire block regardless of the number of errors within each block so a Reed Solomon code would not distinguish between a block in which every bit is in error or a block with just a single error.

This lack of a linear structure for the code was initially found to be a major drawback to the applying the simulation techniques used for the BCH code. This lack of Monte-Carlo simulation techniques implied the use of a full encoder/decoder representation for the Reed Solomon code would be appropriate.

This full encoder/decoder process is limited for the use of one code of known redundancy for each run, as well as using complex binary manipulation to generate the parity check blocks and the error syndromes.

However the data subwords are separated from the demodulated bitstream and each subword is individually checked for error. If there is one error or more in a block then the block is signalled as being in error. Then there is no need for the complex binary operations which would have been necessary for the other method of simulation. In this the Reed Solomon code can then be regarded as an extension of the BCH code. Where the BCH code can be regarded as a Reed Solomon code of block length 1, i.e. one whose subwords come from the Galois field of $GF(2)$, therefore the Reed Solomon code is linear over the Galois field from which the subwords are drawn.

The total number of bit errors in the data stream is not of primary importance for the Reed Solomon correction code as much as the distribution of the errors, as the code relies on data subwords in error rather than individual bits. Though it is put in store for the calculation of the post Modulation bit error rate. The simulation process is a continuation of the work in Section 7.2.2 on BCH codes.

The Reed Solomon code can therefore be regarded as being linear over the extended Galois Field over which the data subwords were defined. The algorithm used to simulate the Reed Solomon code is given below:

- 1 generate a series of random binary digits. Copy this series, break this into a series of subwords and put a copy into memory for comparison later on;

2 modulate these digits according to the type of modulation code being used;

3 add error via a noise channel¹. The distribution or number of errors can be altered by using either correlated noise from the neighbouring digits or by convolving the noise induced bitshift with ISI induced bitshift;

4 demodulate the data stream received through the channel;

5 break the demodulated data stream into a series of data subwords. Check each subword for error against the original subwords;

6 count the number of subwords. When the correct number has been reached for a codeword count the number of subwords in error. The number of subwords in error acts as a index to an array. This array stores the number of codewords with that amount of subwords in error;

7 print out the array to give the number of codewords with that amount of subwords in error in;

The method in the algorithm is a corollary to the method used for the BCH code where the errors were given as individual bits of data.

7.2:4 Design of the Model for Noise and ISI

7.2:4i) Model for Independent Noise

The noise model used was that given by Katz and Campbell[7.4]. This assumes that noise induced bitshift has a Gaussian PDF. This can be developed by using the NAG (National Algorithms Group)(7.5) model for a Random Variable of Gaussian PDF. This standard function may be called from within the programme and can be combined with other functions.

The reason for the choice of the Katz and Campbell Noise PDF for this model as opposed to the PDF given by Ryley and Loze[7.6], ~~as~~ the recent results showing that the combination of a more accurate Non-Lorenzian pulse, such as that given by Middleton Miles and Noyau[7.7] and the Ryley and Loze model of the PDF of Noise Induced Bitshift are very similar to those given by the original Katz and Campbell PDF for Noise induced bitshift.

7.2:4 ii) Model for Correlated Noise

An additional reason for using a zero mean Gaussian model for the noise induced bitshift is the ability to add correlation between successive sampled noise pulses. The addition of two or more zero mean Gaussian noise sources result in a zero mean Gaussian noise output. This in addition to the values of the feedback coefficients found by the use of the Wiener Filter in Section 7.2:1, allowed the experiments on noise correlation to be undertaken.

7.2:4 iii) Model for ISI

The effect of ISI as an error mechanism lies in its convolution with noise to induce bitshift errors as has been discussed in Section 2.4:2. It is possible however to measure the level of bitshift which is induced solely from ISI. The statistical properties of the effect of the ISI are those of the mechanism itself. To change from an ISI to the induced bitshift one simply divides the ISI by the gradient of the derivative of the replay pulse at the zero crossing.

ISI is dependent upon a series of pulses on either side of the pulse under observation. However this is very difficult to model for each pulse in the data stream as the model requires a detailed knowledge of the series of pulses either side of the pulse in question. However assuming that ISI can be regarded as a Random Variable it is possible to derive a model for ISI.

To get the correct value of ISI induced bitshift an algorithm was used, in which the value of a random variable corresponds to the cumulative PDF for the ISI induced bitshift. This gives a single value of ISI induced bitshift which is then added onto the value of noise induced bitshift to give the combined value of bitshift:

- 1 read into an array the individual values of the ISI and the cumulative probability;
- 2 generate the random data as done in the earlier algorithms for the noise only error source, complete the Modulation and generate a value of noise induced bitshift;

- 3 use a random number generator to obtain a number between zero and one. Select the value of ISI whose cumulative probability is immediately below the random number;
- 4 add the value of the ISI induced bitshift onto the Noise induced bitshift, to give the combined induced bitshift. If the combined value is outside the limits of the timing window signal a bitshift error by removing the transition from the window;
- 5 demodulation and error monitoring is done as in the above algorithms for Noise only error sources.

