University of Plymouth

PEARL **https://pearl.plymouth.ac.uk**

04 University of Plymouth Research Theses 01 Research Theses Main Collection

2017

Reducing the Complexity of Equalisation and Decoding of Shingled Writing

Abdulrazaq, Muhammad Bashir

http://hdl.handle.net/10026.1/9332

http://dx.doi.org/10.24382/1166
University of Plymouth

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.

Reducing the Complexity of Equalisation and Decoding of Shingled Writing

PhD Thesis Report.

Plymouth University, UK

By:
Muhammad Bashir Abdulrazaq

1st Supervisor:
Dr.Mohammed Ahmed Zaki

2nd Supervisor:
Dr. Paul Davey

May 14, 2017

Reducing the Complexity of Equalisation and Decoding of Shingled Writing

Muhammad Bashir ABDULRAZAQ

May 14, 2017

Copyright ©2016

This copy of thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without author's prior consent.

Reducing the Complexity of Equalisation and Decoding of Shingled Writing

A thesis submitted to Plymouth University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

by
Muhammad Bashir Abdulrazaq
May 14, 2017

School of Electronics, Mathematics, and Computing
Faculty of Science and Technology
Plymouth University, UK

Abstract

Reducing the Complexity of Equalisation and Decoding of Shingled Writing
by
Muhammad Bashir Abdulrazaq

Shingled Magnetic Recording (SMR) technology is important in the immediate need for expansion of magnetic hard disk beyond the limit of current disk technology. SMR provides a solution with the least change from current technology among contending technologies. Robust easy to implement Digital Signal Processing (DSP) techniques are needed to achieve the potentials of SMR.

Current DSP techniques proposed border on the usage of Two Dimensional Magnetic Recording (TDMR) techniques in equalisation and detection, coupled with iterative error correction codes such as Low Density Parity Check (LDPC). Currently, Maximum Likelihood (ML) algorithms are normally used in TDMR detection. The shortcomings of the ML detections used is the exponential complexities with respect to the number of bits. Because of that, reducing the complexity of the processes in SMR Media is very important in order to actualise the deployment of this technology to personal computers in the near future.

This research investigated means of reducing the complexities of equalisation and detection techniques. Linear equalisers were found to be adequate for low density situations. Combining ML detector across-track with linear equaliser along-track was found to provide low complexity, better performing alternative as compared to use of linear equaliser across track with ML along track. This is achieved if density is relaxed along track and compressed more across track. A gain of up to 10dB was achieved. In a situation with high density in both dimensions, full two dimensional (2D) detectors provide better performance. Low complexity full 2D detector was formed by serially concatenating two ML detectors, one for each direction, instead of single 2D ML detector used in other literature. This reduces complexity with respect to side interference from exponential to linear. The use of a single bit parity as run length limited code at the same time error correction code is also presented with a small gain of about 1dB at BER of 10^{-5} recorded for the situation of high density.

Dedication

This work is dedicated to Almighty Allah for the life, health, supporting family and supervisors given to me.

Acknowledgement

I wish to express my unquantifiable gratitude to Almighty Allah for the provision of life, health and financial, academic and moral supporters throughout this research and my whole life.

I also give my immense thanks to Etisalat Nigeria for the funding of my PhD programme. Also many thanks to the Etisalat staff, Ahmadu Bello University staff such as Prof. M. B. Muazu, Plymouth University staff, and others who contributed, in one way or the other, to making sure this programme went on smoothly.

My thanks go to my supervisors, Dr Mohammed Zaki Ahmed, Dr Paul Davey and Dr Marcel Ambroze for the support, advice, and guidance through this research programme.

My thanks also go to my colleagues during the research. Your company and advice are of great help to the continuity of my programme.

Most importantly I will like to give my thanks to my wife, Hajara Lawal, kids, Umair, Habiba and A'isha. You are a great motivation for me and source of comfort in times of difficulties.

Lastly, I wish to say thank you to my parent, Khadijat Abdulrazaq, and Abdulrazaq Yahaya for everything that I am. My family and friends back at home who support me with their prayers and advice. I wish you all fruitful life and good health.

Declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award without prior agreement of the Graduate Committee.

This study was financed by Emerging Markets Telecommunication Services Limited (Etisalat) Nigeria Ltd.

A programme of advanced study was undertaken, which included the extensive reading of literature relevant to the research project, development of software based simulations, development of hardware implementation designs and attendance of international conferences on magnetic recording and digital signal processing area.

The author presented his work at 2015 International Conference on Magnetism (ICM 2015)at Barcelona Spain, and 2015 Telecommunication Forum Conference (TELFOR 2015). The author also submitted papers for publication in, TELFOR Journal.

In addition, the author co-authored a paper presented at 2016 Asia-Pacific Magnetic Recording Conference (APMRC 2016).

Word count of the main body of thesis: 32,416

Signed...
. Muhammad Bashir Abdulrazaq

Date..

Contents

List of Figures

List of Tables

Abbreviations

1D	One Dimensional
2D	Two Dimensional
ACSU	Add, Compare, Select Unit
APP	A-Posteriori Probability
AWGN	Additive White Gaussian Noise
BCJR	Bahl, Cocke, Jelinek and Raviv
BER	Bit Error Rate
BM	Branch Metric
BMU	Branch Metric Unit
BPMR	Bit Patterned Magnetic Recording
DM	Distance Metric
DRP	Dithered Relative Prime
DSP	Digital Signal Processing
EAMR	Energy Assisted Magnetic Recording
FEC	Forward Error Correction
FER	Frame Error Rate
FPGA	Field Programmable Gate Array
GCR	Group Coded Recording
GFP	Grain Flipping Probability
GPR	Generalised Partial Response
HDD	Hard Disk Drive
ISI	Inter-Symbol Interference
ITI	Inter-Track Interference
JMB	Joint-Multi Bit
LDPC	Low Density Parity Code
LMR	Longitudinal Magnetic Recording
LUT	Look-Up Table
MAP	Maximum A-posteriori Probability
MIMO	Multiple Input, Multiple Output
ML	Maximum Likelihood
MLSD	Maximum Likelihood Sequence Detector
MMSE	Minimum Mean Square Error
MR	Magneto Resistive
MSE	Mean Square Error
MWC	Multiply With Carry
NRZ	Non-Return to Zero
NRZI	Non-Return to Zero Inverted
PMR	Perpendicular Magnetic recording
PR	Partial Response
PRML	Partial Response Maximum Likelihood
RLL	Run Length Limited
RSC	Reed-Solomon Code

SM	State Metric
SMR	Shingled Magnetic Recording
SNR	Signal to Noise Ratio
SOP	Sum Of Products
SOVA	Soft Output Viterbi Algorithm
TBL	Trace-Back Length
TBU	Trace-Back Unit
TDMR	Two-Dimensional Magnetic Recording
TMR	Transverse Magnetic Recording
VA	Viterbi Algorithm
VHDL	VHSIC Hardware Description Language
WGN	White Gaussian Noise
ZF	Zero Forcing

Chapter 1

MAGNETIC RECORDING

1.1 Introduction

Being able to access information at a different location from the location where the data was created or at a different time from the time at which the information was created is arguably the most important feature of modern digital technological era.

Whether the digital information is to be accessed at a later time or different location, there is normally a form of storage, which can be temporary, until the data are processed, or permanent storage, to be accessed at any later time.

Over time, different medium technologies for storage of information were used. They evolved into supporting data storage in various forms, and are able to handle various capacities of information. Magnetic storage has been one of the oldest and still one of the most important storage technologies. Hard Disk Drive (HDD), especially, plays an important role in making the access to information realisable, affordable and portable.

Magnetic HDD is now the most common recording medium used in millions of personal computers found in our offices and homes as well as industrial application. They form the backbone of all communications on the internet and intranets.

IBM first recognises HDD, in the early 1950s, as a suitable memory device that can offer means of random access to data that is stored on it. IBM released its first commercial HDD, "IBM model 350", in 1956. This HDD, which came with "IBM 305 RAMAC system", had a capacity of 3.75 MB [1]. For a period of about 50 years since then, the capacity of magnetic HDD storage has been exponentially increasing amounting to an increase of about six figures within 50 years. But due to a limit imposed by the superparamagnetic limit of the Perpendicular Magnetic Recording medium, the increase is stalled. Due to this problem, new ideas of how to keep the capacity expanding must be thought of. The limit of the current technology was evaluated to be about $1Tb/in^2$ [2].

1.2 Magnetic Medium

From inception, there is a constant drive by manufacturers to increase the capacity of storage media, to meet the expectations of users and the increasing demand for storage space in portable devices.

The most popular format of data stored on a magnetic medium is a two-level bit. That is either a 0 or a 1. This is due to increase in susceptibility of the data to noise and distortions if the levels go beyond two stages. The distortions go beyond the gain in capacity or an exponential increase in complexity results from use of more levels of magnetisation [3]. The writing is done by changing the direction of magnetisation of a portion of the medium. Two directions of magnetic field are therefore required to represent the digital information stored in the form of 0s and 1s.

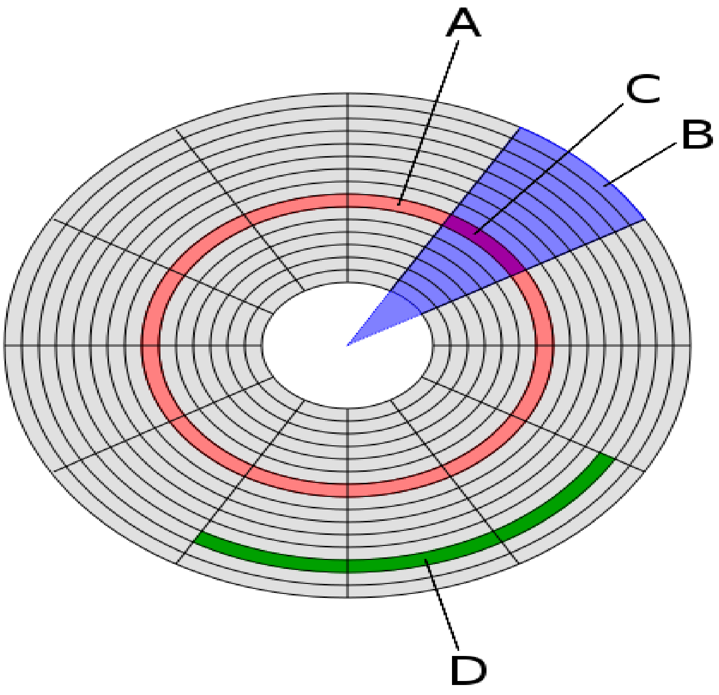


Figure 1.1: Disk structure: A=Track, B=Geometrical sector, C=Track sector, D=Data Sector/Cluster. [4]

The bits are spatially written as a sequence, one after the other, in the direction of motion of the write-head. One such continuous sequence is called a track. Tracks of given length(number of bits) are grouped side by side to form a data sector. Data sector can be part of or comprising many physical sector(s) of the disk. In older HDD, tracks are separated by a space called guard band to avoid over-writing data from neighbouring track during write process or reading part neighbouring track during read process (interference). Guard bands are also placed between data sectors(blocks of data) to separate one sector from another. Figure 1.1 shows a simplified structure of a hard disk[4]. More complex arrangement are usually implemented in modern hard disk to utilise the larger size of the outer physical sectors.

Depending on the relative direction of the field to the direction of motion of the write head or to the plane of the HDD plate, three modes of storing data can be identified. These are, Longitudinal, Transverse and Perpendicular Magnetic recording [5].

In Longitudinal Magnetic Recording (LMR), data are stored such that the direction of the magnetic field is parallel to the plane of the medium and to the direction of motion of the write head, with respect to the medium. The direction of magnetic field here refers to, the direction faced by the north and south pole of the field. This method of data storage has been popular and is the dominant method of data storage in HDD up to 2005 [6]. An illustration of the storage mode is shown in Figure 1.2.

Transverse Magnetic Recording(TMR) is a storage mode in which, the direction of the magnetic field is parallel to the plane of HDD plate but perpendicular to the direction of motion of the write head with respect to the plate. An illustration of the writing scheme is shown in Figure 1.3. This method of writing data was not popular, as it presented no significant advantage over LMR [5].

But the drive to get more reliable storage systems with higher capacity continued and led to the re-introduction of Perpendicular Magnetic Recording (PMR). In this method, the direction of the field is perpendicular to the plane of the HDD plate. This makes it possible to make the writing area smaller, the write head smaller, and have the bits moved closer to its neighbour without having demagnetisation field affecting the neighbouring bits very much. Because PMR produces stronger effective field in the direction perpendicular to the medium, materials with higher coercivity are used with soft magnetic under-layer (SUL). This gives it a better thermal stability. This is now the most common technique used in HDD writing. This technology, which was laying around for many years, was re-introduced by Toshiba in 2005, and since then it drove LMR out of the market [6]. Figure 1.4 shows an illustration of PMR recording method.

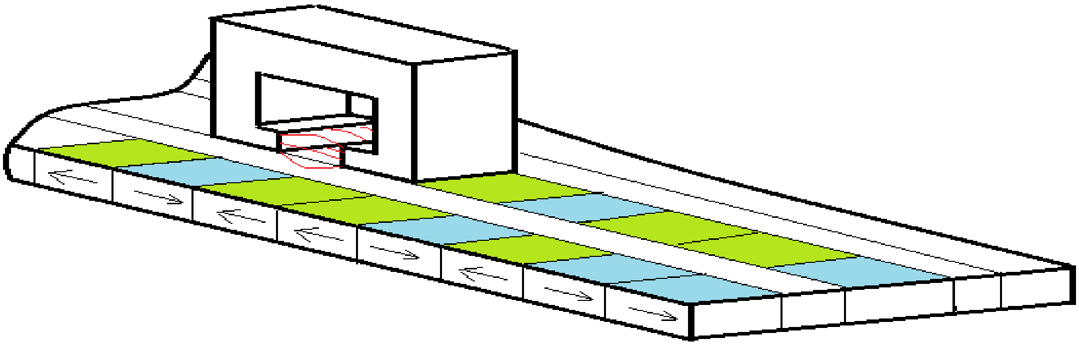


Figure 1.2: Longitudinal Magnetic Recording

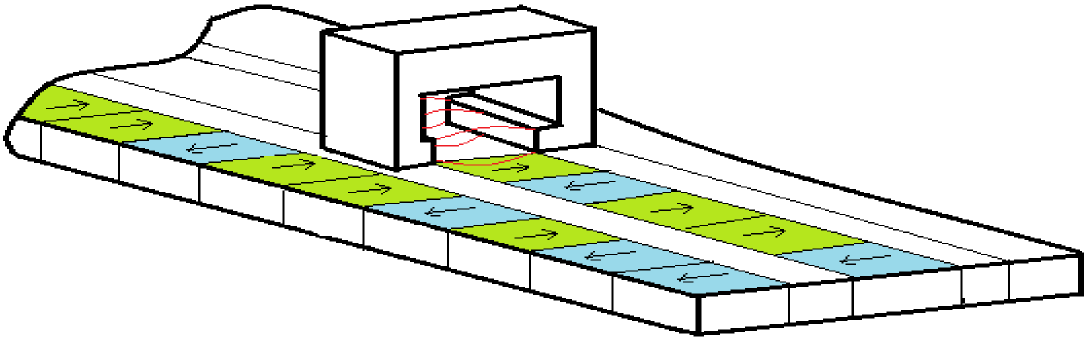


Figure 1.3: Transverse Magnetic Recording

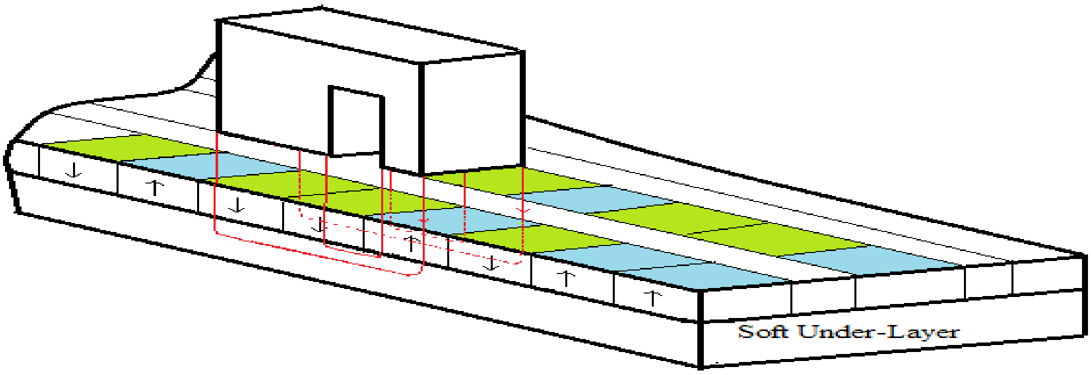


Figure 1.4: Perpendicular Magnetic Recording

1.3 Reading Magnetic Disk

An efficient read head is important to the success of a magnetic recording scheme. There are basically two types of read heads used in magnetic medium. The inductive read head, which is called “inductive head” for short, and Magneto Resistive (MR) read head.

Inductive head is effectively an inductor. The same inductive circuit is usually used to act as the write head and read head alternately. When the inductor is powered by current (representing data), it is used as a write head, whereas it reads the magnetisation of the disk as it flies across the surface of the HDD. It picks the magnetisation through the induced EMF described by Faraday’s law of induction [7].

Inductive head responds to changes in a magnetic field (magnetic transition). It responds to the horizontal component of the magnetic field as it picks the transitions of the field from one direction to another. This implies that it differentiates the flux on the medium. The output voltage is, therefore, proportional to the change in magnetisation of the area over which the head is flown [8].

On the other hand, MR heads consist of a separate inductive write head and a separate MR read head as shown in figure 1.5. MR substances are materials that have a resistivity that depends on magnetic field around the substance. MR elements respond to the perpendicular component of a magnetic field and changes resistance when the field changes. Therefore, when used in LMR, MR head reads changes in the field direction or strength not the actual value of the field. This is because the field on longitudinal magnetic medium is only perpendicular at the points of transition.

The read signal by MR read head is proportional to the perpendic-

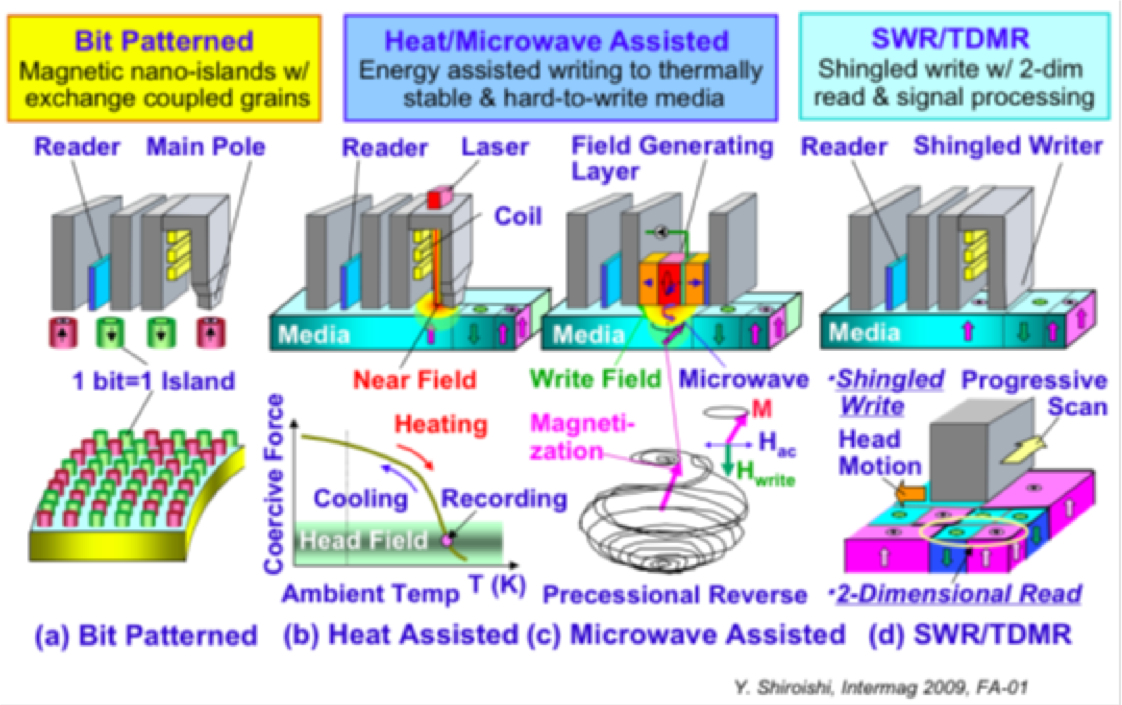


Figure 1.5: Primary magnetic recording technology alternatives, from the work of Shiroishi et. al. presented at the International Magnetics Conference, May 2009.

ular component at the transition regions in LMR medium. The signal is effectively similar to the signal read by an inductive head for isolated response [8]. But when used with PMR medium, the head reads the actual direction of the magnetisation. This is because the field is perpendicular to the plane of the disk. Therefore, the value read will be the actual data stored.