However this procedure does take a large amount of computer time due to the search of all possible cumulative probabilities. The selection of the value of ISI induced bitshift can be simplified provided that the probability of all the different values of ISI is approximately uniform. This is true for Manchester Modulation code where the probability of ISI is uniform due to the code structure.

The selection of the value of ISI is done by multiplying the random number by the number of values of ISI in the array and selecting that value. This leads to a simple and quick approach for accessing the value of the ISI than via the above with no loss of accuracy. This same idea can be applied to ISI in different Modulation schemes including Miller code with only a negligible loss of accuracy.

7.2:5 Improvement in Error Rates due to demodulation techniques

As has been shown in Section 6.4 it is possible to Improve upon the error rate at the receiver part of the channel by careful demodulation of the received information, this can be seen in the results of the experiments.

The first set of results in Table 7.2 a & b below are for a set of data with Manchester Modulation without using the modification given in section 6.4:1. The second pair of tables, Table 7.3 a & b refer to data encoded with Manchester Modulation with a check back to the previous two transmissions. This is undertaken when there is no transition at the mid point of a data bit cell. This method finds the most likely transition for the data cell.

The code used is a BCH code of length 63, the bit error rate was originally set at 1/100 for the unmodified modulation code. The noise error source used was varied from the uncorrelated noise in tables 7.2a and 7.3a to correlated noise in 7.2b and 7.3b. This was done to test the effect of noise correlation upon the amount of errors recovered by the coding scheme.

Both sets of tables can be compared with the theoretical results given by the binomial expansion, given that the error rate for the modified Manchester should be the square of the error rate for the unmodified code. This figure is arrived at as to cause an error in the modified Manchester coding it is necessary to have two bits in error in successive transitions which for an uncorrelated noise source is $P(\epsilon) \cdot P(\epsilon) = P(\epsilon)^2$, where $P(\epsilon)$ is the probability of error.

Total number of transmitted errors	6344
Transmitted error rate	0.00512
<hr/>	
Number of words with no errors is	14271
Number of words with one error is	4555
Number of words with two errors is	749
Number of words with three errors is	90
Number of words with four errors is	4
Number of words with five errors is	1

Table 7.2 a Uncorrelated Noise with no Look -up Table

Total number of transmitted errors	10014
Transmitted error rate	0.00707
<hr/>	
Number of words with no errors is	11740
Number of words with one error is	5991
Number of words with two errors is	1567
Number of words with three errors is	249
Number of words with four errors is	25
Number of words with five errors is	6
Number of words with six errors is	2

Table 7.2 b Correlated Noise with no Look -Up Table

total number of transmitted errors	65
Transmitted error rate	0.000052426
<hr/>	
Number of words with no errors is	19616
Number of words with one error is	63
Number of words with two errors is	1

Table 7.3 a Uncorrelated Noise Error Source with a Look-Up Table

total number of transmitted errors	165
Transmitted error rate	0.000133071
<hr/>	
Number of words with no errors is	19517
Number of words with one error is	19
Number of words with two errors is	3

Table 7.3 b Correlated Noise Error Source with a Look-Up Table

To be able to estimate the effect of noise correlation on the Manchester Modulation code the coding gain was calculated for each of the two codes.

The first for the uncorrelated noise in Tables 7.2a and 7.3a :

Original error rate: 0.00512

Improved error rate : 0.0000543

The error rate improvement is 94.36 for the correction

To obtain the coding gain it is first necessary to convert the error rate to decibels, for the above:

Original SNR = 29 dB.

Improved SNR = 32 dB

Coding Gain = 3 dB

To compare this with the Correlated noise in Tables 7.2a and 7.2b gives :

Original Error rate : 0.00707

Improved Error Rate: 0.000133

The Error Rate improvement is : 61.64

Obtaining the coding gain by converting the error rates to SNR gives:

Original SNR = 19.6 dB.

Improved SNR = 21.5 dB

Coding Gain = 1.9 dB

These figures for the improvement given by the coding gain are not outstanding., This is due to a high initial error rate, a more realistic result with a higher error rate of approximately 10^{-5} would yield better results.

The figures for the coding gain in decibels come from the graph for Katz and Campbell's PDF for noise induced errors in Chapter 6. The level of improvement is dependent upon the original error rate in addition to the level of autocorrelation. The improvement in the error rate is for uncorrelated noise theoretically should be the reciprocal of the error rate.