MR heads are significantly smaller than inductive heads. This makes them more popular than inductive heads. In a normal setup, the MR read element is placed in between shields alongside the inductive write head, to protect it from the effect of magnetic field from the write inductor as can be seen in figure 1.5.

1.4 Data Format

Depending on what is read by the read-head, and requirements of synchronisation and clocking, the data to be written on the medium is given a certain format to make sure it is suitably read when reading is taking place.

The two formats of the written data used on digital magnetic medium can be categorised as Non-Return to Zero (NRZ) or Non-Return to Zero Inverted (NRZI).

The direction of the field in NRZ format represents the data whether 0 or 1. That is to say, one direction of magnetisation represents 0 while the other represents 1. This is straight forward but only convenient where the read-head can read the absolute value of the magnetic field on the medium. This is true if MR read head is used on a PMR medium. But when an inductive head (which reads transitions) is used, NRZ can lead to inconveniences in detection or even fatal convoluted errors. Therefore, NRZI is used in such cases.

NRZI is a format in which a change in the direction (rather than the actual direction) of a magnetic field is used to represent data. A transition indicates the presence of a 1, while lack of a transition is used to represent a 0. This means even if there is an error in the detection of a bit, it will not affect the read data ahead [9]. The task of converting the data to the NRZI is termed pre-coding [10]. Figure 1.6 show an example of NRZ and NRZI for set of bits 10011101.

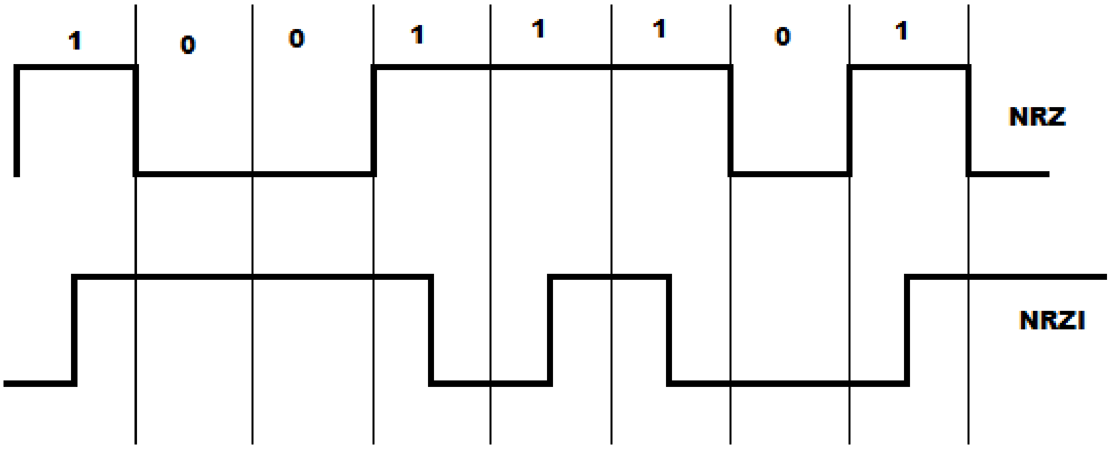


Figure 1.6: In Non-Return to Zero(NRZ) amplitude represents data. In Non-Return to Zero Inverted (NRZI) change in amplitude represents data

1.5 Media Trilemma

Increasing data areal density (number of bits per unit area) of magnetic HDD by reducing the size of bit along or across track has its limitations. Mechanically the size of the write head is usually the limiting factor as MR read heads are usually smaller than the write heads.

But the physics of the magnetic recording material also presents some challenges. If the write head is made smaller, then the magnetic field strength of the head will have to be reduced accordingly to limit its reach to a smaller area. This makes it more difficult for the magnetic field to flip the magnetisation of a region. This problem is coined the **write-ability problem**. But the write-ability problem can be resolved by choosing a HDD material with low coercivity (to be easily writeable). This, however, makes the magnetisation easily destroyed or reversed by a small random thermal fluctuation. This problem is called the **thermal stability problem**. In order to avoid the two problems above, a HDD medium with high coercivity has to be used. Also, a way of making the field strength of the write-head very strong has to be found. But strong writing field means the field will have effect to a large distance from the bit position. This creates larger interference between

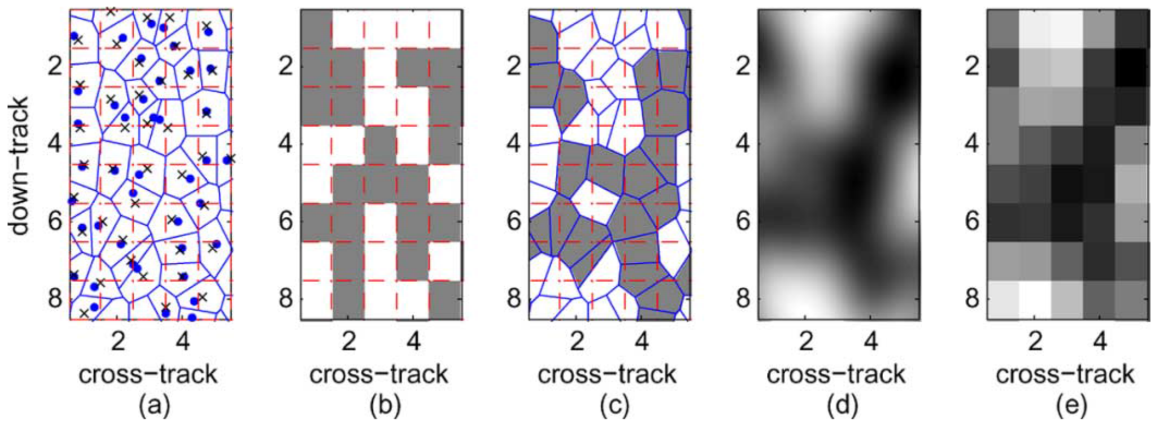


Figure 1.7: Illustrative example of TDMR writing and reading. (a)Voronoi cells. (b)input data bit array. (c)recorded Voronoi cells. (d)multiple scans concatenated in parallel. (e)and equivalent discrete channel output sampled at the centre of the bit cell.[44]

neighbouring bits thereby reducing Signal to Noise Ratio (SNR). This problem is called the **SNR problem**. These three problems, coined the Media Trilemma by [11], impose a super-paramagnetic limit on the maximum achievable areal density of current magnetic HDD technology.

At the base of all the limits, to maximum areal density, is the grain size of the magnetic material. The grains are the small irregular sized areas that form a magnetic surface. A grain is the smallest portion of the magnetic plate that can be magnetised with a unique magnetic field. This means a bit of information can only be stored on an area not less than one grain particle. Refinement of the materials leads to magnetic material of smaller grain size. The average size is about 10 nm in size. Therefore, the highest areal density of HDD that can be achieved now, putting the Media Trilemma and grain size into consideration is now estimated to be 1 Tb/in^2 [2]. Figure 1.7 shows a depiction of Random magnetic grains and how writing and reading to it affects the capacity of the medium.

1.6 New HDD technologies

In order to go beyond the superparamagnetic limit of HDD and continue the capacity expansion of the medium, either a more refined technology has to be presented and used or a new approach to the way current technology is used has to be created.

Among the popular new ideas and approaches suggested for the continued expansion of magnetic HDD capacity are Bit Patterned Magnetic Recording (BPMR), Energy Assisted Magnetic Recording (EAMR) and Shingled Magnetic Recording (SMR) [12]. The three technologies mentioned here follow different approaches to tackling the problem of going beyond the superparamagnetic limit. Depictions of these three technologies are shown in figure 1.5.

1.6.1 Bit Patterned Magnetic Recording

BPMR tries to address the SNR problem of the media trilemma by making sure the bit positions are as isolated as possible from each other. In BPMR it is suggested that the magnetic disks should have specially designed protrusions (islands) on their surfaces, to carry the magnetisation of each bit. The extra surface at the vertical face of the island and the non-magnetic material separating the islands increases the separation of the bits' magnetisation, thereby reducing interference from neighbouring bits[13]. This improves the SNR of the medium and therefore gives chance for higher density of data.

The drawback of this approach is the fact that when reading from or writing to the disk, the head must precisely locate the position of the island in order to effectively read or write. This fact makes it very susceptible to deviations of the head due to movements of the device, or deviation of the location of the islands during fabrication of the disk.

Variation in island size, location and shape can introduce noises into the read information [14]. Another drawback is that the technology is a drastic change from the current technology, which means new computing devices that support the technology and new manufacturing process must be established.

1.6.2 Energy/Heat Assisted Magnetic Recording

In order to address the problem of writeability in the media trilemma, it was also suggested that heat energy sources such as microwave or laser be used to soften the magnetic medium when writing the data. Heat softens magnetic materials and makes it easier to change the magnetisation of the material or a portion of it temporarily [15], [16]. This means the coercivity of the medium doesn't have to be reduced. This is because the heat will make it soft enough to be written onto, by a relatively weaker field. The thermal stability of the medium will therefore still be intact when the medium cools off after the writing is finished.

But this technique has its own drawbacks. The drawback is that of the requirement of extra energy for writing data. This energy, even though it is not much when compared to the energy used by the disk, is not suitable for mobile devices that run on batteries and large scale storage systems whose energy consumption may be significantly increased. There is the need for new type of lubricant capable of withstanding high temperature and new system design [17].

1.6.3 Shingled Magnetic Recording

SMR tries to address the writeability and stability problem without any drastic change in the current magnetic media technology. This makes it popular and desirable by manufacturers. Its approach is to circumvent the limitation of track width being limited by write-head width by overlapping data tracks when writing. The read-head which can be significantly smaller than the write-head can be made of suitable size for reading the data.

The drawback of SMR is the fact that the overlapped tracks are inherently connected. This means a small part or portion of the data can not be modified or updated without affecting all the neighbouring tracks connected to it. A whole sector of data must be updated if an update in place operation is to be carried out. Interference from neighbouring tracks also becomes an important issue which is also an issue in the other techniques mentioned above [12].

SMR represents a smoother transition from current technology. It is achievable with current technology if a robust Digital Signal Processing (DSP) technique is applied. Therefore, despite the challenges of SMR, it becomes more popular as a technology that will, in the immediate future, keep the capacity expansion of magnetic media on track.

SMR disks have already started coming into the market with, Seagate first announcing a 25% increase in its disk capacity by using SMR technology. This was released in 2014. This made it possible for their HDD to have 1.25TB per platter which replaces their 1TB disks [18]. This makes its 4 platter HDD to now have 5TB capacity. In September 2014, HGST also announced an SMR disk of capacity 10TB which is filled with helium to reduce internal turbulence and resistances. The disk is mainly meant for archiving purposes [19] [20].

1.7 Effective Capacity

The raw capacity of a magnetic HDD is affected by the grain size and bit size as explained. But the effective capacity is usually different.

This is because there are losses or gain in capacity due to channel coding, which may be inserted for timing synchronisation purposes, reduction of interference, or other purposes. Error correction coding and other control information also reduce the raw capacity of the disk. Some of the codes involved are explained below.

1.7.1 Run Length Limited code

Reading magnetic disks needs clocking system, to make sure the data read is almost perfectly synchronised to the written data. If there is a long run of a single bit, the system may deviate from the write timing. That is why it is important to have frequent transitions from one bit to another to make sure the clocking is regularly updated. Popular coding or clocking technique, used in magnetic media, is the Run Length Limited (RLL) code. RLL is also used to separate transitions from one magnetisation to another so that more data can be compressed on the disk [21].

RLL codes define a minimum number of consecutive zeros (for separation of transitions in systems in which the read head only reads transition from one bit to another different bit) and a maximum number of consecutive zeros (for timing synchronisation) that can occur in a particular length of bits. They are defined as RLL(d,k,m,n,r) or RLL(d,k) for short, where:
d = Minimum number of consecutive zeroes allowed (including clock).
k = Maximum number of consecutive zeroes allowed (including clock).
m = Minimum number of data bits to be encoded.
n = Number of code bits (including clock) for each of the m data bits.
r = Number of different word lengths in a variable length code [22].

A popular RLL code used in magnetic storage media is RLL(1,7). It

Table 1.1: RLL(1,7) mapping table[22]

Data nibble	RLL (1,7) code	Data nibble	RLL (1,7) code
00	101	00 00	101 000
01	100	00 01	100 000
10	001	10 00	001 000
11	010	10 01	010 000

is a rate 2/3 code. This implies it takes 2 data bits and convert them to 3 data bits. Therefore, it can be written in full as RLL(1,7,2,3,1). Table 1.1 shows how to map bits to an RLL code.

In order to encode a pair of bits, the pair ahead of the pair to be encoded is checked whether the four bits can be found in the third column (first column on the right-hand side). If found, the RLL code on the right-hand side is used for the four bits. If not found, the RLL code in the left hand is used for the first two bits.

This reduces the disk capacity by a factor of two third due to coding but can allow closer compression of data to a density ratio of 1.33. This means, RLL(1,7) can lead to increased data density of recording media. The increase being by a factor of 1.33.

Other coding techniques exist suitable for different conditions such as RLL(0,2) also known as Group Coded Recording (GCR).

1.7.2 Forward Error Correction

Despite all precautions taken to make sure the correct data is read from a storage medium, errors occur. They sometimes occur due to electronic noise and sometimes due to interferences. Therefore, the data are usually coded such that if an error occurs, it can be identified and possibly be corrected. This kind of system is called Forward Error Correction (FEC).

In older magnetic storage media, Reed-Solomon Codes (RSC) are used to encode the data as a FEC techniques [23]. RSC groups the bits into blocks, and encode groups of such blocks using algebraic equations. This makes it possible that, at the receiving end, the equations will not be satisfied if there is an error in one of the blocks. If an error is identified, an algorithm is used to identify the closest code to the received code that will satisfy the coding equation.

Nowadays Low Density Parity Check (LDPC) gains more popularity as an error correction code for magnetic HDD. In this scheme, many interconnected simple and short parity equations are formed in a long stream of data. They are later decoded iteratively until a solution is found, or a certain number of iterations is reached without finding a solution. LDPC is simple to decode and can be parallelised easily for fast processing. But it has a drawback of complex encoding for normally good performing codes [24].

Generally, FEC reduces the raw capacity of storage medium because extra bits of information are needed to form the code. But it makes the system more resilient to noise such that, the data can be compressed even more. The product of the gain and loss in capacity gives the average gain of using a FEC. A good FEC should have an over unity gain in capacity.

1.8 Aims

The current technological status of SMR/TDMR as released by Seagate was able to achieve only 25% increase as against around a 1000% (10 times) that could be achieved if effective, easy to implement digital signal processing techniques are available [44]. The major difficulty is the very high complexity of the 2D detectors that can possibly provide the required performance at very high density. As the interferences get higher at high density, more symbols need to be collected to produce good estimation of saved data. The complexities of the 2D detectors normally used get exponentially higher as the number of symbols increase. Therefore, finding an efficient low complexity channel coding, FEC, equalisation, detection and decoding techniques plays a very vital role in maximising the storage capacity of the magnetic HDD. This is why finding a computationally simple algorithm with reasonable hardware and energy requirement is very important in magnetic storage and SMR in particular.

Due to the importance of DSP in achieving the projected capacity in SMR as explained above this research is directed towards finding a computationally less complex equalisation, detection and decoding algorithms for retrieving data from a Shingled Magnetic Recording medium. It is also desired to make the algorithm less hardware intensive. That is to make sure it can be implemented on hardware with a reasonable amount of hardware requirement.

1.9 Objectives

The following objectives are outlined to facilitate the achievement of the stated aims of the research:

1. A shingle based Perpendicular Magnetic Recording channel is to be modelled.
2. Linear equalisers are to be made, in order to filter the data from the channel to a required, simpler target response.
3. Implementation of linear equaliser that will cancel one of ITI or ISI to reduce the detection problem into one-dimensional problem will be carried out.

4. One-Dimensional Partial Response Maximum Likelihood (ML) detector based on Viterbi and BCJR algorithm is to be implemented.

5. Optimised equaliser, which minimises noise amplification, with minimum length possible is to be searched.

6. Best target response, algorithms and density in terms of bit error rate (BER) and frame error rate (FER) are going to be investigated.

7. Hardware implementation using Field Programmable Gate Array (FPGA) platform will be conducted.

8. Full two-dimensional (2D) ML detector to be implemented.

9. Analysis of various 2D targets for different data density along the tracks and across the tracks is to be made.

10. Attempt at simplification and reduction of complexity of the full 2D algorithm be made.

11. Analysis of comparative performance gain of full 2D detector as compared to the combination of 1D detector and a linear equaliser will be done.

12. Incorporation of Forward Error Correction into the detection algorithm to be done.

13. To find algorithms, target and raw data density that will give the best performance among the investigated algorithms.

14. To determine the complexity reduction gain, to get adequate performance or performance improvement, for the best performing algorithm.

15. And to determine the hardware requirement of the best performing algorithm.

1.10 Methodology

The following steps are taken in order to achieve the aims and objectives of the research.

- A software model of a Shingle based Perpendicular Magnetic Recording Channel is formed, which put into consideration the noise that affects the data, due to position jitter, electronic white noises, inter-symbol interference and inter-track interference (2D noise model). C++, MATLAB and VHDL were used at various places for the modelling.

- A linear Zero Forcing (ZF) equalisers and Minimum Mean Square Error (MMSE) equalisers were made, to validate the channel model and have their performances as a reference for comparing the other equalisers to be formed.

- A code is developed, which uses channel response and target response, to determine a linear equaliser that will shape the signal to any desired target response in one direction. A ZF equaliser is used in the other direction to remove the other interference.

- A Viterbi based detector was developed to carry out ML detection in the other direction as a means of Partial Response ML (PRML) detection.

- ZF was used across track, while PRML was applied along track. Also, ZF was used along track while PRML was applied across track. Performances were analysed and compared.

- Performance check for the targets, equaliser lengths, detector trace-back depth, medium data density along and across track, percentage jitter/white noise were carried out to determine optimal parameters.

- VHDL equivalents of the two versions of ZF combined with PRML (1D detection) were implemented on an Altera Cyclone V SockIt design kit for further analysis and hardware requirement.
- Full 2D joint track detector was implemented by reducing the problem into two serial detectors along and across track.
- Various targets and densities were investigated to determine best performing parameters.
- Similar paths are identified and eliminated to reduce computational and hardware complexity.
- Further analysis to determine gain over combined ZF equaliser and PRML was carried out.
- Two (even inner and odd outer) serial single parity bits of block length 4 were used as an error correction code. The parity codes were separated by a block interleaver in one instance and then a Dithered Relative Prime (DRP) interleaver in another instance.
- Full analysis of the 2D detector for optimal performance was carried out.