7.3 Results with BCH Error Control Coding

7.3:1 Uncorrelated noise with no ISI

This model was run as a check for the error rate and the program the results with the binomial expansion. Several different lengths of code were tested, codes of 31, 63 and 127 bits in length. Several different bit error rates were ranging from 0.05 to 0.0005 were used. This gives a nominal range in decibels of 23 to 29 dB from Figure 6.2.

To compare these results with the Binomial expansion a one-sided chi-squared test, the degrees of freedom were fixed by the codeword with the largest number of errors in each set used, this gave the following set of results:

Bit Error Rate	Correlation	Chi Square	DF	Significance Level
0.05	0.998	5.425	7	66%
0.005	1.0	0.061	3	99.5%
0.0005	1.0	0.040	2	98%

Table 7.4 Uncorrelated Noise and No ISI

From Table 7.4 it can be seen that the conclusion that the noise induced errors can be represented by a binomial distribution. The accuracy of the error model is excellent.

In addition to using the Miller code for Modulation the Manchester code was used with a two bit Look back and ahead for replacing lost transitions which had been

shifted outside the timing window. This gave similar results for the significance levels.

7.3.2 Correlated noise with no ISI

The correlation feedback loop for the noise source was then added to the program while using both Manchester and Miller Modulation codes. No ISI was added initially to test the choice of Modulation code against the effect of noise autocorrelation.

Differing BCH code lengths were also used to test the hypothesis that Error Correction codes which employ a shorter code vector are more prone to noise correlation than the longer codes. This is due to the correlation which is significant over a short length of data, five transitions becoming less noticeable in a long stream of data. This can be shown below, in the two sets of data. The first is for BCH coding of length 7 the second of length 255.

The hypothesis is that a shorter word length would be affected more by the noise correlation than would a word of greater length. For a modulated word of length 14 modulation bits the majority of bits in the word would be within the range of the correlation of the central bits.

The results for both the run of the 7 data bit word and the 255 data bit word were compared using the Chi-squared test on Minitab[7.3] to the binomial expansion. To compare these results to those for uncorrelated noise gave the effect of the noise autocorrelation on the error rates. The results are given below:

Miller Modulation

	Correlated noise	Uncorrelated noise
Number of words with no errors is	170996	177096
Number of words with one error is	6062	6061
Number of words with two errors is	92	92
Number of words with three errors is		1

Table 7.4a Effect of Correlated Noise Short Data Length Miller Modulation

	Correlated noise	Uncorrelated noise
Number of words with no errors is	1356	1327
Number of words with one error is	1691	1729
Number of words with two errors is	1125	1123
Number of words with three errors is	494	475
Number of words with four errors is	143	155
Number of words with five errors is	44	40
Number of words with six errors is	13	9
Number of words with seven errors is	3	2
Number of words with eight errors is	1	

Table 7.4b Effect of Correlated Noise Long Data Length Miller Modulation

Manchester Modulation

	Correlated noise	Uncorrelated noise
Number of words with no errors is	167537	167425
Number of words with one error is	9359	9472
Number of words with two errors is	240	230
Number of words with three errors is	4	3

Table 7.5a Effect of Correlated Noise Short Data Length Manchester Modulation

	Correlated noise	Uncorrelated noise
Number of words with no errors is	604	637
Number of words with one error is	1243	1302
Number of words with two errors is	1373	1323
Number of words with three errors is	795	793
Number of words with four errors is	463	450
Number of words with five errors is	179	170
Number of words with six errors is	57	61
Number of words with seven errors is	21	17
Number of words with eight errors is	4	4

Table 7.5b Effect of Correlated Noise Long DataLength Manchester Modulation

These sets of data were now compared using a chi-squared test to see if the noise Autocorrelation gave a significant effect on the amount of errors over the codeword given the same error probability. This gives the following set of results:

For Miller Modulation:

The code vector of length 7

Chi squared test value of 1.027,

two degrees of freedom

these results give a value of 60% for the level of significance that the two streams of data came from the same source.

The code vector of length 255

Chi squared test value of 3.443

six degrees of freedom

These results give a value of 75% for the significance level that the two streams of data came from the same source.

For Manchester Modulation:

The code vector of length 7 ;

Chi squared test value of 1.196

two degrees of freedom

These results give a value of 55% for the significance level that the two streams of data came from the same source.

The code Vector of length 255;

Chi squared test value of 3.636

six degrees of freedom

These results gives a value of 72% for the significance level that the two streams of data came from the same source.

Conclusions

These results bear out the hypothesis that the shorter the code vector the greater the effect of the noise correlation. In addition it can be shown from these results that the choice of Modulation code is not a major factor in the effect of the noise correlation on the error distribution.

The error distribution for both Modulation codes subject to correlated noise, are not significantly different from those of the binomial expansion for the uncorrelated noise. Hence all further experiments in this thesis using the BCH code need only use the model for uncorrelated noise. Further there is no correlation between successive values of the ISI induced bitshift. Hence it is established that the convolved error probability of noise and ISI pulse to pulse is

independent.

7.3:3 Effect of ISI on BCH encoded sequences

The model used for ISI was that given in section 7.2:4. Simulation runs were made to compare the effect of ISI on error rate for the two modulation codes discussed in Section 5.4. These codes have been shown in Section 5.4 to have different runlength characteristics.

Miller Modulation code

	Uncorrelated noise	Uncorrelated noise and ISI
Number of words with no errors is	33396	2961
Number of words with one error is	5202	7337
Number of words with two errors is	392	10654
Number of words with three errors is	19	7736
Number of words with four errors is	1	5422
Number of words with five errors is		2572
Number of words with six errors is		905
Number of words with seven errors is		302
Number of words with eight errors is		73
Number of words with nine errors is		19
Number of words with ten errors is		7

Table 7.6a Effect of the addition of ISI to Miller Modulation 31 bits

Manchester Modulation code

	Uncorrelated noise	Uncorrelated noise and ISI
Number of words with no errors is	17081	4792
Number of words with one error is	2704	6958
Number of words with two errors is	204	4884
Number of words with three errors is	10	2271
Number of words with four errors is	1	767
Number of words with five errors is		177
Number of words with six errors is		42
Number of words with seven errors is		8
Number of words with eight errors is		0
Number of words with nine errors is		1

Table 7.6b Effect of the addition of ISI to Manchester Modulation 31 bits

Table 7.6a and Table 7.6b confirm that for both Modulation codes the addition of ISI to the noise leads to a substantial degradation in the error rate. This is entirely expected as ISI is the dominant error source. This then leads to the need for extra redundancy in the error correction coding to be able to handle these additional errors.

However from these tables it can be seen that the amount of degradation in the error rate due to the defect of ISI is similar in both cases. One would expect the amount of degradation due to ISI to be worse in the Manchester Modulation

Manchester has a lower minimum runlength constraint than Miller. The explanation for the lack of this effect is the assumption of a symmetric replay pulse[7.8].

7.4 Results involving Reed Solomon Error Correction Encoding

7.4:1 Noise only, no ISI

In Section 4.5 Reed Solomon coding was assumed to be an extension of BCH coding. So for the Galois field $GF(2^5)$ the code streams are all of length 31, there are 31 non zero elements in $GF(2^5)$. The results for this Reed Solomon code are shown in Table 7.7 :

Codes of length 31 subwords with Miller Modulation

Bit Error Rate	0.05	0.005	0.0005	0.00005
Subword Error Rate	0.226	0.025	0.0025	0.00025
Words With 0 errors	0	4616	92529	992163
Words With 1 error	4	3595	7214	7706
Words With 2 errors	10	1361	251	31
Words With 3 errors	39	360	6	
Words With 4 errors	76	59		
Words With 5 errors	131	6		
Words With 6 errors	147	2		
Words With 7 errors	179			
Words With 7 errors	165			
Words With 9 errors	99			
Words With 10 errors	79			
Words With 11 errors	39			
Words With 12 errors	20			
Words With 13 errors	7			
Words With 14 errors	4			
Number of words	1000	10000	100000	1000000

Table 7.7 Reed Solomon Encoding for Noise Only 31 five Bit words

Table 7.7 can be compared with Table 7.8 below shows the error distribution for 127 seven bit Reed Solomon words, i.e. a codevector of 889 bits. However the error rate of 0.05 was too high to enable any of the results to be tabulated as the average number of subwords in error was 39.

Codes of length 127 subwords with Miller Modulation

Bit Error Rate	0.05	0.005	0.0005	0.00005
Subword Error Rate	0.301	0.0345	0.003495	0.00035
Words With 0 errors		111	63950	956373
Words With 1 error		497	27650	42695
Words With 2 errors		1101	6290	907
Words With 3 errors		1716	1003	13
Words With 4 errors		1940	97	1
Words With 5 errors		1717	10	
Words With 6 errors		1299		
Words With 7 errors		749		
Words With 7 errors		436		
Words With 9 errors		191		
Words With 10 errors		94		
Words With 11 errors		33		
Words With 12 errors		11		
Words With 13 errors		4		

Table 7.8 Reed Solomon for Noise Error source 127 Seven bit subwords

This can be compared to the binomial distribution. Let the probability of a bit being in error be, P_{ϵ} . Let the subword error probability, be P_{β} . The length of a subword to be, b . These three elements can be related by the formula below:

$$P_{\beta} = 1 - (1 - P_{\epsilon})^b$$

Using this formula for the subword error rate the binomial distribution of subword errors for the 31 and 127 subword Reed Solomon codes can be found. These are then tabulated in tables 7.9 and 7.10.