1.11 Contribution to Knowledge

The following are some contributions of this research to knowledge:

- In the conduct of the research, it was found that for an SMR with Inter-Track Interference (ITI) and Inter-Symbol Interference (ISI), where a linear equaliser is to be used to cancel one of the interferences and PRML used to cancel the other, more capacity can be achieved by using linear equaliser along the tracks to cancel ISI while PRML is used across the tracks to detect the data with ITI. A drawback is, the use of linear equaliser along track and PRML across track makes the HDD have a slower speed.

- In the case of a full joint track 2D detector, the complexity of detection can be reduced by splitting the detector into along and across track detectors. The data of each track is reduced to a smaller number of tracks to be jointly detected. 2D ML detector is used to determine the likelihood of symbols representing the central bit with ITI. Then another 1D ML detector uses that information across track to finally detect the data.
- In full 2D joint track detection, better performance, and consequently higher capacity is achieved, when 2D ML detection is started across the track then the final detection done along track. It performs better than when 2D ML detection is started along tracks then final detection done across the tracks. This is especially evident at high densities and ITI.
- A scheme for combining detection and decoding of data is presented with the aim of saving energy and computations and improvement of performance in an SMR medium.
- The use of single odd parity bit to serve as RLL(0,6) code at the same time used in error correction is demonstrated in this research. This saves computation and delay in the processing of information.
- Serially cascaded parity bits, as a FEC, is demonstrated which harnesses the simplicity of decoding single parity bit to achieve an appreciable performance gain in SMR.

1.12 Structure of Thesis

This Thesis reports the conduct and results of the research titled "Reducing the Complexity of Equalisation and Decoding of Shingled Writing" and it is structured in the following way.

Chapter 1 gives a general introduction to magnetic storage medium, problems hindering capacity increase and technologies to help overcome the problems. The aims and objectives of the research are also itemised with the methodology intended to achieve the aim.

Chapter 2 reviews literature based on modelling SMR and magnetic medium in general. It also discusses literature based on equalisations and detections of information on a magnetic medium.

Chapter 3 presents the code design and implementation of the channel model, equalisers and detectors that were designed and used in the research. It involves the design of Perpendicular Magnetic Recording (PMR) channel medium using jitter noise model, equaliser designs and Maximum Likelihood (ML) detection using Viterbi Algorithm and BCJR.

Chapter 4 presents the performance of Zero Forcing, MMSE equaliser, Across-track PRML, and Along-track PRML. They are based on a target length of 3 and ITI of 3 tracks.

Chapter 5 presents the performance of full 2D detectors based on VA and BCJR for different target lengths. Effect of adding two concatenated single parity bits are also investigated. The complexity of the various equalisation and detection/decoding techniques are analysed and the relative advantages and disadvantages of the various equalisers and detectors are also discussed in this chapter.

Chapter 6 was used to presents conclusions derived from the research, contributions made and suggestions for further research.

Chapter 2

SHINGLED MAGNETIC RECORDING

In this chapter, reviews of literature based on modelling SMR and magnetic medium, in general, is presented. The chapter also discusses literature based on equalisations and detections of information on a magnetic medium.

2.1 Introduction

Magnetic recording medium used in HDD is used to store data in small areas, magnetised to be either positively or negatively magnetised. That is, either north or south pole facing a given direction to represent either 0 or 1.

The magnetised areas are desired to be of regular predictable shapes. When the storage has a low areal density, it can be approximately assumed to be of the desired regular shape. But at very high areal density, the actual shapes of the grains are randomly irregular and are very significant to the quality of data saved [25]. In modelling the channel read and write processes, designing the equalisers and detectors, this feature has to be considered to have a good system.

2.2 Nature of Magnetic Media

The grains of a magnetic medium are the smallest unit of the magnetic medium which can assume a particular magnetisation as a whole. In order to rigorously investigate the media, a model is very important. As such various models sprang up during the development of the media. Among the models are Voronoi model[26][27], and simpler forms of it such as Four Grain model [28], and Grain Flipping Probability (GFP) model [11].

2.2.1 Voronoi Model

Magnetic medium modelling is broken into three basic parts, Modelling the grains, modelling the writing process and modelling the read process. Voronoi model assumes the medium is made up of grains represented by Voronoi regions of irregular shapes. The steps followed by [27] to create the medium model involves, first creating a random distribution of grain nuclei, with a desired average grain area (A), and a certain standard deviation (σ_A) of the area, which is greater than the final desired standard deviation. An iterative process is carried out to tune the standard deviation to the desired σ_A using the following steps as follows:

1. The grain nuclei are replaced by the grain centroid in order to smooth out irregularities in the distribution of the grain size.
2. Grains with size less than $A - 2\sigma_A$ are removed.
3. Grains of size more than $A + 2\sigma_A$ are split into two new grains putting the new grain nuclei slightly offset from the original nucleus.
4. Steps 1 to 3 are repeated until the average grain area and its standard deviation is sufficiently close to the desired target.

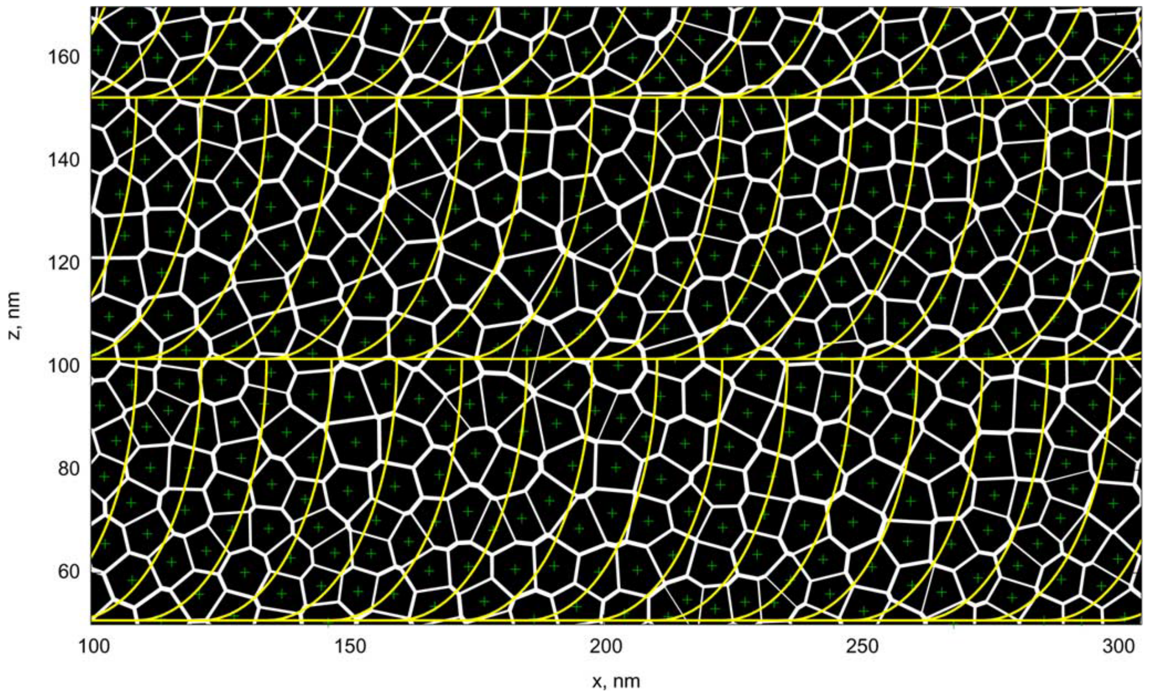


Figure 2.1: Example grain layout, with grain centroids (green), grain boundaries (white), and bit cell boundaries (yellow) with , $\sigma_A/A = 0.25$, 10 grains/bit, Ratio of bit width to its length(Bit Aspect Ratio=BAR)=4:1. [27]

The boundaries between the grains are assumed to be non-magnetic. And to model them, the grain areas obtained above are shrunk by a certain percentage (say 10%) inward toward the nucleus.

A picture of some example grains (black), with grain boundaries (white), bit cell boundaries (yellow) and centroids (green), is shown in Figure 2.1. The bit width in this example is roughly 12.5nm and length of 50nm.

The next problem after modelling the grain distribution of the disk is modelling the write process. This involves deciding which grain is magnetised by which bit and in what direction. The grains are assumed to initially have random magnetisations "x". A function is defined which gives the probability that a grain in a bit cell region is magnetised by the bit over or near it. The function represents a small probability that neighbouring grains are going to be overwritten by a bit that is currently written. That function is used to determine whether a grain is magnetised or not.

The read data ($y(t_1, t_2)$), at position "t_1" along a track and position "t_2" across the track, in this model is a convolution(written as "*" in this equation alone) of the magnetisation ($x(t_1, t_2)$) of each bit cell region and the read head response ($h(t_1, t_2)$) as shown in equation 2.1.

$$y(t_1, t_2) = x(t_1, t_2) * h(t_1, t_2) \tag{2.1}$$

The head transition response can be represented as a hyperbolic tangent function [29] [30] [31], an error function [32] [33] [34] or a Gaussian function [35]. Equation 2.2 and 2.3 show one dimensional version of the read head transition response ($s(\ )$) using $tanh(\ )$ and $erf(\ )$.

$$s(t) = V_{max} tanh\left(\frac{2t}{0.579\pi T_{50}}\right) \tag{2.2}$$

$$s(t) = V_{max} erf\left(\frac{0.954t}{T_{50}}\right) \tag{2.3}$$

where V_{max} is the maximum amplitude obtainable from an isolated response; "t" is the position or time separation of the magnetisation read with respect to the central position of the read head; and T_{50} is the time taken or bit length covered for the amplitude of the transition response to rise from $-V_{max}/2$ to $V_{max}/2$. Because speed of head is considered to be constant for any HDD, time taken from one position to another is proportional to the distance (distance=speed*time). This means "t" can represent either time or distance/position. These equations represent a single transition and figure 2.2 shows a plot of equation 2.2 and 2.3.

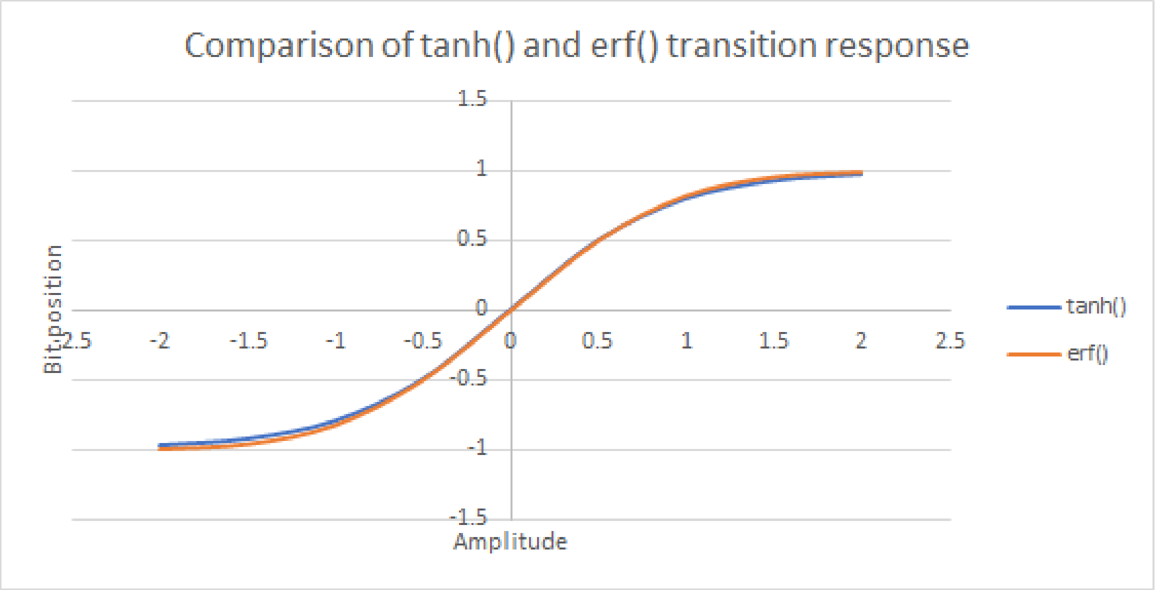


Figure 2.2: A comparison of transition responses based on tanh() and erf() with amplitude of $V_{max} = 1V$ and $T_{50} = 1B$, where B=bit period

A large value of T_{50} signifies large areal density, while a low value of T_{50} means low areal density. This relationship stems from the fact that isolated response of a bit reaches a maximum at the centre of the bit. Therefore the longer the bit along track (low density), the longer it takes to reach maximum amplitude. Shields around the read head are used to reduce the number of bits picked up by the read-head. The height of the shield from the disk surface is normally the same as the height of the read-head. The height reduces amplitude of the response if it is large but must be made large enough to avoid the head scratching the disk when there is downward position jitter. The shields are spaced from the read-head to avoid the shields diverting the magnetic field that is to be read into themselves. A balance is therefore found between interference and reduction of the effective field due to the proximity of shield. The relative change in resistance of the MR read-head also affects the amplitude of the response. High percentage change in resistance means the pre-amplifier can produce read current of larger amplitude [35] [36].

As can be seen from figure 2.2,for $T_{50} = 1B$ and $V_{max} = 1V$, using tanh function, the values of amplitude of a positive response(transition from a -1 to a +1) rises from -0.5V to 0.5V, within a time of 1 unit (-0.5B to 0.5B) where B is a bit period. This time interval is the definition of T_{50}.

To obtain an isolated response, a positive and negative transitions are super-imposed one bit period apart[37]. Equation 2.4 shows the two dimensional equation of isolated response using a Gaussian function with a Bessel function undershoot [27].

$$h(t_1, t_2) = \frac{exp(-r_1^2/2)}{2\pi\sigma_{t_1}\sigma_{t_2}} - \frac{K_0(r_2)}{2\pi l_{t_1} l_{t_2}} \tag{2.4}$$

where;

$$r_1^2 = t_1^2/\sigma_{t_1}^2 + t_2^2/\sigma_{t_2}^2 \tag{2.5}$$

and

$$r_2^2 = t_1^2/l_{t_1}^2 + t_2^2/l_{t_2}^2 + \epsilon \tag{2.6}$$

where "t_1" and "t_2" are the positions along and across track, σ_{t_1} and σ_{t_2} are used to set the 2D width of the central Gaussian part of the isolated response in the two given directions (it does the work of T_{50}). K_0 is Bessel function of the first kind of order 0 where, l_{t_1} and l_{t_2} are used to set the width of the "undershoot" response, and ϵ is a small valued figure used to prevent the Bessel function falling into singularity at $t_1 = t_2 = 0$ [27]. The Bessel function undershoot is added to guarantee that the frequency response at $\omega_{t_1} = 0$ and $\omega_{t_2} = 0$ is exactly zero. ω_{t_1} and ω_{t_2} are the wave numbers(from Fourier transform) of magnetisation waveform along and across track respectively. This means a uniformly magnetised medium ($\omega_{t_1} = \omega_{t_2} = 0$) will not have external field [35].

Small amount of White Gaussian noise is added to the convolution of magnetisation to channel response, which represents other electronic noises.

2.2.2 Grain Flipping Probability Model

Compared to regular Voronoi model, the Grain Flipping Probability (GFP) model tries to use a more realistic model for determining which grain is to be flipped. Rather than just deciding the magnetisation based on whether the nucleus of the grain falls under the read head or not; the GFP model uses a probability of flipping. The GFP is computed using micromagnetic simulation during medium characterisation and stored in a look-up table (LUT).

Micromagnetic simulation keeps track of the evolution of the magnetisation of each grain during the write process. It also associates a flipping probability to each grain. It is a very time consuming process but gives a more accurate probability of flipping, and consequently a more realistic model of the medium. The model presented by [11] avoids the task of constantly evaluating the GFP after each write process. The GFP is evaluated at the beginning, during the characterisation of the medium.

The steps involve: First, tensor and grain statistic information are generated for the Voronoi magnetic grains. Micromagnetic simulation is then run, given a head field, for a range of bit patterns using the data previously computed. This yields granular magnetisation of the grains. The statistics are used for characterising the magnetic medium, which is then stored in a multidimensional LUT. That information is used to flip the grains quickly.

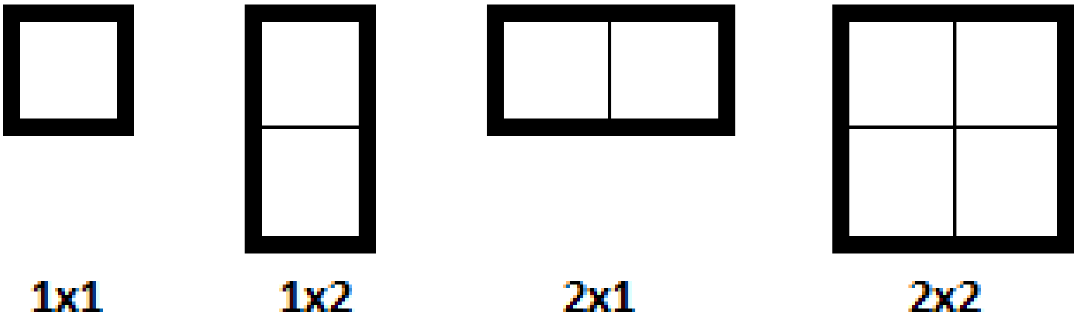


Figure 2.3: The 4 Grains

2.2.3 Four Grain Model

To make modelling of the medium easier, simpler models emerged from the Voronoi model. One of them is the Four Grain Model[26] [28] [38].

In this model, the grains are assumed to be of only four possible sizes, each of them with a probability of occurrence associated with it. The sizes are 1x1, 1x2 and 2x1 and 2x2. Depictions of the grains are shown in figure 2.3.

The magnetisation of the medium is decided based on the last bit that affected the grain nucleus. Figure 2.4 shows an illustration of writing on a 4 grain modelled medium as compared to an ideal medium.

2.2.4 Jitter Noise Model

Another model that tries to capture the effect of the irregular grains' sizes and irregular transitions due to the irregular grains is the jitter noise model.

Jitter is a deviation in the position of read-head or the boundary of a bit. If the read-head jolts forward perhaps due to movement of the HDD, it reads a bit ahead of the time it is supposed to read it.

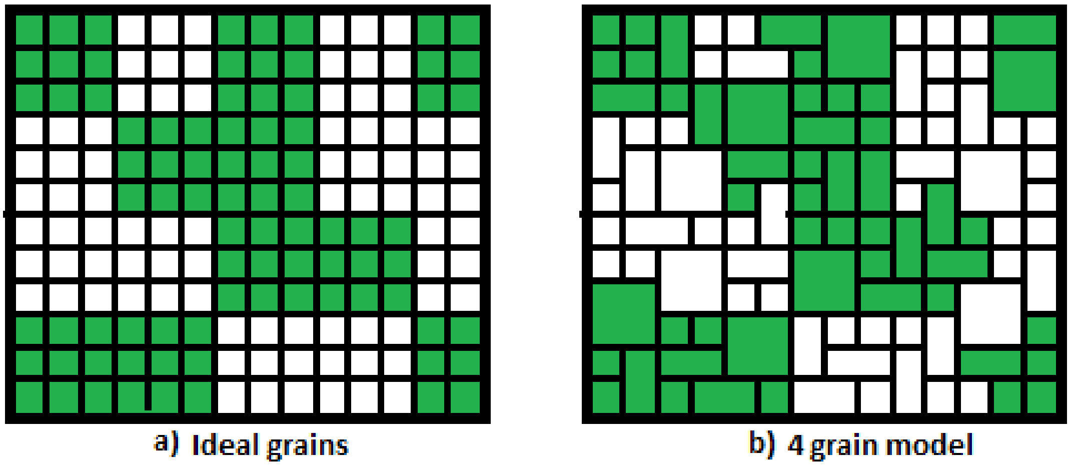


Figure 2.4: The Magnetisation of Ideal and 4 Grain Model

Similarly, if it jolts backwards, it reads the bit later than the time it should read the bit. This position jitter, therefore, results in changes in the timing of the bits. The irregularity of the grains also causes either to have the boundary of bit deeper into the area where the bit is supposed to be located, or the boundary to be located even before the position of the bit. The result is also a shift in the time at which the head registers the magnetisation of the bit. The timing circuit itself can have electronic noise that makes its reading randomly changed by small values. These deviations in timing due to the position, electronic noise or irregularity of magnetic grains is termed timing jitter. Therefore, they can be modelled as a random deviation in the time of transition in a read-head response. The changes in the value or magnitude of the read signal as a result of this timing jitter is called the jitter noise [39]. Figure 2.5 shows an example of how transition jitter affects response of read-head to saved data.

In jitter noise model, a random noise is added to the read back signal of each transition to represent either a shift of the transition in favour of or against the bits forming the transition. This can be represented as shown in equation 2.7 [37].

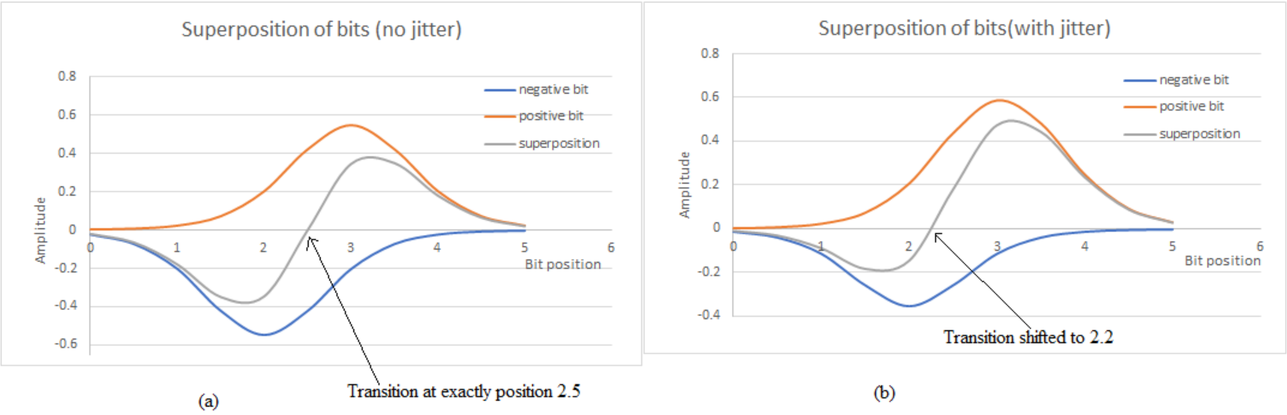


Figure 2.5: Isolated responses and their superposition. (a) Ideal isolated pulses with no jitter. (b) When jitter is present, transition point shifts

$$y(t) = \sum_i d_i s(t + a_i - iB) \tag{2.7}$$

where a_i is a random timing jitter noise; B is the bit period; $s(\ )$ is the transition response of the read-head; and "i" counts through all bits that have an ISI contribution to the signal at time "t", and d_i is bit transition which is $+1$ for positive transition(0 to 1), -1 for negative transition (1 to 0) or 0 for no transition(0 to 0 or 1 to 1). But only one value is sampled at each bit position. Therefore, the waveform that will be read is more of steps at each bit position, which ideally is desired to be either -1 or +1. Figure 2.6 shows an example of how interference and noise distorts the desired waveform.

Equation 2.7 can be expanded, using Taylor's series expansion or other means, to isolate pure interfered signal, jitter noise contribution, and white noise contribution as shown in equation 2.8.

$$y(t) = \sum_i d_i s(t - iB) + n_j(t) + n_w(t); \tag{2.8}$$

n_j is jitter noise, while n_w is white noise that emanates from electronic noises and other sources. The values of a_i is assumed to have White Gaussian (WG) distribution.

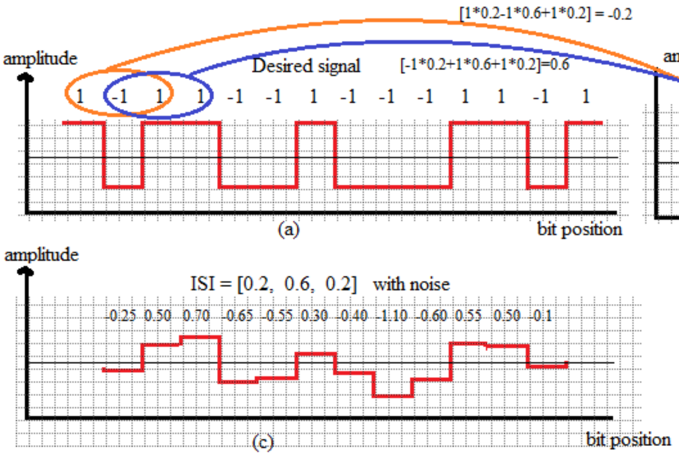


Figure 2.6: Effect of interference and noise on read signal. (a) Desired saved signal. (b) Interference from neighbouring bits changes amplitudes. (c) Noise (jitter or AWGN) further distorts the amplitude.

The boundaries at the beginning and end of a bit can have jitter in opposite direction such that the boundaries will move towards each other. If no care is taken in the simulation, they can move beyond each other, which results in a bit with negative area, something that is physically impossible. Because of that, limit is put on the maximum amplitude of jitter to prevent the scenario described here. The limit of amplitude of a_i is, therefore, chosen to be 0.5B. This means the worst bit area will be when the leading boundary has jitter of -0.5B while the trailing boundary has 0.5B. In this extreme case, the bit area is proportional to $0.5B - (-0.5B) = 0$. This is the equivalent of an erasure where, due to shape of the grains, a bit got erased when writing the bits that are in its neighbourhood. Because of this limit imposed on the magnitude of a_i, it is said to have truncated white Gaussian distribution. This models the net effect of the grain sizes and is computationally simple for simulation purposes.

2.3 Shingled Magnetic Recording

Write-head usually limits how small a bit can be on a magnetic medium. In order to get smaller bits, sharper or smaller write heads are needed.

This, as explained before, presents a problem of write-ability of the medium. Shingled Magnetic Recording(SMR) attempts to overcome the write-head limitation by overlapping bits and tracks. This means a wider write-head can be used to magnetise a portion of the disk by overwriting the undesirable portions. When writing the next bit along the same track or a neighbouring bit on the next track, portions of the written bit will be overwritten, except for the needed portion that will represent the data stored.

With this proposition by [12], the magnetic recording medium is expected to go beyond the $1Tb/in^2$ super-paramagnetic limit of current technology without a drastic change in the media technology [40].

The write head needs only to have one carefully designed corner of the head which will be leaving the data that will not be overwritten by the write head. Therefore, the data tracks are heavily overlapped, to utilise the usual space left as a guard band for data storage. The depiction of how the tracks are written can be seen in Figure 2.7.

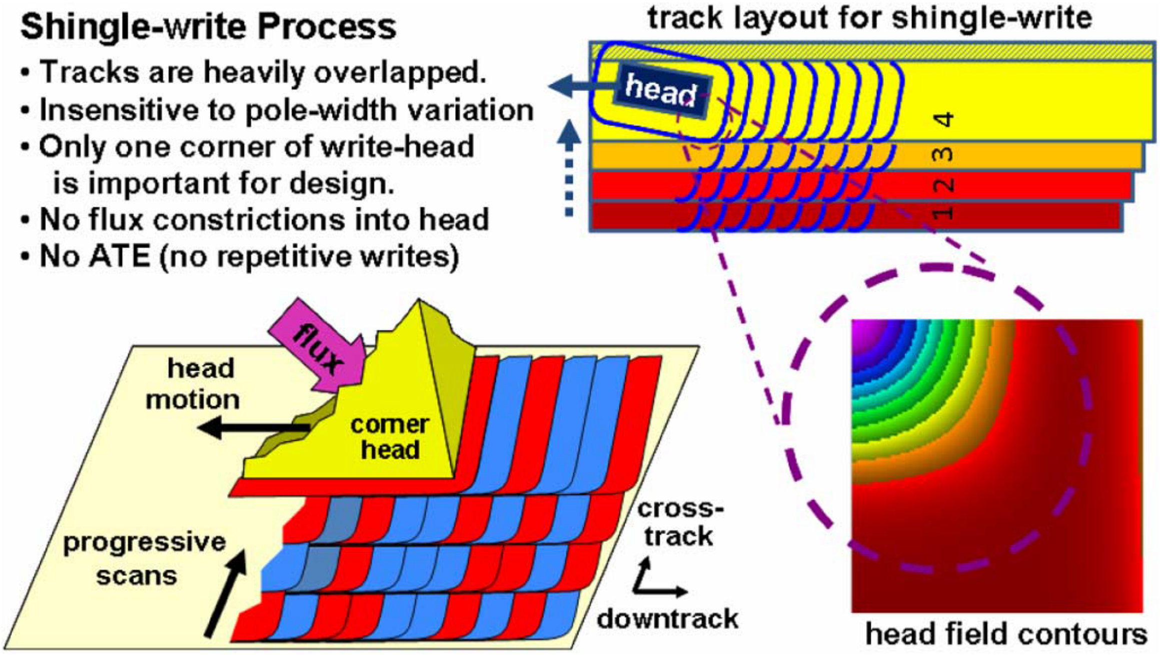

Shingle-write Process
• Tracks are heavily overlapped.
• Insensitive to pole-width variation
• Only one corner of write-head is important for design.
• No flux constrictions into head
• No ATE (no repetitive writes)
track layout for shingle-write
head
1
2
3
4
flux
head motion
corner head
progressive scans
cross-track
downtrack
head field contours

Figure 2.7: Shingled Writing [12]

The large areal density targeted by SMR and the closely packed tracks make Inter-Track Interference (ITI) very prominent as well as Inter-Symbol Interference (ISI).

In addition to the problem of increased interference, the system of SMR does not allow "update-in-place" for a bit or small group of bits that are less than a whole sector. This is because the write head will over-write neighbouring bits whenever a bit is to be written (except the last bit of a sector). The shingles will either have to be grouped into smaller sectors which will reduce the gain in the capacity of SMR; or put up with writing a whole sector when a portion of it will be updated or modified, which will slow down the writing speed in case of updating. A new data management protocol will, therefore, be needed for the operating system to work with the new system [41] [42] [43].

The capacity of the medium can therefore only go beyond the $1Tb/in^2$ current technology limit as suggested in [12] [44] only if a robust signal processing is used to get the information out of the interferences and noise. Two Dimensional (2D) signal processing is advanced by researchers as the most appropriate way of keeping the integrity of the data at its best.

2.4 Equalisation Techniques

Different techniques used for equalisation of the read information from a magnetic medium have been proposed to be used in SMR equalisation. Among the techniques suggested for PMR and are possibly useful in SMR are Zero Forcing (ZF) equaliser , Minimum Mean Square Error (MMSE) equaliser [45], 1D and 2D Partial Response (PR) target equalisers.

2.4.1 Zero Forcing

ZF equaliser (inversion detector) is a simple linear equaliser/detector that detects the data directly from the signal. It does so through algebraic evaluation of the signal. It simply inverts or evaluates the equations or expressions that define the inter-relationships of the symbols [46].

Let the signal read by the read-head located at position "t_1" along track and position "t_2" across track be represented by $y(t_1, t_2)$. Let the magnetisation saved at a position "τ_1" units away from the read-head along track, and "τ_2" units away from the read-head across track be represented by $x(t_1 - \tau_1, t_2 - \tau_2)$. Let the response of the read-head due to a magnetisation at position $[\tau_1, \tau_2]$ away from the head be represented by $h(\tau_1, \tau_2)$. If the Additive White Gaussian Noise (AWGN) picked by the read-head at the head position $[t_1, t_2]$ is given by "$n(t_1, t_2)$", then the received signal ($y(t_1, t_2)$) can be represented using equation 2.9.

$$y(t_1, t_2) = \int_{\tau_1} \int_{\tau_2} h(\tau_1, \tau_2) x(t_1 - \tau_1, t_2 - \tau_2) d\tau_1 d\tau_2 + n(t_1, t_2) \quad (2.9)$$

Equation 2.9 gives the contributions, at positions "t_1" and "t_2", of all the magnetisations (x) from positions positions $[\tau_1, \tau_2]$ in a continuous time mode. The relationship can be broken into discrete form, which is the normal mode in which the medium is sampled.

To represent the data in discrete form, let k_1 and k_2 represents the positions of bits on the medium along the tracks and across the track in units of bit period. Let "t_1" and "t_2" be the position at which the response is determined, with read-head response given by $h(k_1, k_2)$ and the data saved be "x". Then the discrete signal to be received by the read head $y(t_1, t_2)$ can be expressed as in equation 2.10 [47] [48] [49].

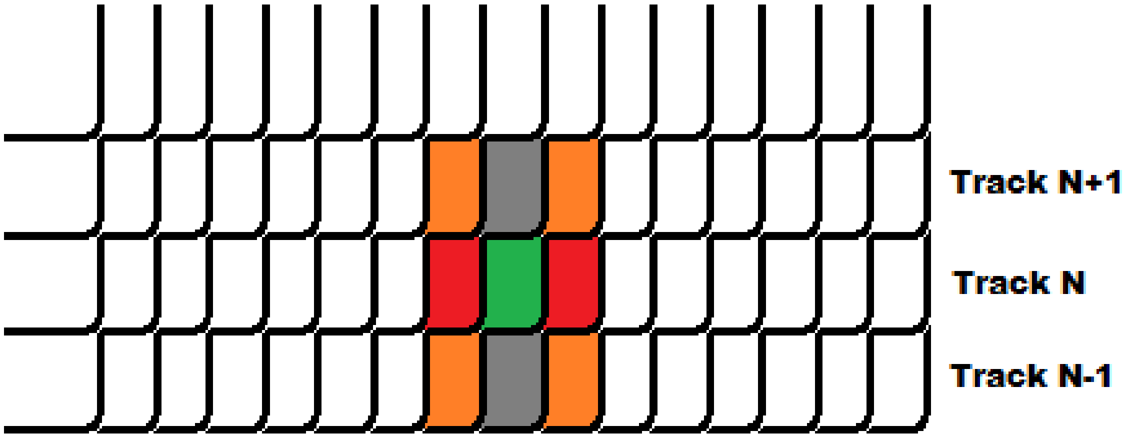


Figure 2.8: The bit intended to be read (green) affected by bits ahead and behind it(red). Also affected by neighbouring bit in track before and after it(grey) which were affected by bits ahead and before them(orange).

$$y(t_1, t_2) = \sum_{k_1} \sum_{k_2} h(k_1, k_2) x(t_1 - k_1, t_2 - k_2) + n(t_1, t_2) \tag{2.10}$$

In a situation where most of the interference comes from a few neighbouring bits, range of k_1 and k_2 can be reduced to the few bits in equation 2.10, for simplicity and reduced computational complexity.

When the most significant interfering bits are the bit before and after the central bit, the bits adjacent to those ones on the track before and after the central track; then there will be 8 bits interfering with the central read bit, as shown in figure 2.8. The range of k_1 and k_2 will, therefore, both be -1 to +1 if "t_1" and "t_2" are both considered to be 0. Equation 2.10 can therefore be written as shown in equation 2.11

$$y(0, 0) = \sum_{k_1=-1}^{1} \sum_{k_2=-1}^{1} h(k_1, k_2) m(k_1, k_2) + n(0, 0) \tag{2.11}$$

In order to evaluate the result from the signal using a zero forcing equaliser, the signals from the interfering bits are read and used to cancel the interference.

The equalisation can be done using a 2D equaliser (of 9x9 taps in the case above) or preferably and for simplicity, two 1D equalisers (each of 3x3 taps in the case above) used one after the other. If the expression is written in Matrix equation form where; "H" is the matrix of coefficients from the channel response, "X" is matrix formed from the saved data at the read position, "N" is the matrix of electronic noise picked up when reading the data and "Y" is a matrix of the received signal; then the received signal can be expressed as shown in equation 2.12 [50]

$$Y = HX + N \tag{2.12}$$

To evaluate the approximate values of "X", the equation can be multiplied by the matrix inverse of "H" (hence the name inversion detector) as shown in equation 2.13.

$$\hat{X} = H^{-1}Y = X + H^{-1}N \tag{2.13}$$

In situations like SMR, where the ISI and ITI interferences extend a long way, the formation of H matrix necessarily needs approximations. For a 1D equaliser, from equation 2.10, the received signal at position "t" can be written as:

$$y(t) = .. + h(t-1)x(t-1) + h(t)x(t) + h(t+1)x(t+1) + .. \tag{2.14}$$

Therefore
$y(t+1) = .. + h(t-1)x(t) + h(t)x(t+1) + h(t+1)x(t+2) + ..,$
$y(t+2) = .. + h(t-1)x(t+1) + h(t)x(t+2) + h(t+1)x(t+3) + ..,$
and so on.

The expressions for the signal can, therefore, be represented in matrix form as shown in equation 2.15. Note that h_i is equivalent to $h(t+i)$ and similarly $y(t)$ and $x(t)$ are replaced by y_t and m_t respectively.

$$\begin{bmatrix} \vdots \\ y_{t-2} \\ y_{t-1} \\ y_t \\ y_{t+1} \\ y_{t+2} \\ \vdots \end{bmatrix} = \begin{bmatrix} ... & \vdots & \vdots & \vdots & \vdots & \vdots & ... \\ ... & h_0 & h_1 & h_2 & h_3 & h_4 & ... \\ ... & h_{-1} & h_0 & h_1 & h_2 & h_3 & ... \\ ... & h_{-2} & h_{-1} & h_0 & h_1 & h_2 & ... \\ ... & h_{-3} & h_{-2} & h_{-1} & h_0 & h_1 & ... \\ ... & h_{-4} & h_{-3} & h_{-2} & h_{-1} & h_0 & ... \\ ... & \vdots & \vdots & \vdots & \vdots & \vdots & ... \end{bmatrix} \begin{bmatrix} \vdots \\ x_{t-2} \\ x_{t-1} \\ x_t \\ x_{t+1} \\ x_{t+2} \\ \vdots \end{bmatrix} \tag{2.15}$$

The interference gets weaker as one moves further away from each bit. Therefore, depending on the density of the data, most of the bit energy of each position is spread around it. This means the number of taps of the equaliser can be truncated to the length that will include most of the energy.

If say in a channel, whose head response is symmetric about the central bit, there are "p" bits each before and after the central received signal to be equalised, and "q" bits each before and after the saved signals that will be estimated, equation 2.15 can be approximated and truncated to equation 2.16 as shown.

$$\begin{bmatrix} y_{t-p} \\ y_{t-p+1} \\ \vdots \\ y_t \\ y_{t+1} \\ \vdots \\ y_{t+p} \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & .. & h_q & .. & h_{2q} \\ h_{-1} & h_0 & .. & h_{q-1} & .. & h_{2q-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{-p} & h_{1-p} & .. & h_{q-p} & .. & h_{2q-p} \\ h_{-p-1} & h_{-p} & .. & h_{q-p-1} & .. & h_{2q-p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{-2p} & h_{1-2p} & .. & h_{q-2p} & .. & h_{2q-2p} \end{bmatrix} \begin{bmatrix} x_{t-q} \\ x_{t-q+1} \\ \vdots \\ x_t \\ \vdots \\ x_{t+q} \end{bmatrix} \tag{2.16}$$

For a stable result the "H" matrix should be invertible. This means $q \leq p$. From the equation above, truncated H can, therefore, be written to be as shown in equation 2.17.

$$H = \begin{bmatrix} h_0 & h_1 & .. & h_q & .. & h_{2q} \\ h_{-1} & h_0 & .. & h_{q-1} & .. & h_{2q-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{-p} & h_{1-p} & .. & h_{q-p} & .. & h_{2q-p} \\ h_{-p-1} & h_{-p} & .. & h_{q-p-1} & .. & h_{2q-p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{-2p} & h_{1-2p} & .. & h_{q-2p} & .. & h_{2q-2p} \end{bmatrix} \tag{2.17}$$

As an example, if $p = q = 2$, it means the taps of the equaliser is 5 (= 2p+1) and the bits that can be estimated are 5 (= 2q+1). The H matrix can, therefore, be written as shown in equation 2.18.

$$H = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & h_4 \\ h_{-1} & h_0 & h_1 & h_2 & h_3 \\ h_{-2} & h_{-1} & h_0 & h_1 & h_2 \\ h_{-3} & h_{-2} & h_{-1} & h_0 & h_1 \\ h_{-4} & h_{-3} & h_{-2} & h_{-1} & h_0 \end{bmatrix} \tag{2.18}$$

Zero forcing is a simple direct solution with very small computational complexity but has a major shortcoming of noise amplification. The noise "N", is amplified by a factor "H^{-1}" as can be seen in equation 2.13. This amplification is severe when there is heavy interference, as obtainable in SMR medium. This makes ZF sub-optimal and therefore not suitable for high-density magnetic recording [51].