Binomial Expansion using the Subword Error Rate for 31 Subword code

Bit Error Rate	0.05	0.005	0.0005	0.00005
Subword Error Rate	0.226	0.025	0.0025	0.00025
Words With 0 errors	0	4597	92541	992270
Words With 1 error	3	3617	7173	7662
Words With 2 errors	14	1377	270	29
Words With 3 errors	40	337	7	
Words With 4 errors	71	60		
Words With 5 errors	127	7		
Words With 6 errors	162			
Words With 7 errors	169			
Words With 7 errors	147			
Words With 9 errors	111			
Words With 10 errors	71			
Words With 11 errors	40			
Words With 12 errors	19			
Words With 13 errors	7			
Words With 14 errors	3			
Words With 15 errors	1			
Number of words	1000	10000	100000	1000000

Table 7.11 Binomial Error Distribution for 31; five, bit data words

However for the longer 127 subword Reed Solomon code with high bit error rate of 0.05 is impractical to tabulate. Due to the subword error rate of 0.301 which when modelled gave an average of 39 subwords in error in each of the codewords. This far exceeds the maximum for all the other error rates.

Bit Error Rate 0.05	0.05	0.005	0.0005	0.00005
Subword Error Rate	0.301	0.0345	0.003495	0.00035
Words With 0 errors		116	64106	956516
Words With 1 error		525	27554	42532
Words With 2 errors		1173	6309	937
Words With 3 errors		1761	922	14
Words With 4 errors		1950	100	
Words With 5 errors		1714	9	
Words With 6 errors		1246	1	
Words With 7 errors		769		
Words With 7 errors		412		
Words With 9 errors		195		
Words With 10 errors		72		
Words With 11 errors		31		
Words With 12 errors		11		
Words With 13 errors		3		
Words With 14 errors		1		

Table 7.10 Binomial Error Distribution for 127 Seven bit data words

From these results it is possible to assess that the error correction potential of the Reed Solomon code. When combating uncorrelated noise it is less effective than the linear BCH code.

Correlated Noise

Simulation is now used to model the effectiveness of the Reed Solomon code in combating noise which is correlated pulse to pulse. These results are shown in Table 7.11 and Table 7.12.

Codes of length 31 subwords with Miller Modulation

Bit Error Rate 0.05	0.05	0.005	0.0005
Subword Error Rate	0.226	0.025	0.0025
Words With 0 errors	7	704	19221
Words With 1 error	42	1492	17013
Words With 2 errors	67	1273	7461
Words With 3 errors	72	737	2603
Words With 4 errors	73	335	600
Words With 5 errors	77	106	77
Words With 6 errors	50	36	14
Words With 7 errors	29	5	1
Words With 8 errors	24	1	
Words With 9 errors	7	1	
Words With 10 errors	7		
Words With 11 errors	2		
Words With 12 errors	1		

Table 7.11 Reed Solomon Encoding for Correlated Noise Only; 31, five Bit words

Table 7.11 can be compared with Table 7.12 for 127 seven bit subwords.

Bit Error Rate	0.05	0.005	0.0005	0.00005
Subword Error Rate	0.301	0.0345	0.003495	0.00035
Words With 0 errors	14	1612	54120	
Words With 1 error	46	4070	71520	
Words With 2 errors	123	5246	47079	
Words With 3 errors	221	4347	19970	
Words With 4 errors	270	2732	6270	
Words With 5 errors	307	1317	1527	
Words With 6 errors	301	516	306	
Words With 7 errors	290	176	60	
Words With 7 errors	173	51	5	
Words With 9 errors	116	16	2	
Words With 10 errors	76	4		
Words With 11 errors	33	1		
Words With 12 errors	20			
Words With 13 errors	7			
Words With 14 errors	2			
Words With 15 errors	1			

Table 7.12 Reed Solomon for Correlated Noise Error source: 127, Seven bit subwords

Because of the subword structure of the Reed Solomon code it is expected that the performance of the Reed Solomon code is better than the BCH code in combating Correlated noise. In fact for the same bit error rate there should be an improvement in the subword error rate for Correlated noise. The Tables 7.11 and 7.12 confirm this but suggest that the improvement is too small to be significant.

7.4:2 Effect of the addition of ISI to Reed Solomon Error Correction Codes

Inevitably the addition of ISI into the system will lead to an increase in the Error Rate. The simulation results which demonstrate this can be shown in Tables 7.13 and 7.14.