2.4.2 Minimum Mean Square Error

Rather than cancelling all Inter-Track Interference(ITI) and Inter-Symbol Interference(ISI) completely, Minimum Mean Square Error (MMSE) tries to make the mean square error between the estimated and saved data minimum, even if it means allowing some ISI and or ITI. This can be achieved by treating the magnetic channel as a Multiple Input

Multiple Output (MIMO) channel [52] [53].

For linearly combined MIMO channel, equaliser that minimises the mean square error can be represented by equation 2.19 [50], [54].

$$\hat{X} = (H'H + \sigma^2 I)^{-1} H'Y = (H'H + \sigma^2 I)^{-1} H'(HX + N) \quad (2.19)$$

where σ^2 is the variance of the noise "N" on the channel, I is an identity matrix and H' is a complex conjugate of H.

After determining the equaliser coefficients, the MMSE equaliser is treated in the same way as ZF equaliser. Which means two 1D equalisers can be used in series or a single 2D equaliser used.

Even though MMSE performs, better than ZF, its result is also not optimal. Generally, linear detection techniques do not determine the most likely bit or sequence of bits. Therefore, linear equalisers are used to filter and prepare the signal for detection while more complex non-linear techniques are employed for detection.

2.4.3 Partial Response Equaliser

Removing all the interferences, as in ZF, has the limitation of amplifying noise too much, whereas leaving some of the interference as in the case of MMSE reduces SNR of the signal. Both techniques do not seek for the most likely saved signal. Therefore, a more rigorous search for the most likely saved data is normally conducted in order to get a better result [55].

Partial Response (PR) equalisers try to reduce the signal to a selected target. A target is selected such that equalisation to such target will have as little amplification of noise as possible. The target should also be good for an efficient Maximum Likelihood (ML) detection. The PR equaliser limits or reduces the spread of the interference to the given selected target length. This helps in limiting the complexity of the ML detector.

There are different types and families of target responses in PR equalisation and detection. They are usually defined by the discrete delay polynomial (function). In the system, a delay of single bit period can be represented by D. D^2 is a delay of 2 bit period. Therefore, D^k is a delay of k bit period. Various powers of D are added to get the delay polynomial. For example, 1+D means the summation of the current symbol to the symbol (1 bit period delay) before it.

It is worth noting that, Magneto-Resistive (MR) read heads, as used in Longitudinal Magnetic Recording (LMR), detects bit transitions (vertical components of magnetisation), not the absolute value of the bit. Therefore, when a sequence of bits say "0100" is to be read, the head will detect that there is a change from bit 0 to bit 1 (i.e. 1-0 or positive bit transition). In the next read period, it will detect a change from 1 to 0 (i.e. 0-1 or negative bit transition), and the next will be 0 to 0 (no bit transition). In other words, the binary data with two symbols is now read in a system with 3 possible symbols (+1, 0 and -1). This is equivalent to modifying the data by the polynomial 1-D, which is the differentiation of the magnetisation of the disk. Therefore, 1-D is inherent in the read head system of LMR. The target polynomials are therefore always attached to this inherent polynomial.

The PR4 family of targets are based on the powers of 1+D multiplied by the inherent 1-D. Table 2.1 shows a few of the targets.

The targets are expressed a little different in Perpendicular Magnetic Recording (PMR). This is because the MR read head detects the actual bits' magnetisation field (perpendicular component), and passes through zero during transition [49]. Therefore, other more suitable tar-

Table 2.1: PR4 family of target polynomials

Name	Target Polynomial	Isolated Pulse Response
PR4	$(1-D)(1+D) = 1-D^2$	1 1
EPR4 (Extended PR4)	$(1-D)(1+D)^2 = 1+D-D^2-D^3$	1 2 1
E^2PR4	$(1-D)(1+D)^3 = 1+2D-2D^3-D^4$	1 3 3 1

get responses are investigated and [56] shows that the target response can be extended to the general form

$$G(D) = (1-D)(1+D)^P(D^Q-1). \tag{2.20}$$

where $(1+D)^P$ represent the impulse response of single bit and (D^Q-1) represent the superposition of pulses. In general, Generalised PR (GPR) targets can have any form, but examples of the forms presented by [56] are shown in table 2.2

Table 2.2: Other PR coefficients

	P	Q							
PMR1	1	2	-1	0	2	0	-1		
PMR2	1	3	-1	0	1	1	0	-1	
PMR3	2	2	-1	-1	2	2	-1	-1	
PMR4	2	3	-1	-1	1	2	1	-1	-1

The selection of target was investigated by [45] and a MMSE method was presented. This method tries to maximise the ratio of the minimum squared euclidean distance of the PR target to the squared noise penalty introduced by the filter. This helps find an optimal target without resorting to noise prediction.

In order to implement PRML, the input data (in LMR) is first pre-coded in order to prevent chain errors that may occur due to memory the channel has. For PR4 channel, the pre-coding equation is $1/(1+D^2)$. The data is then written on the medium.

2.4.4 2D PR target equaliser

In situations where a 2D detector is used for the detection, the system will also need a 2D equaliser.

A method of implementing 2D GPR equaliser is presented by [57]. Given that a saved 2D signal at time "k" is expressed as a column vector, and is represented by x_k. Given that a 2D received signal for the same time is expressed as a column vector, and is represented by y_k. Given the 2D target response, expressed in matrix form, is converted to a column vector, and is represented by g^T where exponent of T here means transpose. If the equaliser coefficients, also converted into column vector, is represented by f^T, then the error (e_k) between equalised signal and the saved data, convolved with the target, can be expressed as shown in equation 2.21

$$e_k = f^T y_k - g^T x_k \tag{2.21}$$

The mean square error (MSE) can, therefore, be obtained as,

$$E(e_k^2) = f^T R f - 2f^T T g + g^T A g \tag{2.22}$$

where $A = E(x_k x_k^T)$ is the autocorrelation matrix of the channel input;
$R = E(y_k y_k^T)$ is the autocorrelation matrix of the channel output;
and $T = E(y_k x_k^T)$ is the cross-correlation matrix of the channel output and input.

A trivial solution exists when $f = 0$ and $g = 0$. In order to avoid that, while trying to minimise the MSE, a constraint on "g" was imposed to make sure certain entries of "g" have some specific values. The constraint is given by equation 2.23.

$$E^T g = c \tag{2.23}$$

where E^T is a matrix with a number of rows equal to the number of entries in "g" that doesn't have to be optimised. Whereas "c" is the column vector resulting from the application of the constraint. As an example from [57], assume a 3x3 target coefficient vector "g" is constrained as

$$g = [g_{-1,-1}, g_{-1,0}, g_{-1,1}, g_{0,-1}, 1, g_{0,1}, 0, 0, 0]^T.$$

From this, we can obtain the matrix of constraint (E^T) as

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

and the resulting column vector "c" is

$$c = [1000]^T.$$

Applying the constraint and evaluating its Lagrange function for minimising the MSE, the equation yields the following function in equation 2.24.

$$J = f^T R f - 2f^T T g + g^T A g - 2\lambda^T (E^T g - c) \tag{2.24}$$

where "λ" is a column vector containing Lagrange multipliers.

Further calculations yielded the optimised target and equaliser coefficient vectors as follows:

$$\lambda = \left(E^T (A - T^T R^{-1} T)^{-1} E\right)^{-1} c \tag{2.25}$$

$$g = (A - T^T R^{-1} T)^{-1} E \lambda \tag{2.26}$$

$$f = R^{-1}Tg \tag{2.27}$$

2.5 Detection Techniques

After preparation of the received signal by using PR equalisation and or filtering, the signal is utilised by detectors to get the data out of the equalised signal. Maximum Likelihood (ML) detection is usually used to determine the data. The ML detection can be either Maximum Likelihood Sequence Detection (MLSD) or Maximum A-posteriori Probability (MAP) detection.

2.5.1 Maximum Likelihood Sequence Detection

MLSD is a detection or decoding technique used to get a sequence of bits that has the most likelihood of occurring depending on the received data. The most popular MLSD detector used in magnetic media is the Viterbi Algorithm (VA).

VA was introduced in 1967 as a means for decoding convolutional codes. In 1972, it was shown by Forney that the MLSD problem for the channel with ISI and AWGN is solved by VA [58] [59]. It was later recognised as useful in ML detection of magnetic recording systems with or without PR equaliser. The combination of PR equalisers and VA detectors in magnetic storage systems for mitigating ISI effects gave birth to many commercially successful products [60].

In its basic form, Viterbi detector is a maximum likelihood detector that determines the most probable sequence of data in binary form (0s and 1s). It does that by finding a valid sequence of bits with minimum distance from the received data.

The detector can functionally be divided into three sections. The Branch Metric Unit (BMU), Add Compare Select Unit (ACSU) and the Trace-Back Unit (TBU). Figure 2.9 shows the arrangement of the VA units.

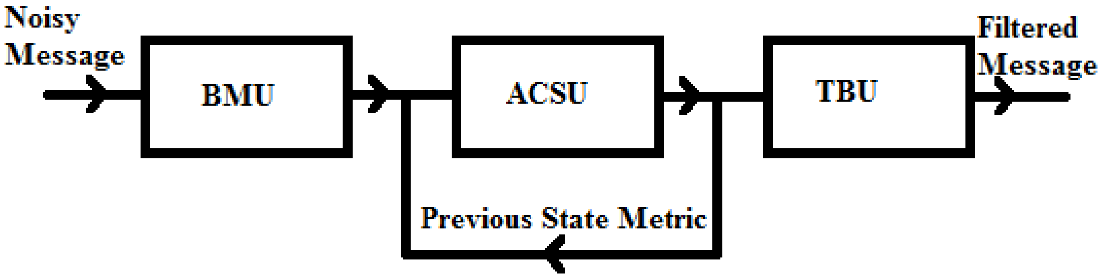


Figure 2.9: Functional units of Viterbi Algorithm. Branch Metric Unit(BMU), Add Compare Select Unit(ACSU) and Trace-Back Unit(TBU).

In the BMU at time "t", the distance of the received signal (y), from all the possible number (r) of saved/transmitted signals (x_r), is calculated as the distance metric (DM). If the received data is in binary form, the Hamming distance is used to find the DM. Hamming distance is the number of differences in the bits of the received and expected/possible data. If the received data is continuous or truncated, the Euclidean distance is used to find the DM as expressed in equation 2.28.

$$BM(t,r) = (y_t - x_r)^2 \tag{2.28}$$

Each possible state of the VA is associated with a metric called state metric (SM). SM shows the likelihood that the system is presently in that state. It is the sum of all the DMs of the symbols that lead to that particular state. Initially, the SM is initialised to be the same for all states if the system can possibly start from any of them. In the case where the starting state is initially known, it is given a favourable SM at the start.

In the ACSU, the DM of every possible branch is added to the SM of the state from which it originates to get the branch metric, BM (**ADD**). The BMs going to the same state are then compared (**COMPARE**) to find the one with the smallest distance. The BM of the closest branch (smallest distance) is selected and saved as the present SM of the terminating state (**SELECT**). A history is kept in the state, to indicate the branch selected for use in the next unit in the determination of data. This process is repeated until a certain length of data (trace-back length), or the whole data is processed.

In the TBU, the SMs are compared to see the state with the most favourable(minimum) SM. The state with the minimum SM is selected as the starting state for trace-back. If the trace-back length is adequately long, the trace-back can be started from any state. This is because the trace-back converges to the same state, most of the time, after some number of steps. A portion of the traced-back data from the history is saved as the detected data [61].

A variation of Viterbi detector is used to give soft output for usage by another decoder or detector. This variation is called Soft Output Viterbi Algorithm (SOVA). SOVA gives out a result that represents the reliability of the decision on whether a zero or a one is the most probable output. Therefore, it works on two level symbol (bit) which is either 0 or 1 [62].

The complexity of VA is fundamentally determined by the number of states and branches per state. A branch or an edge is a possible transition from one state to another. In a binary system, there are two possible destinations from each state. The destinations correspond to an input of 0 or an input of 1. This means each state will also have two possible sources which can terminate into it. This is to say we have two edges per state. Multi-bit input, with 2 bits, have four edges per state and system with "b" bits input will have number of edges N_e per state given by

$$N_e = 2^b \tag{2.29}$$

The number of states is determined by the length of the memory coupled with the number of bits per input. For a system with target length "k", the memory of the VA is given by $m = k - 1$ (constraint length). The number of states for this system with "b" bits per input is:

$$N_s = 2^{bm} \tag{2.30}$$

This means total number of edges is $N_{total} = N_s N_e$, which is equivalent to:

$$N_{total} = 2^{bk} \tag{2.31}$$

The complexity is directly proportional to the number of edges. This is because, for each edge, BM must be determined, a group of "N_e" BMs must be compared in each of the "N_s" states to get SMs. Those SMs may have to be compared to determine the best state from which to trace-back data from [61].

2.5.2 Maximum A-posteriori Probability Detection

Another ML detection method that can be used in magnetic medium detection is the Maximum A-posteriori Probability (MAP) detection. It tries to reduce the probability of error in the detection of any given symbol. A trellis-based implementation of MAP was first proposed by [32] as an alternative to Viterbi decoding. It is now popularly called BCJR (Bahl, Cocke, Jelinek, and Raviv). While Viterbi decoders reduce the possibility of a word (sequence) error, BCJR decoders/detectors reduce the probability of bit or symbol error to the minimum. It can be used with convolutional codes or other blocks codes. It is later used in equalisation problems such as in the Shingled Magnetic Recording (SMR) [63].

The probability of an error in a symbol, given the received signal and the signal around it, is determined for each symbol. That is $p(y = 0/x = 1)$ and $p(y = 1/x = 0)$. Bayes theorem is used to calculate the a-posteriori probability of error given a zero or a one was saved on the medium.

If the noise is additive white Gaussian (AWGN), the probability of receiving y, given x was transmitted through the channel is given by:

$$p(y/x) = \frac{\exp(-\frac{(x-y)^2}{2\sigma^2})}{\sqrt{2\pi\sigma^2}} \tag{2.32}$$

where σ is the standard deviation of the noise, x is the transmitted (saved) symbol, and y is the received (read) symbol. Normally for any channel at a given time, the σ is the same or approximately equal for all possible sampled symbols. Therefore σ and other constants can be removed from the calculation to simplify the process. This can, therefore, be reduced to

$$p(y/x) = \exp(-\frac{(x-y)^2}{2\sigma^2}) \tag{2.33}$$

But when equation 2.33 is used, there must be regular normalisation, to make sure the total probability is always equal to 1.

In MAP algorithm, forward recursive probability (α), backward recursive probability (β), and transition probability (γ) are utilised to find the most probable symbol or bit. The α for all states is initialised, with the first state having a probability of 1 and the rest having probabilities of 0 if the sequence is known to start from the first state. The β is initialised such that all states have equal probabilities which sum up to 1 if the sequence can end in any state.

When a symbol "y" is received at a given instance, the transition probabilities (γ) of all edges for each state, are calculated according to equation 2.33. Those transition probabilities are then used to calculate the forward recursive probability for the next symbol using equation 2.34.

$$\alpha_{t+1} = \sum_{s} \alpha_t \gamma^{p,q} \tag{2.34}$$

where "p" represents the present state, "q" represents the next state and "s" means summation over all states that lead to the state for which we are determining the α value. Also "t" signifies the time (symbol position). The step above is repeated for the next symbol and so on until the last symbol is reached in the selected block.

The backward recursion then starts in a similar manner from the last bit by calculating the β for each symbol using equation 2.35.

$$\beta_{t-1} = \sum_{s} \beta_t \gamma^{p,q} \tag{2.35}$$

When both the α and β values of any symbol are determined, the A-Posteriori Probability (APP) of the symbol can be determined using equation 2.36.

$$p(x) = \sum_{s_x} \alpha_p \gamma^{p,q} \beta_q \tag{2.36}$$

The term s_x means summation over states that infer a bit "x" is stored. This way the probability of 0 and 1 are calculated if the data is binary. The bit with the largest probability is selected as the detected bit. A more detailed explanation can be found in [49] [64].

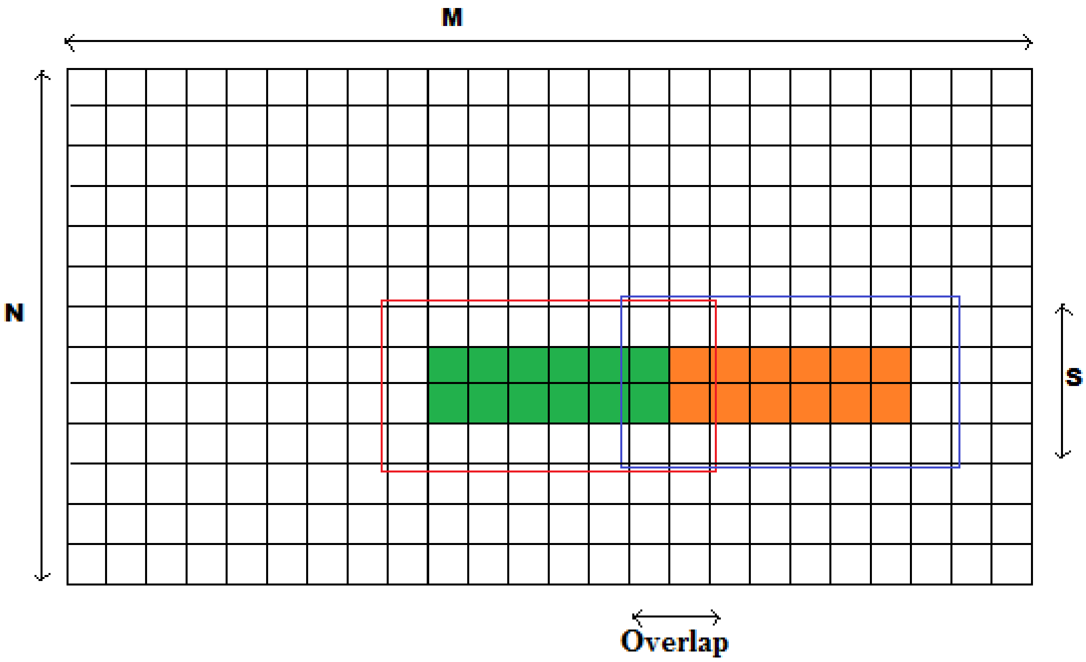


Figure 2.10: Red rectangle encloses symbols used to determine green symbols. Blue rectangle encloses symbols used to determine orange symbols. Red and blue rectangles overlap in two band of bits

In [12] this technique is applied to TDMR system. It was noted that the complexity of the ML detection is exponential with respect to the number of rows in a group or cluster (M). Therefore [12] proposed an approximation that divides the whole block (sector) into smaller strips of manageable size. Figure 2.10 shows a depiction of such strategy, where 24x13 block of 2D data is divided into 8x4 strips (two overlapping strips shown) with 6x2 data estimated in each strip (coloured bits). If one bit is desired in each strip, the strip can be reduced to 3x3 or even less for complexity reasons.