Bit Error Rate 0.05	0.05	0.005	0.0005	0.00005
Subword Error Rate	0.226	0.025	0.0025	0.00025
Words With 0 errors	0	4616	92529	992163
Words With 1 error	4	3595	7214	7706
Words With 2 errors	10	1391	251	31
Words With 3 errors	39	360	6	
Words With 4 errors	76	59		
Words With 5 errors	131	6		
Words With 6 errors	147	2		
Words With 7 errors	179	1		
Words With 7 errors	165			
Words With 9 errors	99			
Words With 10 errors	79			
Words With 11 errors	39			
Words With 12 errors	20			
Words With 13 errors	7			
Words With 14 errors	4			
Total Number of Bit Errors	45301	44750	44557	

Table 7.13 Noise and ISI effect on Reed Solomon codes 31 five bit Subwords

Bit Error Rate	0.05	0.005	0.0005	0.00005
Subword Error Rate	0.301	0.0345	0.003495	0.00035
Words With 0 errors	111	63950	956373	
Words With 1 error	497	27650	42695	
Words With 2 errors	1101	6290	907	
Words With 3 errors	1716	1003	13	
Words With 4 errors	1940	97	1	
Words With 5 errors	1717	10		
Words With 6 errors	1299			
Words With 7 errors	749			
Words With 8 errors	436			
Words With 9 errors	191			
Words With 10 errors	94			
Words With 11 errors	33			
Words With 12 errors	11			
Words With 13 errors	4			
Total Number of Bit Errors	45301	44750	44557	

7.14 Noise and ISI on Reed Solomon Coding with 127 7 bit words

From Tables 7.13 and 7.14 it is possible to see that Reed Solomon code is not effective against the combination of the two non clockloss error mechanisms of Noise and ISI.

7.4:3 The use of Clock based recovery in Manchester Modulation code and its application to Reed Solomon coding.

It has been shown in section 6.4 that at low recording densities, provided there is no clock loss. It is possible to recover the majority of the lost signal from the additional transitions used in Manchester Modulation.

This is done by referencing the series of transitions to the clock, with the knowledge that there has to be a transitions in the middle of every data window. Then a transition can be generated if there is no transition in the centre of that window, by reference to the transitions either side.

The results that were obtained from combining Manchester Modulation and Reed Solomon code with an independent Gaussian noise source are tabulated in Table 7.15. In this simulation the bit retrieval system has been used to replace lost transitions from the centre of the data window. This should lead to a reduction in the initial bit error rate. These results have been tabulated in Table 7.15 and can be compared to those results given for the Reed Solomon Code with no look ahead and back to assist in reducing the error rate. The data retrieval results overleaf can be compared to the code length of 127 subwords without this method of retrieval in Table 7.8 above.

Initial Bit Error Rate	0.05	0.005	0.0005	0.00005
Initial Subword Error Rate	0.301	0.0345	0.003495	0.00035
Words With 0 errors	11	22317	377347	3976710
Words With 1 error	75	12962	21977	23126
Words With 2 errors	240	3760	659	64
Words With 3 errors	433	742	6	
Words With 4 errors	627	107		
Words With 5 errors	696	11		
Words With 6 errors	611			
Words With 7 errors	503			
Words With 7 errors	327			
Words With 9 errors	237			
Words With 10 errors	106			
Words With 11 errors	69			
Words With 12 errors	29			
Words With 13 errors	16			
Words With 14 errors	7			

Table 7.15 Manchester Modulation and Reed Solomon coding

Table 7. 15 suggests that there is a marked improvement in the error rate for Reed Solomon code when it is combined with the Manchester Modulation transition recovery. The error rate is reduced to give a subword error rate for the 127 word Reed Solomon code which is shown in Table 7.16.

Original Byte Error Rate	New Byte Error Rate	Improvement
0.3017	0.04382	688%
0.0345	0.00461	748%
0.00345	0.000459	752%
0.00035	0.0000458	764%

Table 7.16 Subword error rate with Look ahead Manchester decoding.

As can be seen from Table 7.16 the level of improvement in the Reed Solomon code subword error rate is nearly constant. This is unexpected as the probability of a double is proportional to the square of the error rate. We may then expect the improvement to be proportional to the error rate.

7.5 Results involving Ferreira Codes

7.5:1 Bitflip

In Section 6.5 the problem of bitflip in the Ferreira[7.9] codes was discussed. This is caused by the ~~attempted~~ correction of the loss of a double transition as the loss of a single transition. This leads to error propagation as the position of the received transitions no longer match to those at the input to the channel. This can be illustrated by Table 7.17 where the presence of bitflip in the recording is tabulated against the bit error rate.