Pre-computations are normally used to reduce the amount of computations needed in BCJR, VA and other detection techniques. In [65] the algorithm was simplified using 7 bits per strip using VA as a ML detector. Unnecessary calculations are identified and eliminated while carrying out all calculations that can be done prior to reception of data before detection starts.

2.5.3 Partial Response Maximum Likelihood

Partial Response Maximum Likelihood (PRML) detection is a combination of PR equalisation and ML detection. PRML detection was first introduced in the 70's by IBM [8] for use with LMR. This is later used in the PMR media, which helped in increasing the areal density, as compared to threshold detection.

PRML is simply the use of PR equaliser to reduce the interference in the received signal to a manageable amount while taking care not to cause excessive noise amplification. A ML detector, mostly VA, BCJR or other less used methods such as Neuro ITI canceller, is then used for final detection of the saved data. This technique is used in almost all magnetic detectors [66] [67].

2.5.4 Full 2D detection

Due to its complexity, full 2D detection was not receiving much attention until recently [38] [57] [68]. The Multi-track Joint 2D detection simultaneously detects multiple tracks. It uses signals from the tracks which were equalised using the 2D GPR target equalisers. According to [57], in order to simultaneously detect three tracks, five tracks (including the tracks beside the main tracks) are read and used to get equalised data for the three main tracks. A full 2D Viterbi detector or its soft output version is used to extract the bits from the tracks simultaneously. Figure 2.11 shows the stages of the process.

In [38] it was shown that there is some improvement over the usage of 1D detector but the gain reduces as the data density increases. Also [68] presented a result for relatively low density system, and was able

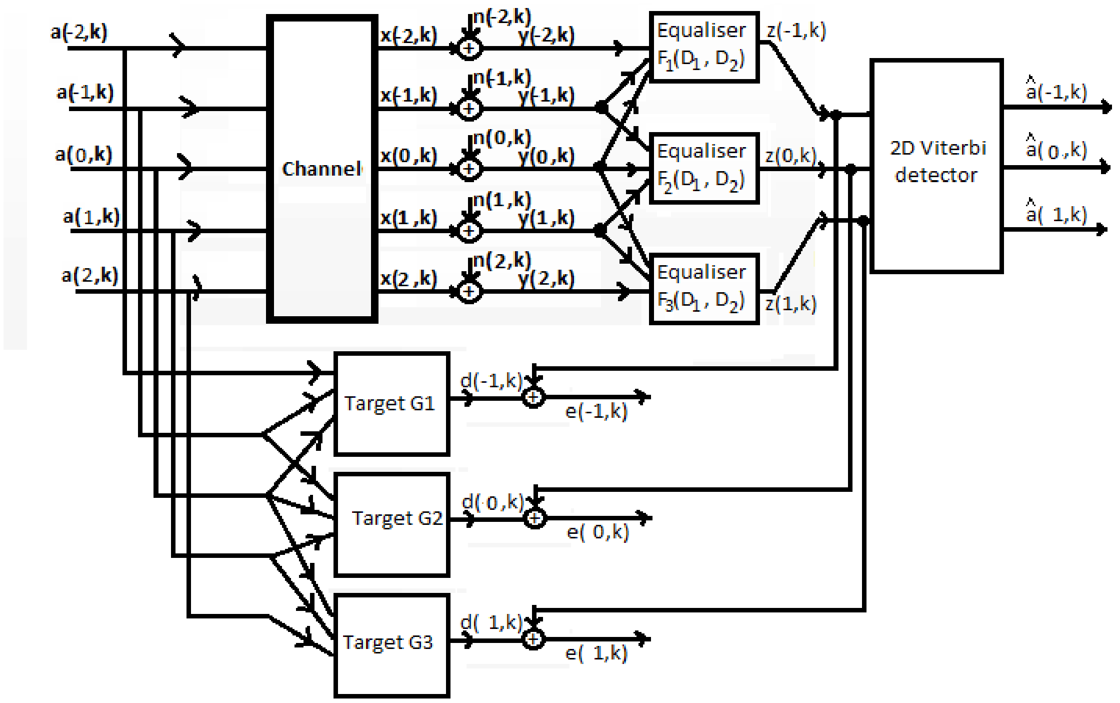


Figure 2.11: Joint Track Equalisation. a=binary data, x=saved data, y=read noisy data, z=equalised data, d=dibit response, e=error, and $\hat{a}$=detected data. [57]

to record a considerable improvement over the use of 1D detector. This suggests that further improvement is required in order to make use of full 2D detectors for high density SMR.

2.6 Forward Error Correction

The aim of Forward Error Correction (FEC) is to add some information (redundancy) to data before it is transmitted or saved, so that when an error occurs, it can be identified and possibly corrected at the receiving end. The redundant information can be a repetition of the original data or a result of a more complex function of the original data.

This feat is achieved in binary data mostly by using parity bits. A single parity system is a system that adds a single bit in order to make the number of 1s even (even parity) or be made odd (odd parity). More complex parity codes add more than one parity bit to a block or group of data in order to make sure the whole code satisfy some defined parity equations [69].

There are two main types of FEC coding. The block codes and the convolutional codes [70].

Block codes take a given block of bits and add a certain number of parity bits to it. The block length can vary from a few bits (eg. Hamming 7,4 code) to thousands of bits (as in Low Density Parity Code; LDPC). To determine the output reliably, the whole block must be processed together.

On the other hand, convolutional codes, use a repetitive coding technique over a short length of the data. The same coding formula is used over the fixed short length of the data repetitively, taking steps of 1 or more bits at a time.

Reed Solomon (RS) codes were the most popular codes used for error correction in HDD [37] [71] [72]. But in recent times, LDPC code is gaining popularity and it is driving the capacity of the disks to near Shannon limit [13] [73] [74].

Very often, the error correction coding becomes very complex for an optimal performance to be achieved, and therefore concatenation of more than one simple codes is used to achieve a better result, at the same time have simpler decoding algorithm.

The concatenated codes are normally separated by an interleaver. Interleaver is an algorithm that rearranges the data, to randomise the errors, which may be in clusters. Clusters of errors normally occur after decoding of the first code is done [75].

Different types of interleavers are available. Among them is the simplest called Matrix interleaver. This interleaver arranges the block of data in matrix form row after row and then reads it column after column. Another type of interleaver is the random interleaver. It creates pseudo-random numbers for the addresses to which each bit is taken. A version of the random interleaver that performs very well with turbo codes is called the Dithered Relative Prime (DRP) interleaver. DRP interleaver breaks the interleaving process into 3 stages. The whole data is first broken into smaller blocks of equal length which are each randomised. This is called the read dither. The whole resulting data is then randomised using a relative prime linear congruential generator. Finally, the resulting data is broken into other smaller blocks and each is randomised again. This is termed the write dither [76].

2.7 Summary

The discussions in this chapter highlighted how the nature and current technology of magnetic HDD affect the drive to increase the capacity of HDD and how one of the suggested technologies(SMR) is trying to overcome the challenges. We have seen that at high density, magnetic HDD is dominated by jitter noise, inter-track interference and inter-symbol interference. These two dimensional interferences necessitate the use of two-dimensional magnetic recording techniques in order to get a desirable performances at the targeted densities of the proposed technology. Researchers have used 2D equalisers together with 1D detectors or full 2D detectors to tackle the problems. In the rest of this report, implementation of the techniques that use equalisers, equalisers with 1D maximum likelihood detector, and equalisers with full 2D detectors are presented to address the issue of complexity of detectors. Modifications in the use of the detectors or their structure is presented and the results are presented, compared and analysed for situations in which we can use simpler or more complex detectors.

Chapter 3

MODELLING

This chapter presents the code design and implementation of the channel model, equalisers and detectors that are designed and used in the research. It involves the design of Perpendicular Magnetic Recording (PMR) channel, using jitter noise model, equaliser designs and Maximum Likelihood (ML) detection using Viterbi Algorithm.

3.1 Channel Model

The channel modelling was carried out in the following steps:

3.1.1 Read Head Response/ISI

The channel transition response was first modelled, using hyperbolic tangent function from equation 2.2 defined in Chapter 2. The value of voltage is assumed to be normalised using V_{max}. This means V_{max} itself is normalised to a value of 1. The time "t" and "T_{50}" are also normalised by the bit period "B".

Isolated response involves a positive transition at time "$t + 0B$" (transition from 0 to 1) and then a negative transition at time "$t+1B$" (transition from 1 to 0). This means the 1D response will be the superposition of $s(t)$ and $s(t+1B)$. But the function represents the value of voltage at the edge of the bit, whereas in reading the medium, readings are taken at the centre of the bits. This means, to actually represent

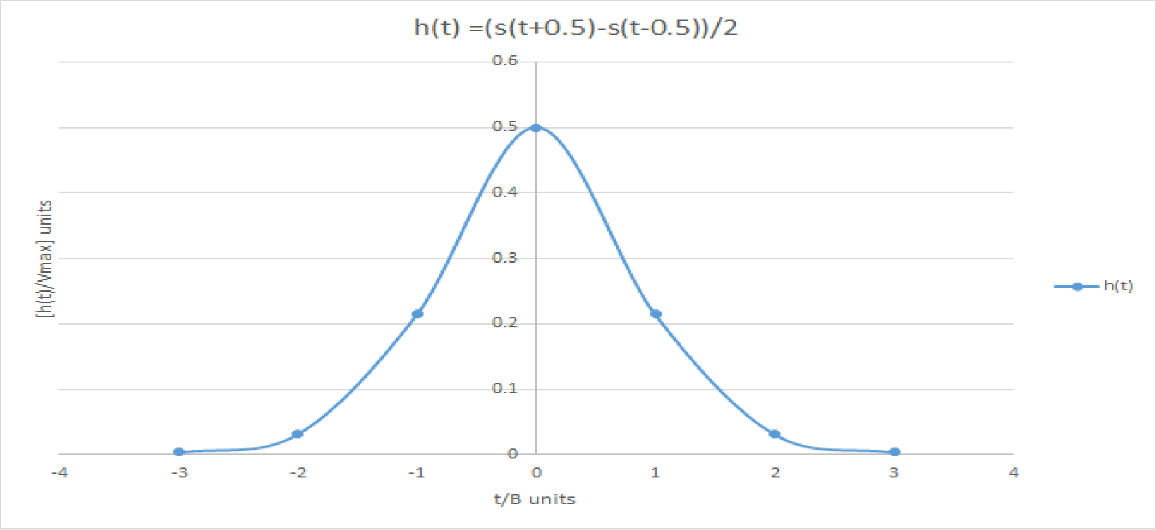


Figure 3.1: Channel Isolated Response for $V_{max} = 1$ and $T_{50} = 1$

the response of the read-head, we have to shift the function, by $B/2$, to the centre of the sampled bit. Therefore, the isolated response can be represented by equation 3.1, and the plot of the response is as shown in figure 3.1 for $V_{max} = 1$ and $T_{50} = 1$. This isolated response is the 1D equivalent of equation 2.4.

$$h(t) = (s(t + B/2) - s(t - B/2))/2 \tag{3.1}$$

If we represent transition from one bit to another as "$d($ $)$", then the transition can be expressed as shown in equation 3.2, for a binary data "x" (0 or 1). The value of "d" is one of +1, 0 or -1 for positive transition, no transition, or negative transition respectively.

$$d(t) = (x(t + 1B) - x(t))/2 \tag{3.2}$$

When more than one bit is saved on the medium, the response read at any point in time, is the sum of the responses of all the bit transitions seen from that point. Assuming the bits are saved at position $p = k_1$ to k_2, then the resulting response at position "t" is given by

$$y(t) = \sum_{p=k_1}^{k_2} d(p)s(t - pB) \tag{3.3}$$

This means the data read, is a convolution of the channel transition response and the data transitions.

When equation 3.2 is substituted in equation 3.3 and reordered to collect similar xs together, the equation becomes

$$y(t) = \sum_{p=k_1}^{k_2} x(p)(s(t - pB + 1) - s(t - pB))/2 \qquad (3.4)$$

which is equivalent to

$$y(t) = \sum_{p=k_1}^{k_2} x(p)h(t - pB) \qquad (3.5)$$

The received signal can therefore also be expressed as the convolution of the saved bits (x) with the isolated response of the channel (h). The contributions of the bits at position $pB \neq t$ is what constitutes the ISI in the medium.

3.1.2 Jitter Noise Modelling

The equations above assume a perfect transition point. But due to jitter noise, the transitions do not usually occur at the exact position of pB and or $(p+1)B$. Rather, there is random deviation of the position of transition, due to the irregularity of the grains' sizes. This random jitter can be modelled as a random white Gaussian number (a_p), of zero mean and standard deviation σ_j, added to the transition time of equation 3.3. This is shown in equation 3.6 [37].

$$y(t) = \sum_{p=k_1}^{k_2} d(p)s(t + a_p - pB) \qquad (3.6)$$

The standard deviation of the jitter noise (σ_j) is, therefore, a function of the grain granularity. It is very significant (high σ) at high density and has less effect (low σ_j) at low density. The deviations in transition time (a_p), are normally truncated to $\pm 0.5B$ whenever its amplitude is greater than 0.5B. This is to make sure the minimum of the bit energy is never below zero.

In order to make simulation of the channel simpler, Taylor expansion of equation 3.6 is taken, to give:

$$\begin{aligned} y(t) = &\sum_{p=k_1}^{k_2} d(p)s(t-pB) + \sum_{p=k_1}^{k_2} d(p)a_p s'(t-pB) \\ &+ \sum_{p=k_1}^{k_2} \frac{d(p){a_p}^2}{2!} s''(t-pB) + \sum_{p=k_1}^{k_2} \frac{d(p){a_p}^3}{3!} s'''(t-pB) + \ldots \end{aligned} \quad (3.7)$$

The successive terms become smaller as higher derivatives of $s(\ )$ are divided by factorials, and multiplied by powers of a_p. Therefore, an approximation of the expansion involves dropping the higher derivatives of the expansion. The last derivative taken determines the order of the approximation. In our model, the first-order approximation is taken. Therefore, the equation for the read data, without additive noise, can be written as

$$y(t) = \sum_{p=k_1}^{k_2} d(p)s(t-pB) + \sum_{p=k_1}^{k_2} d(p)a_p s'(t-pB) \qquad (3.8)$$

The first term of equation 3.8 is the ideal head response shown in equation 3.3, which is equivalent to, and can be substituted by equation 3.5. The second term is the first order approximation of the jitter noise. But as explained, in PMR, the read-head reads the value of the bit stored on the medium, not the transition. Therefore, the data term is rearranged by substituting in equation 3.5, to yield equation 3.9.

$$y(t) = \sum_{p=k_1}^{k_2} x(p)h(t - pB) + \sum_{p=k_1}^{k_2} d(p)a_p s'(t - pB) \qquad (3.9)$$

And from definition of $s(t)$ in equation 2.2.

$$s'(t - pB) = \frac{2}{0.579\pi T_{50}} sech^2 \left(\frac{2(t - pB)}{0.579\pi T_{50}} \right) \qquad (3.10)$$

3.1.3 2D interference/ITI

Equation 3.9 is a 1D model of the channel response along track. In a 2D situation where interference also comes from across track (ITI), the 2D response is the product of the response along the track and the response across the track as can be inferred from equation 2.4 [11] [27].

In discrete form, it is the convolution of the 1D response along track with the other 1D response across track which can be written as shown in equation 3.11.

$$z(t_1, t_2) = \sum_{p=l_1}^{l_2} \sum_{q=k_1}^{k_2} x(p, q)h_1(p)h_2(q) + \sum_{p=l_1}^{l_2} \sum_{q=k_1}^{k_2} d(p, q)a_{p,q}h_1(p)s'(q) \qquad (3.11)$$

where h_1 and h_2 are the responses along and across the track respectively. t_1 and t_2 are the sampling position across and along the track respectively. Also, "p" and "q" count the bits' positions along and across the tracks respectively.

The first term still represents the data, but now with ISI and ITI ($y(t_1, t_2)$), while the second term represents the 2D jitter noise ($n_j(t_1, t_2)$). This is equivalent to determining the ISI with jitter noise then convolving the signal across track with the ITI response.

3.1.4 Additive White Gaussian Noise

The electronic devices used in reading the HDD, introduce noises that are normally assumed to be Additive White Gaussian Noise (AWGN). These are random noises that may arise from various sources such as dirt or scratch on the medium, thermal noises of amplifiers, or resistance of the MR read head.

The standard deviation of the noise is determined and generated. It is then simply added to the read head response, be it the 2D or a 1D version of it. Equation 3.12 shows a 2D response with AWGN in a condensed representation form.

$$z(t_1, t_2) = y(t_1, t_2) + n_j(t_1, t_2) + n_w(t_1, t_2) \tag{3.12}$$

where $n_w(t_1, t_2)$ is the AWGN at sampling time t_1 and t_2.

3.1.5 Signal to Noise Ratio

The definition of Signal to Noise Error (SNR) is the ratio of signal energy or power (S) to noise energy or power (N) as shown in equation 3.13. This is normally expressed in logarithmic (decibel) form as shown in equation 3.14 [77].

$$SNR = \frac{S}{N} \tag{3.13}$$

$$SNR = 10log_{10}\left(\frac{S}{N}\right)(dB) \tag{3.14}$$

If the variance(σ^2) of the signal and that of the noise are known, then S and N can be replaced by σ^2_{signal} and σ^2_{noise} respectively. Some literature use this definition of SNR in analysing performance of magnetic channels [78].

In other literature, instead of signal energy, peak signal from the medium or saturation level of isolated pulse is used. In such situation instead of σ^2_{signal}, V^2_{max} is used [38] [57]. In the three literature cited in this paragraph, the σ^2_{noise} is considered to be from the additive white noise added to the signal before equalisation. In [68], who also used peak signal in place of signal energy, the noise energy is considered to be the sum of jitter noise and white noise energy all of which are added before equalisation. This definition, as shown in equation 3.15, is the definition adopted in this report.

$$SNR = 10log_{10}\left(\frac{V^2_{max}}{\sigma^2_j + \sigma^2_w}\right)(dB) \tag{3.15}$$

where σ_j and σ_w are the standard deviations of the jitter noise (n_j) and AWGN (n_w) respectively. V_{max} is the peak voltage of a single ideal isolated transition.

3.1.6 Implementation of the Channel Model

MATLAB and VHDL models of the channel were created for the research experimentation. T_{50}, as a ratio of bit period (B), is initially set as one of the parameters used in the initialisation function. SNR of the channel is also an input parameter for the function.

First, the initialisation function determines the channel isolated response (h_1) along tracks using equation 3.1. Between 21 to 29 terms are evaluated, one bit period apart, depending on the density (T_{50}) or the equaliser length needed. The central value in the "h_1" array is chosen to be the value with the peak amplitude in the function.

The jitter response (s') is also evaluated from equation 3.10, with the same length as the "h_1" array evaluated above. It is evaluated such that the middle term and the one after have the peak value of the response. It should be noted that the half period shift mentioned in section 3.1 is implemented in determining both the isolated and jitter responses.

V_{max} is chosen to be 1 unit. The amplitude of the jitter response is modified, such that after multiplying "d" with "a_p" and then convolving with jitter and ITI response, the noise power will be the chosen percentage of the total noise. This sets the signal such that after adding the jitter and white noise, the total noise power is equivalent to the chosen operating SNR.

Let the ratio of the jitter noise power, with respect to the sum of white and jitter noise power, be represented by "λ_j". Then we can express "λ_j" as shown in equation 3.16.

$$\lambda_j = \frac{\sigma_j^2}{\sigma_j^2 + \sigma_w^2} \tag{3.16}$$

The factor required to modify the amplitude of the jitter response is therefore given by equation 3.17. The factor is used to divide all the terms of the jitter response.

$$U = \sqrt{0.5 \sum (h_2)^2 \sum (s')^2} \tag{3.17}$$

where h_2 is the response across track (ITI response), s' is the jitter response and 0.5 is the power (variance) of transition array "d". With this modification, "$a_{p,q}$" will simply need to be an AWGN of standard deviation σ_j.