Error Rate	Ferreira	Derivative Code
0.01	Yes	No
0.001	No	No
0.0001	No	No
0.00001	No	No

Table 7.17 Bitflip for Ferreira Integer Composite Codes

The effect of Bitflip on data recording is illustrated by Tables 7.18 and 7.19. Table 7.19 is for the Ferreira code with a bit error rate of 0.01 it exhibits the characteristic effects of a bitflip error, an average of 50% error rate for the bitflip. Table 7.19 is for

the Ferreira code with a bit error rate of 0.001 it exhibits none of the characteristic effects of a bit slip error.

So to conclude, bit slip errors occur in Ferreira codes at comparatively high Error Rates which will not occur in real data recording equipment. In these cases the Error rates range from 10^{-5} to 10^{-10} which implies that the probability of a bit slip error in an independent noise channel ranges from 10^{-10} to 10^{-20} , so the probability of Bit slip error in a Noise channel is negligible. It is not however possible to draw the same conclusions about correlated noise and channels with prevalent Clock loss error mechanisms.

One of the possible explanations for the presence of such high bit slip is that the code used was a fixed pulse code not a fixed length code. These codes have a fixed number of transitions rather than a fixed length of modulation code to represent the data. However, it is possible that bit slip may be inherent in Ferreira codes due to the method of error detection employed by the code.


```

number of generated errors          3172
TOTAL NUMBER OF WORDS IS          248000
total number of transmitted errors      8
number of data streams with          0 errors in is 247355
number of data streams with          1 errors in is   645
number of data streams with          2 errors in is    0
number of data streams with          3 errors in is    0
number of data streams with          4 errors in is    0
number of data streams with          5 errors in is    0
number of data streams with          6 errors in is    0
number of data streams with          7 errors in is    0
number of data streams with          8 errors in is    0
number of data streams with          9 errors in is    0
number of data streams with         10 errors in is    0
number of data streams with         11 errors in is    0
number of data streams with         12 errors in is    0
number of data streams with         13 errors in is    0
number of data streams with         14 errors in is    0
number of data streams with         15 errors in is    0
number of data streams with         16 errors in is    0
number of data streams with         17 errors in is    0
number of data streams with         18 errors in is    0
number of data streams with         19 errors in is    0
number of data streams with         20 errors in is    0
number of data streams with         21 errors in is    0
number of data streams with         22 errors in is    0
number of data streams with         23 errors in is    0
number of data streams with         24 errors in is    0
number of data streams with         25 errors in is    0
number of data streams with         26 errors in is    0
number of data streams with         27 errors in is    0
number of data streams with         28 errors in is    0
number of data streams with         29 errors in is    0
number of data streams with         30 errors in is    0
number of data streams with         31 errors in is    0
DAMYLES      job terminated at 14-FEB-1992 09:05:41.96

```

Table 7.18 Ferreira Code no Bitflip

TOTAL NUMBER OF WORDS IS		24800
total number of transmitted errors		2491151
number of data streams with	0 errors in is	25
number of data streams with	1 errors in is	0
number of data streams with	2 errors in is	3
number of data streams with	3 errors in is	0
number of data streams with	4 errors in is	0
number of data streams with	5 errors in is	0
number of data streams with	6 errors in is	0
number of data streams with	7 errors in is	0
number of data streams with	8 errors in is	0
number of data streams with	9 errors in is	0
number of data streams with	10 errors in is	1
number of data streams with	11 errors in is	1
number of data streams with	12 errors in is	5
number of data streams with	13 errors in is	6
number of data streams with	14 errors in is	22
number of data streams with	15 errors in is	52
number of data streams with	16 errors in is	122
number of data streams with	17 errors in is	193
number of data streams with	18 errors in is	403
number of data streams with	19 errors in is	664
number of data streams with	20 errors in is	1057
number of data streams with	21 errors in is	1419
number of data streams with	22 errors in is	1961
number of data streams with	23 errors in is	2356
number of data streams with	24 errors in is	2682
number of data streams with	25 errors in is	2736
number of data streams with	26 errors in is	2653
number of data streams with	27 errors in is	2443
number of data streams with	28 errors in is	1934
number of data streams with	29 errors in is	1509
number of data streams with	30 errors in is	1042
number of data streams with	31 errors in is	679

DAMYLES job terminated at 1-NOV-1991 15:05:05.26

Table 7.19 Ferreira Code with Bitflip

References

- 7.1) N.Wiener** "Extrapolation, Interpolation and Smoothing of Stationary Time Series" Wiley New York
- 7.2) C.F.N.Cowan and P.M.Grant** "Adaptive filters, Chapter 2" Prentice Hall 1975
- 7.3) B.F. Ryan B.L.Joiner T.A. Ryan** "Minitab : a Users Handbook" Duxbury Press Boston 1975
- 7.4) E.Katz and T.Campbell** "Effect of Bitshift Distribution on Error Rate in Magnetic Recording" IEEE Transactions on Magnetics Vol MAG-13 No.3 May 1979 pp 1050-1053
- 7.5) A.Ryley and M.K.Loze** "Bitshift Errors due to Wideband Noise in Digital Magnetic Recording" Eight International Conference on Video Audio and Data Recording Birmingham England 24-26 April 1990 pp 15-19
- 7.6) B.K.Middleton, J.J. Miles and R.H.Noyau** "Magnetic Properties of Materials, Transition Shapes and Digital Recording Properties" IEEE Trans on Magnetics Vol MAG-26 No.5 Sept 1990 pp 2128-2130
- 7.3) National Algorithms Group** "Fortran Mini Manual Mark 10: Programme, G05DDF Normal Distribution" 1983
- 7.8) L. Gemon** "Statistique Appliques au Codage Correcteur d'Erreur pour les Enregistrements Magnetiques Numeriques" Research Report Polytechnic of Wales 1991
- 7.9) H.C. Ferreira and S.Lin** "Of Integer Compositions and Error Correcting (d,k) Block Codes"Submitted to IEEE Transactions on Information Theory

Chapter 8

Conclusions and further work

This thesis presents the theoretical results for the effect of Modulation and Error Correcting Codes in combating Non-Clockloss Errors. These results are supported by computer simulation. However as has been stated in the introduction only non-operational values were used for the probability of error. This was due to the restrictions on the amount of computing time allowable for each programme. However these results confirm theoretical predictions and suggest trends at operational levels of error rate.

Noise from a bandwidth limited channel such as those which occur in digital magnetic recording is correlated pulse-to-pulse, although this has little effect upon the error statistics when compared to the uncorrelated case. It has been shown that ISI is independent from pulse to pulse in magnetic recording. These results confirm the widespread assumption that both Noise and ISI and hence all non clockloss errors are independent pulse to pulse.

The channel models considered show that the Asymmetric Channel (ASC) dominates. An appropriate strategy for pulse error reconstruction was developed for Manchester Modulation code.

Modulation coding strategies considered both simple, direct and block codes, as well as those with fixed lengths and a variable number of pulses and those with variable lengths and a fixed number of pulses. The phenomenon of bit-slip was found to be more prevalent in fixed pulse codes than in fixed rate codes.

In particular the Ferrieria codes, were shown to be vulnerable to bit-slip. A new coding scheme for fixed pulse runlength based combined error correcting and modulation codes with more stringent runlength constraints was developed to give an improved performance against bit-slip errors.

Error control coding was investigated: both linear codes (B.C.H.), in which the codevector consists of a series of individual bits, and non-linear codes (Reed Solomon), where the codevectors are divided into a series of distinct subwords, the performance of these codes were measured against noise and ISI. The results from which form the main part of Chapter 7.

Throughout the work a symmetrical replay pulse was assumed which is equivalent to assuming that there is no perpendicular recording component. It was shown that this assumption resulted in an underestimation of the level of ISI induced bit-shift. This was because the symmetrical model resulted in cancellation of ISI components which would not occur in more accurate pulse shape models.

The results of chapter 6 and 7 suggest areas of interest for further work: in particular there are three areas which may be developed:

1) The Power Spectral Density of Modulation codes.

Signal-to-noise considerations dictate that the power density spectrum of the modulated codevector matches the frequency response of the recording channel.

Currently there exist models for the Autocorrelation function and hence the power spectral density of direct modulation and shorter block codes. However preliminary work suggests that it is possible to develop the methods to find the Power Spectral Density of longer block codes such as EFM and 8/10 by appropriate Matrix Manipulation.

2) Combined Error Correction and Modulation codes.

These have been designed to combine the runlength constraints of Modulation codes with the Hamming distance properties of the Error Control codes. Their purpose is to develop a reliable method of storing data at high densities. As with conventional Error Correcting Codes, combined codes should be matched to the error channel. Hence for combined codes for digital recording the ASC should be fully exploited.

To assess combined codes the data rates of perfect Error Control codes together with the limits on the Information Rates of fixed pulse Modulation codes will be required. Further, it will be necessary to assess the Power Spectral Densities of any new codes. Finally the resistance of these codes to error propagation by either dropouts jitter or bit slip should be investigated.

3) Bitflip Analysis

Bitflip is a major source of error in digital data recording. Its effect is similar to that of a major dropout as it corrupts every piece of data to the end of the block under examination. We may conjecture that the problem of bitflip is very sensitive to the choice of channel code used. This is because bitflip is directly related to the ability of the Modulation code to maintain good clocking or synchronisation which is generated from the Modulation code. Work is necessary to develop codes with specific bitflip resistant properties.