The ITI response is chosen at the beginning to be at most the interference of three tracks in this research. Therefore,
$h_2 = [\rho_1, 1, \rho_2]$
when there are three tracks interfering. The value 1 at the middle shows the read head reads the highest possible amplitude from the middle track. Whereas ρ_1 and ρ_2 are fractions of the maximum central amplitude, registered by the read-head, from the side tracks.

For two track interference, $h_2 = [\rho_1, \rho_2]$
where ρ_1 and ρ_2 represent the fraction of the total amplitude which each of the two tracks contributed. This means in this case $\rho_1 + \rho_2 = 1$.

During simulation, random binary data is generated and modulated to -1 and +1. This modulated data is assumed to be the saved data (x). The data is assumed to be written in sectors of 4kB (4096x8), consisting of 8 tracks each with 4096 bits/track. The guard band between sectors is assumed to have data of -1s all over it. This means an all zero (-1) track has to be written before each sector if there is no guarantee that it will be as expected. The last track of every sector is assumed to be at least twice the size of a normal track. This is because, in SMR system, part of the other tracks get over-written as successive tracks are written, except for the last track of each sector. A depiction of the sector is shown in figure 3.2.

Along each track, transitions are evaluated using "$x(p, q)$" and saved as "$d(p, q)$", using equation 3.2. An AWGN of variance σ_j^2 is generated and multiplied, term by term, to "$d(p, q)$". The resulting values are convolved with the jitter response (s') to generate the jitter noise for each bit position.

The saved data ($x(p, q)$) along each track is also convolved with the isolated response array (h_1) to get the ISI data of each track. This

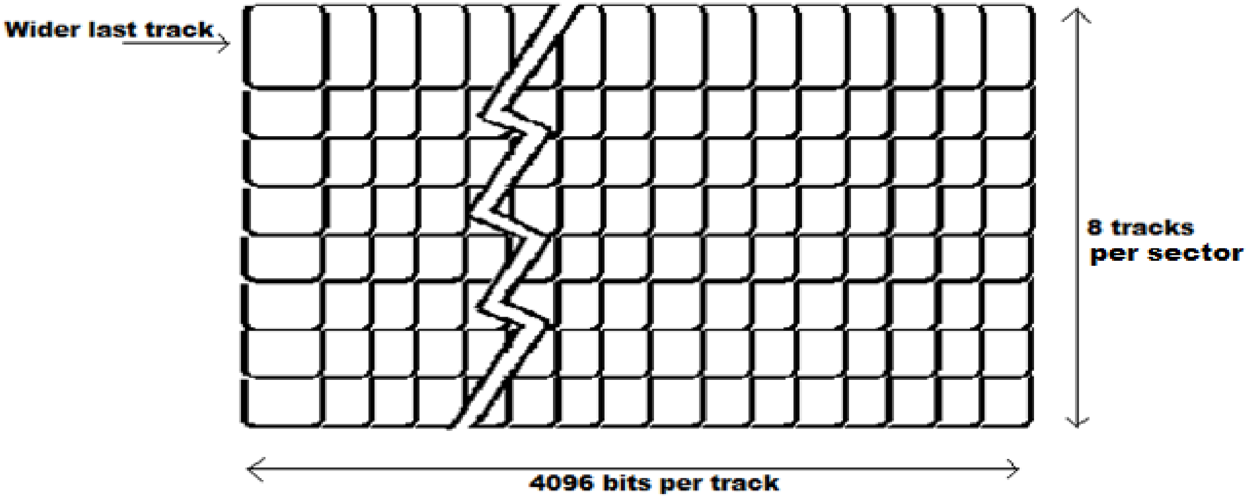


Figure 3.2: Sector dimension of the channel model

result is then added to the jitter noise determined above.

The signal which now contains ISI and 1D jitter noise is now convolved across track with the ITI response (h_2) as shown in equation 3.11. This means, for our 8 tracks, plus 1 guard track of all -1s, and an extra track width for the last track, ten tracks adjacent to each other are convolved with the ITI response. For equalisation purposes, extra random numbers are saved as tracks, from adjacent sectors, to get extra information in a situation where more tracks are needed for equalisation. More read data are sometimes needed for equalisation than for detection as in the example of section 2.5.4.

The final channel output ($Z(p,q)$)is then determined, by adding AWGN of variance σ_w^2 to the signal above, as shown in equation 3.12.

3.1.7 Example of Channel

Table 3.1 is an example of a channel with three tracks, 10 bits per track as given here. Horizontally before the sector and after the sector, there will be some bits' positions left blank or forced to zeros (-1s), as in the example we are going to use. These are the guard bands horizontally. Before and after the horizontal guards, there will be bits from preceding and succeeding tracks, represented by random numbers in our example.

Table 3.1: Sector of 3 by 10 bits in the form of -1s and 1s

1	-1	-1	1	-1	1	1	-1	-1	1
1	-1	1	1	-1	-1	-1	1	1	1
-1	1	1	1	-1	-1	1	-1	1	-1

A track before the first track will be forced to 0s, as a guard band to separate the sector from the one above it. The last track is assumed to be twice the width of the other tracks. This is due to the fact that no part of it is over written by any succeeding track. The whole of this can be illustrated in table 3.2.

Table 3.2: Sector of data in the medium

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
1	-1	-1	1	-1	-1	1	-1	-1	1	-1	1	1	-1	-1	1	-1	-1	1	1	-1	-1
1	1	-1	1	-1	-1	1	-1	1	1	-1	-1	-1	1	1	1	-1	-1	-1	1	1	-1
-1	1	-1	1	-1	-1	-1	1	1	1	-1	-1	1	-1	1	-1	-1	-1	1	-1	-1	1
-1	1	-1	1	-1	-1	-1	1	1	1	-1	-1	1	-1	1	-1	-1	-1	1	-1	-1	1

We can determine the transition of the magnetisation (Data) by taking half of the difference between the data at each position and the bit before it along a track (The bits before the first column, of data displayed, are assumed to be all -1s). The first (guard) track gives a difference of all zeros, which shows no transition if all bits are forced to -1s. The first value in second track is determined by $0.5 * (1 - (-1)) = +1$. This shows positive transition. The next (2nd) transition on the second track is determined from data in the second position and first position of the second track as given by $0.5 * (-1 - 1) = -1$. This signifies negative transition. That continues to the last bit and then repeated for other tracks. The result is shown in table 3.3 below.

Table 3.3: Transitions from preceding to present bit

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	-1	0	1	-1	0	1	-1	0	1	-1	1	0	-1	0	1	-1	0	1	0	-1	0
1	0	-1	1	-1	0	1	-1	1	0	-1	0	0	1	0	0	-1	0	0	1	0	-1
0	1	-1	1	-1	0	0	1	0	0	-1	0	1	-1	1	-1	0	0	1	-1	0	1
0	1	-1	1	-1	0	0	1	0	0	-1	0	1	-1	1	-1	0	0	1	-1	0	1

White Gaussian Noise (WGN) is multiplied by the transition values to get the random jitters. The SNR of the WGN is determined from a percentage of the total SNR. For example for a total $SNR = 15$dB, with jitter contributing 80% and AWGN contributing 20%, the jitter and AWGN SNR are given by
$SNR_j = SNR - 10log_{10}(0.8) = 15.97dB$ and
$SNR_w = SNR - 10log_{10}(0.2) = 21.99dB$ respectively.

Table 3.4: Random Jitter

0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.15	0.15	0.00	0.04	-0.04	0.00	-0.07	...	0.07	0.02	0.00	-0.03	0.00	0.13	0.00
-0.09	0.00	0.28	0.01	0.06	0.00	-0.12	...	0.00	-0.05	0.00	0.00	0.16	0.00	-0.06
0.00	-0.01	-0.06	0.04	0.15	0.00	0.00	...	-0.13	0.00	0.00	0.07	-0.02	0.00	-0.02
0.00	-0.04	-0.01	-0.11	0.01	0.00	0.00	...	0.08	0.00	0.00	0.11	0.15	0.00	0.17

Table 3.4 shows some values used in this example. The WGN is generated using MATLAB function "*awgn( )*". Table 3.2 and table 3.4 represent the data written on the medium and the jitter at the transitions of the bits respectively.

For values of $T_{50} \simeq 1$, equation 3.1 can be used to find the following approximated 5 tap Isolated channel response and jitter response.
$h = [0.03,\ 0.22,\ 0.50,\ 0.22,\ 0.03]$ and
$J = [0.00,\ 0.20,\ 0.98,\ 0.98,\ 0.20]$.

It can be observed that the total sum of the channel response is 1. This means the peak amplitude achievable is 1 or -1. The jitter response is normalised such that the sum of squares of the coefficient sums up to 2. This is to counter the fact that, the RMS of the transitions averages to 0.5, for a large data. Therefore, when the jitter response is convolved with the transitions, the power will be normalised to 2*0.5(=1), multiplied by the jitter noise amplitude. The jitter noise amplitude is determined from SNR as explained above.

In this example, we are going to assume the read-head spans only parts of two track with an equal amount of field read from both tracks. This means the ITI can be represented as $ITI = [0.5,\ 0.5]$, or $[1,\ 1]$, for simplicity. First, a convolution of head response (h) is taken with the data in table 3.2.

Therefore we can determine the component of the first track for the first read signal, before jitter is added, using 1st to 5th bit of the track, as:

$Y_{1,1} = 0.03 * (-1) + 0.22 * (-1) + 0.50 * (-1) + 0.22 * (-1) 0.03 * (-1)$

$Y_{1,1} = -1.00.$

The resulting jitter for that range of data can be determined, using convolution of the jitter noise with jitter head response ("J"). Therefore, the first term is calculated as:

$J_{1,1} = 0.00 * (0) + 0.2 * (0) + 0.98 * (0) + 0.98 * (0) + 0.2 * (0) = 0.00$

Therefore, total contribution of read signal from first track in the position of first signal is

$X_{1,1} = Y_{1,1} + J_{1,1} = -1 + 0.0 = -1.00$

The first (guard) track has all its read signal as -1 and jitter as zero because all its saved data is -1s. This means all its transitions are 0s. Therefore, we are going to the 2nd track (first data track) and find its

first read signal in the same way as above.
$Y_{2,1} = 0.03 * (+1) + 0.22 * (-1) + 0.50 * (-1) + 0.22 * (+1) 0.03 * (-1)$
$Y_{2,1} = -0.50.$ and
$J_{2,1} = 0.00{*}(0.15){+}0.2{*}(0.15){+}0.98{*}(0.00){+}0.98{*}(0.04){+}0.2{*}(-0.04)$
$J_{2,1} = 0.0612$

$X_{2,1} = Y_{2,1} + J_{2,1} = -0.50 + 0.0612 = -0.4388.$
The second signal on the second track can similarly be determined using 2nd to 6th bit of the track as shown below.
$Y_{2,2} = 0.03 * (-1) + 0.22 * (-1) + 0.50 * (+1) + 0.22 * (-1) 0.03 * (-1)$
$Y_{2,2} = 0.00.$ and
$J_{2,2} = 0.00{*}(0.15){+}0.2{*}(0.00){+}0.98{*}(0.04){+}0.98{*}(-0.04){+}0.2{*}(0.00)$
$J_{2,2} = 0.000$

$X_{2,1} = Y_{2,1} + J_{2,1} = 00.00 + 0.00 = 0.000.$
The next uses bit position 3rd to 7th and so on until the end of the track is reached. Similar procedure is repeated for other tracks, to determine their components of read signal. Table 3.5 shows the calculated values for all the signals.

Table 3.5: Track Components of Read Signal

-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	...	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
-0.44	0.00	-0.55	-0.59	-0.11	-0.54	-0.40	...	-0.37	-0.42	0.09	-0.47	-0.47	0.44	0.56
0.24	0.18	-0.46	-0.59	0.05	0.12	0.66	...	0.60	0.89	0.45	-0.55	-0.86	-0.34	0.58
-0.11	0.23	-0.41	-0.88	-0.64	0.36	0.89	...	-0.32	-0.07	-0.68	-0.89	-0.50	0.05	-0.51
-0.24	-0.04	-0.57	-0.85	-0.36	0.64	0.90	...	-0.21	-0.02	-0.52	-0.84	-0.42	0.25	-0.30

To find the final read signal, we need to combine interfered (neighbouring) tracks in the appropriate proportion and then add AWGN. Given an $ITI = [1,\ 1]$, the addition of data in two neighbouring tracks is carried out, which will modify the amplitude of the jitter noise added earlier. The factor by which it is modified is:
$\sqrt{ITI(1)^2 + ITI(2)^2} = \sqrt{(1^2 + 1^2)} = 1.4142.$

To restore the ratio of the jitter noise to the AWGN, we have to multiply the amplitude of the AWGN, at the given power (21.99dB), by the modifying factor. Therefore the first final read signal can be determined as:
$Z_{1,1} = X_{1,1} * ITI(1) + X_{2,1} * ITI(2) + \sqrt{2} * AWGN$
$Z_{1,1} = (-1 * 1) + (-0.44 * 1) + \sqrt{2} * -0.076 = -1.547.$

The second data in the first row of read signal is determined as:
$Z_{1,2} = X_{1,2} * ITI(1) + X_{2,2} * ITI(2) + \sqrt{2} * AWGN$
$Z_{1,2} = (-1 * 1) + (-0.00 * 1) + \sqrt{2} * -0.020 = -1.028.$

This continues to the end of the track. The second track can be determined in similar way, using data X from second track and third track as shown:
$Z_{2,1} = X_{2,1} * ITI(1) + X_{3,1} * ITI(2) + \sqrt{2} * AWGN$
$Z_{2,1} = (-0.44 * 1) + (0.24 * 1) + \sqrt{2} * -0.015 = -0.221.$

The complete read-data is shown in table 3.6 below. This example read-data is going to be used in future examples for different equalisers/detectors.

Table 3.6: Read Signal

-1.55	-1.03	-1.54	-1.46	-1.16	-1.55	-1.25	...	-1.42	-1.44	-0.91	-1.46	-1.34	-0.65	-0.40
-0.22	0.03	-0.91	-1.30	0.06	-0.62	0.30	...	0.27	0.28	0.58	-1.04	-1.35	0.12	1.09
0.09	0.59	-0.82	-1.47	-0.61	0.47	1.54	...	0.25	0.92	-0.14	-1.44	-1.36	-0.22	0.05
-0.34	0.07	-1.05	-1.71	-0.93	0.98	1.82	...	-0.51	-0.13	-1.20	-1.83	-0.93	0.39	-0.80

3.2 Linear Equalisers

The linear equalisers made during the course of this research are ZF, MMSE and PR equalisers.

3.2.1 Zero Forcing Equaliser

Zero Forcing (ZF) equaliser is used in this research for testing the channel validity at the beginning of the research work, and later for cancelling either ITI or ISI to reduce the detection problem of the channel into 1D problem.

By assuming the ITI and ISI are linear convolutions of the respective responses to the saved signal, 1D equaliser based on the ITI or ISI response can, therefore, be used across or along track to cancel the ITI or ISI respectively without distorting the un-cancelled interference. Rather than using 2D ZF equaliser, two 1D equalisers are therefore used with each putting into consideration only one direction of the interference.

If say, the received data can be expressed simply in terms of h_1 response across track, h_2 response along track, "x" saved data, n_j jitter noise and n_w white noise as shown in equation 3.18

$$y = h_1 * h_2 * x + n_j + n_w \tag{3.18}$$

Then a linear equaliser ($H_{h_1}^{-1}$) can be determined for cancelling ITI such that:

$$w_1 = H_{h_1}^{-1} y = h_2 * x + H_{h_1}^{-1}(n_j + n_w) \tag{3.19}$$

or in order to cancel ISI first, equaliser ($H_{h_2}^{-1}$) can be determined and used as shown in equation 3.20.

$$w_2 = H_{h_2}^{-1} y = h_1 * x + H_{h_2}^{-1}(n_j + n_w) \tag{3.20}$$

This implies any of the interferences can be cancelled first before the other, rather than using a 2D ZF equaliser. The equaliser terms ($H_{h_1}^{-1}$) ($H_{h_2}^{-1}$) means the inverse of a H matrix formed using the response h_1 and h_2 respectively. It should also be noted that the two equalisations are applied in two different directions.

Using equation 2.17 in forming the H matrix involves truncation, as a form of approximation. If for each data, its significant energy spreads k positions before and after its position, then among the values that will be determined from the ZF equalisation, only the central value utilises the energy from all the significant signals around it. The symbol in the first position, for example, utilises only the symbols after it. This makes the central symbol the most reliable. Therefore, the (central) row of the equalisation matrix responsible for the central symbol is the only row we pick as the equaliser coefficients. A convolution of that row and the received signal, therefore, produces the required results.

In the case of response across tracks, as mentioned in section 3.1.6, we assumed 8 tracks with a track of all -1s before each sector and an extra track at the end, the same as the last track, to representing wider track. Therefore, for a three track interference ($[\rho_1,\ 1,\ \rho_2]$), the H matrix can be represented as shown in equation 3.21.

$$H_{h_1} = \begin{bmatrix} \rho_1 & 1 & \rho_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \rho_1 & 1 & \rho_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \rho_1 & 1 & \rho_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \rho_1 & 1 & \rho_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \rho_1 & 1 & \rho_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \rho_1 & 1 & \rho_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \rho_1 & 1 & \rho_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \rho_1 & 1+\rho_2 \end{bmatrix} \tag{3.21}$$

where, the first column represent interference from the guard band between sectors of all -1. The last term $(1 + \rho_2)$ shows an interference from the track itself for the last wider track of each sector. With this arrangement, there is no need for truncation and therefore all the symbols are evaluated from the inverse of the H matrix without having to pick the central row and then convolving with the read data.

Therefore if E_1 is the equaliser derived from $(H_{h_1}^{-1})$ and E_2 is the equalising row of $(H_{h_2}^{-1})$, then the approximate ZF detected signal can be expressed as:

$$\hat{x} = E_1 E_2 y = x + E_1 E_2 (n_j + n_w) \tag{3.22}$$

3.2.2 Minimum Mean Square Error Equaliser

The MMSE equaliser is treated the same way as the ZF equaliser except for a modification in the equaliser matrix. The equaliser (E) is picked from the MMSE equalisation matrix given by

$$H_{mmse}^{-1} = (H'H + \sigma^2 I)^{-1} H' \tag{3.23}$$

where σ^2 is the variance of the noise on the channel.

This treatment allows some ITI and or ISI in the signal but reduces the noise amplification in ZF. This produces a combined remnant noise and interference, which gives a better result than eliminating the interference totally. Equation 3.23 was shown to reduce the Mean Square Error between the received and the equalised signal with a white noise of variance σ^2 to the minimum. In our implementation, jitter noise is assumed to be a white noise.

The MMSE equaliser can be reduced to ZF equaliser by assigning the value of 0 to σ^2.

3.2.3 Example of ZF and MMSE equaliser

In the example of channel modelling given earlier, read data was determined and presented in table 3.6, for a channel that has isolated response of $h = [0.03,\ 0.22,\ 0.50,\ 0.22,\ 0.03]$, $ITI = [1,\ 1]$, from a sector that has 3 tracks, 10 bit per track. The guard band of the sector is forced to all -1s and last track is at least twice the width of other tracks.

To find the equaliser needed to reduce the interferences to zero (ZF equaliser), we first form the H matrix from "h" as given below. Equaliser of 5 taps is going to be used in this example.

$$H_1 = \begin{bmatrix} 0.50 & 0.22 & 0.03 & 0.00 & 0.00 \\ 0.22 & 0.50 & 0.22 & 0.03 & 0.00 \\ 0.03 & 0.22 & 0.50 & 0.22 & 0.03 \\ 0.00 & 0.03 & 0.22 & 0.50 & 0.22 \\ 0.00 & 0.00 & 0.03 & 0.22 & 0.50 \end{bmatrix}$$

Inverse of the H_1 matrix is given by:

$$H_1^{-1} = \begin{bmatrix} 2.558 & -1.337 & 0.502 & -0.158 & 0.039 \\ -1.337 & 3.256 & -1.597 & 0.577 & -0.158 \\ 0.502 & -1.597 & 3.345 & -1.597 & 0.502 \\ -0.158 & 0.577 & -1.597 & 3.256 & -1.337 \\ 0.039 & -0.158 & 0.502 & -1.337 & 2.558 \end{bmatrix}$$

Therefore, the best ZF equaliser required to cancel ISI in the given example is the middle row, which determines the middle term of the interfered signals. Equaliser is, therefore,
$E_1 = [0.502,\ -1.597,\ 3.345,\ -1.597,\ 0.502]$

Convolution of the equaliser terms and the read signal in table 3.6 can be done in the following manner.

The first term of the first track is determined from 1st to 5th term of the first track as given below.

$$R_{1,1} = (0.502 * -1.55) + (-1.597 * -1.03) + (3.345 * -1.54) +$$

$(-1.597 \ast -1.46) + (0.502 \ast -1.16)$.
$R_{1,1} = -2.535$

The second term in the first row is found using the 2nd read signal to the 6th as shown.

$R_{1,2} = (0.502 \ast -1.03) + (-1.597 \ast -1.54) + (3.345 \ast -1.46) + (-1.597 \ast -1.16) + (0.502 \ast -1.55)$.
$R_{1,2} = -1.883$.

This continues to the end of the track. It is also repeated for all other tracks. To reduce computations in this situation, we will add the last read track to the third one before the convolution, to improve the SNR of the last bit which is the only data signal in the last track. Table 3.7 shows the equalised data for all the sector.

Table 3.7: Equalised Signal

-2.54	-1.87	-0.47	-2.55	-1.16	-0.86	-1.58	-0.10	-0.19	-2.19	-2.11	0.20	-2.34	-1.77
-1.10	-3.29	2.96	-3.05	1.01	1.45	-1.64	0.20	0.20	0.47	-0.89	2.61	-2.05	-2.21
-3.13	-4.13	-1.64	1.25	4.23	3.37	-4.16	-3.86	2.01	-3.18	3.25	-1.80	-4.66	-3.76

To create the ITI canceller we need to form its H matrix too. The H matrix will initially be as shown in the matrix below, where the first column is the contribution of the first(guard) track, the second column represents contributions from the second (first data) track and so on, while the last column is from the extra-wider last track as explained from equation 3.21. The rows represent combination of data interference for each track of data read in table 3.6.

$$H_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Since we have added the last two read data tracks before equalisation along track, it means we have to add the last two rows of the matrix "H_2". Also, because the last two columns contain the same data of the last track, we can add the last two column for the final "H_2" matrix. The first column represents the guard which we already know are all -1s. This means we can just remove -1s from the data of first track to remove the interference manually and ignore the first column in our matrix. These modifications are particular to this scenario and therefore a different situation must be looked according to its nature. The resultant "H_2" matrix is therefore given by

$$H_2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 3 \end{bmatrix}$$

And its inverse is:

$$H_2^{-1} = \begin{bmatrix} 1.000 & 0.000 & 0.000 \\ -1.000 & 1.000 & 0.000 \\ 0.333 & -0.333 & 0.333 \end{bmatrix}$$

This matrix is therefore multiplied by a column matrix formed from adjacent bits from all the tracks (after removing the -1s from first track). The first column is given by

$$C_1 = \begin{bmatrix} 1.000 & 0.000 & 0.000 \\ -1.000 & 1.000 & 0.000 \\ 0.333 & -0.333 & 0.333 \end{bmatrix} * \begin{bmatrix} -1.54 \\ -1.10 \\ -3.13 \end{bmatrix} = \begin{bmatrix} -1.54 \\ 0.44 \\ -1.19 \end{bmatrix}$$

After all columns of data are processed in the same way, the final result will be as shown below.

Table 3.8: Continuous Detected Signal

-1.54	-0.87	0.53	-1.55	-0.16	0.14	-0.58	0.90	0.81	-1.19	-1.11	1.20	-1.34	-0.77
0.44	-2.42	2.43	-1.50	1.17	1.31	-1.06	-0.70	-0.61	1.66	0.22	1.41	-0.71	-1.44
-1.19	-0.57	-1.36	0.92	1.02	0.69	-1.03	-1.05	0.87	-1.61	1.01	-1.07	-1.32	-0.77

After this process, a hard decision can be made where everywhere the data is less than or equal to 0, it is replaced by -1 else it is replaced by +1, as shown below.

Table 3.9: Discrete Detected Signal

-1	-1	+1	-1	-1	+1	-1	+1	+1	-1	-1	+1	-1	-1
+1	-1	+1	-1	+1	+1	-1	-1	-1	+1	+1	+1	-1	-1
-1	-1	-1	+1	+1	+1	-1	-1	+1	-1	+1	-1	-1	-1

The first two and last two columns of table 3.9 are the guard bits, while the data starts from the third column to the 12th. It can be seen that when compared to the original data in table 3.1 or table 3.2 there is only one bit error. The first bit in the second data track (on first guard bit) is in error.

MMSE equaliser fundamentally follows the same pattern except for a modification in "H_1" matrix. For the example given here, with 15dB SNR, the standard deviation of the amplitude of noise will be $\sigma^2 = 10^{-SNR/10} = 0.0316$. This is multiplied by an identity matrix and added to the "H_1" matrix of the ZF case as shown below. Note that because $H_1 = H_1'$ then $H_1' * H_1 = H_1^2$.

$$H_1'H_1 + \sigma^2 I = \begin{bmatrix} 0.50 & 0.22 & 0.03 & 0.00 & 0.00 \\ 0.22 & 0.50 & 0.22 & 0.03 & 0.00 \\ 0.03 & 0.22 & 0.50 & 0.22 & 0.03 \\ 0.00 & 0.03 & 0.22 & 0.50 & 0.22 \\ 0.00 & 0.00 & 0.03 & 0.22 & 0.50 \end{bmatrix}^2 + 0.0316 \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_1'H_1 + \sigma^2 I = \begin{bmatrix} 0.3309 & 0.2266 & 0.0784 & 0.0132 & 0.0009 \\ 0.2266 & 0.3793 & 0.2332 & 0.0784 & 0.0132 \\ 0.0784 & 0.2332 & 0.3802 & 0.2332 & 0.0784 \\ 0.0132 & 0.0784 & 0.2332 & 0.3793 & 0.2266 \\ 0.0009 & 0.0132 & 0.0784 & 0.2266 & 0.3309 \end{bmatrix}$$

$$H^{-1} = (H_1'H_1 + \sigma^2 I)^{-1} H_1' = \begin{bmatrix} 1.8951 & -0.5637 & -0.0032 & 0.0706 & -0.0303 \\ -0.5637 & 1.9701 & -0.5024 & -0.0418 & 0.0706 \\ -0.0032 & -0.5024 & 1.9327 & -0.5024 & -0.0032 \\ 0.0706 & -0.0418 & -0.5024 & 1.9701 & -0.5637 \\ -0.0303 & 0.0706 & -0.0032 & -0.5637 & 1.8951 \end{bmatrix}$$

After this stage, everything goes as in the ZF example.

3.2.4 Partial Response Equaliser

In trying to achieve their aim of detecting the saved data linearly, ZF and MMSE amplify the AWGN resulting to sub-optimal result. Therefore, instead of trying to cancel the whole interference using a linear equaliser, PR equalisers just cancels a portion of the interference, and leave a controlled amount of the interference. This can later be removed more effectively using maximum likelihood (ML) detectors.

The target is usually given by the coefficients representing the controlled interference. As an example for a target of length 3, can be given by $[g_1, g_2, g_3]$. A column vector (G) will be formed, of length equal to equaliser length, as shown in equations 3.24.

$$G = [0, ..., 0, g1, g2, g3, 0, ..., 0]^T \tag{3.24}$$

From this, the equaliser coefficients can be determined from the equation given below.

$$E_{PR} = H^{-1}G \tag{3.25}$$

Convolution of E_{PR} with the data produces a signal where the interference is as defined by the target coefficients. The coefficients are also chosen such that the equalised data will have minimum mean square error between received and equalised data [45].

As an example, assuming the data in table 3.6 is to be shaped to a target of [0.6, 1.0, 0.6], using an equaliser of length 7, following the procedure in section 3.2.3 above for determining equaliser, we can form H matrix of dimension 7x7 and find its inverse. The inverse is then multiplied by a column vector (G) formed from the target, as shown in the expression below.

$$H_1 = \begin{bmatrix} 0.50 & 0.22 & 0.03 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.22 & 0.50 & 0.22 & 0.03 & 0.00 & 0.00 & 0.00 \\ 0.03 & 0.22 & 0.50 & 0.22 & 0.03 & 0.00 & 0.00 \\ 0.00 & 0.03 & 0.22 & 0.50 & 0.22 & 0.03 & 0.00 \\ 0.00 & 0.00 & 0.03 & 0.22 & 0.50 & 0.22 & 0.03 \\ 0.00 & 0.00 & 0.00 & 0.03 & 0.22 & 0.50 & 0.22 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.03 & 0.22 & 0.50 \end{bmatrix}^{-1} * \begin{bmatrix} 0.0 \\ 0.0 \\ 0.6 \\ 1.0 \\ 0.6 \\ 0.0 \\ 0.0 \end{bmatrix} = \begin{bmatrix} 0.1685 \\ -0.4838 \\ 0.7390 \\ 1.4077 \\ 0.7390 \\ -0.4838 \\ 0.1685 \end{bmatrix}$$

The equaliser is therefore
$E = [0.1685, \ -0.4838, \ 0.7390, \ 1.4077, \ 0.7390, \ -0.4838, \ 0.1685]$.

A convolution along tracks, of the coefficients of this shaping equaliser with the read data in table 3.6 will give the data in table 3.10 below:

3.3 ML detectors

After the signal is shaped to the chosen target, the information component of the signal becomes equivalent to the convolution of the target

Table 3.10: Shaped Signal

-3.27	-2.84	-3.25	-2.81	-2.43	-2.09	-1.06	-1.25	-3.33	-3.03	-2.30	-3.01
-2.16	-0.96	-0.43	0.22	0.97	-0.77	-1.18	0.07	0.16	1.28	0.78	-2.00
-3.36	-1.67	1.26	3.56	2.18	-1.66	-2.88	-1.26	0.93	1.88	-0.16	-3.42
-4.13	-2.05	2.09	4.38	1.98	-1.89	-1.79	-0.60	-0.17	-0.90	-2.37	-4.50

response to the data. The signal at every point in time, therefore, depends on the "k" signals before it, where "k" is termed the constraint length of the detector.

The value of the response, for an equalised magnetic channel signal, is the sum of products (SOP) of the target response and the saved data. This is called the dibit response, or the reference symbols, in some part of this report. For example, the dibit response of a target [0.6, 1.0, 0.6] for all possible bit combination is given in table 3.11.

Table 3.11: Dibit Response for target [0.6 1.0 0.6]

bits	SOP with target	Dibit Response
0 0 0	-1*0.6 -1*1.0 -1*0.6	-2.2
1 0 0	+1*0.6 -1*1.0 -1*0.6	-1.0
0 1 0	-1*0.6 +1*1.0 -1*0.6	-0.2
1 1 0	+1*0.6 +1*1.0 -1*0.6	+1.0
0 0 1	-1*0.6 -1*1.0 +1*0.6	-1.0
1 0 1	+1*0.6 -1*1.0 +1*0.6	+0.2
0 1 1	-1*0.6 +1*1.0 +1*0.6	+1.0
1 1 1	+1*0.6 +1*1.0 +1*0.6	+2.2

How the response changes from one value to another depends on the previous state in which the response was, and the present bit. For the target in the example above, the state is determined by the last two bits in the dibit, whereas the earliest bit determines the state it transits to. This transition information can be represented in a trellis as shown in figure 3.3.

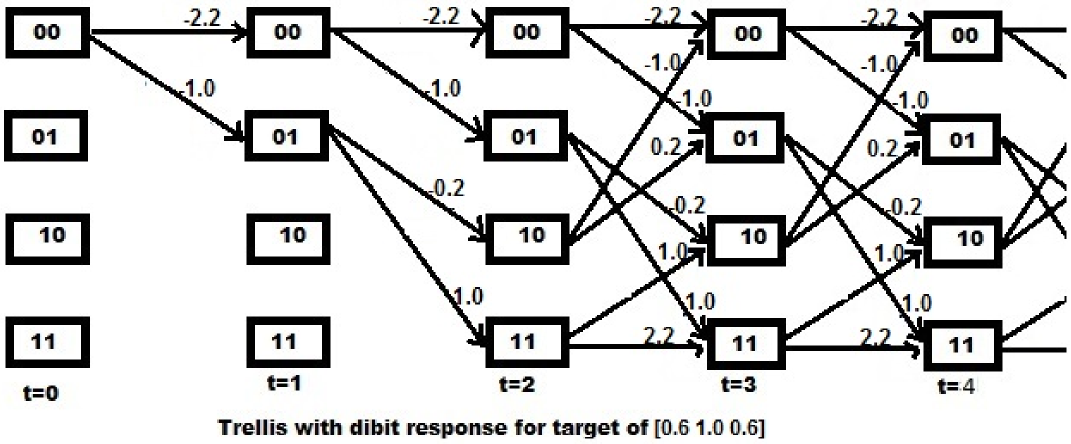


Figure 3.3: Trellis for target [0.6 1.0 0.6]

Figure 3.3 is an example of 1D target of constraint length (k=) 2. The trellis has an input of 1 bit (n) at a time. This is why each state only has only 2 (2^n) possible transitions going out of it (i.e 0 or 1), and two possible transitions going into it. The number of states is given by 2^k that is 4 in our example.

An example of 2D trellis for a two-track interference ($n = 2$ bits), with a target of constraint length $k = 2$ along track, is shown in figure 3.4.

ML detectors such as Viterbi detector and BCJR are based around these trellises.

3.3.1 Maximum Likelihood Sequence Detector

Viterbi detector is used in this section as an MLSD detector for PRML. After the data is equalised to a target, the expected data will ideally be the dibit response, as in the example shown in table 3.11. Therefore, the distance metric (DM) is determined from the difference of these possible expected dibit responses and the received signal.

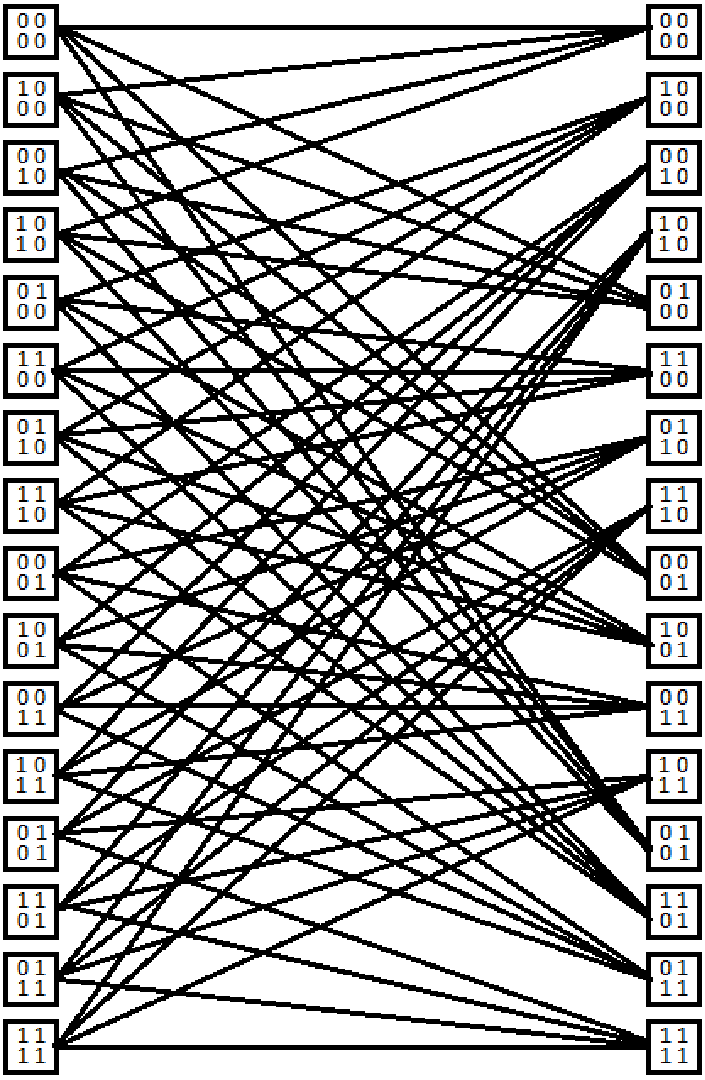


Figure 3.4: 2D Trellis for k=2, n=2

The signal passes through the Viterbi Algorithm (VA) as explained in section 2.5.1. For hard decision VA, the history kept is a simple 0 or 1 in a 1D problem. In a situation where the soft information is needed for another use, the Branch Metric (BM) of both 0 and 1 are saved in the history of the VA instead.

For a 2D scenario, the history kept is a number between 0 to $2^n - 1$ representing the index of the selected transition. Where a soft information is needed as output, the BMs of all the transitions are kept. The minimum BM is chosen to be the next State Metric (SM) and is kept in the history as the selected transition to be used for next SM.

The 1D VA is used when linear equaliser is used to cancel one of the interferences as will be discussed below.

ML Along Track

The signal is first equalised using 1D PR equaliser along track to shape the signal to selected target. 1D ZF equaliser is then used to cancel ITI across track. VA as an ML detection is then applied to the resulting signal to finally detect the data. This technique will be termed as ML Along-Track henceforth

Register exchange mode of populating history is used in the example to be presented. That is a situation in which the history of a state, which leads to the best selected BM, is carried along to the destination state, before adding the latest history. This makes trace-back just a matter of picking up the data of the best state. The best state is determined every cycle because for the 1D cases studied. The complexity is small such that the search for best state can be implemented every cycle without heavy toll on complexity and speed.

Using the example given in table 3.10, which is already shaped along track by shaping equaliser of target [0.6, 1.0, 0.6]. We now need to cancel ITI in the same way as done for ZF equaliser example. The resultant data will be as shown in table 3.12:

Table 3.12: Shaped Signal

-2.27	-1.84	-2.25	-1.81	-1.43	-1.09	-0.06	-0.25	-2.33	-2.03	-1.30	-2.01	-2.2	-2.2
0.11	0.88	1.82	2.03	2.40	0.32	-1.12	0.32	2.49	3.31	2.08	0.01	-2.2	-2.2
-2.53	-1.53	0.51	1.97	0.59	-1.29	-1.18	-0.73	-0.58	-0.78	-1.54	-2.64	-2.2	-2.2

The last two columns are just added as a padding for the purpose of traceback.

To carry out Viterbi ML detection along track, we start initialising the SM and history of each state. If the state is definitely known to start from 00 state, the first term of the first track will be initialised with the smallest metric (0) for state 00. The other states are assigned equal probabilities that sum up to a total of 1. From then on, the procedure continues as shown in the example of figure 3.5. The traceback length (TBL) is assumed to be 4 symbols (bits) and target length 3 (0.6, 1.0, 0.6).

From the figure, it could be seen that the state with the minimum SM is the second state. This means the output will be taken from that state in a register exchange implementation. This output is the earliest bit (that is the first shown by an arrow). In this case, the output is a “1”. The reference symbols, shown in table 3.11 and figure 3.3, are used in the squared term to find DMs. These DMs are added to the SMs to get BMs. For every state, the minimum SM coming into that state is selected as the next state SM, and the history (except for the first bit) is appended to either 0 or 1 if the top or lower BM is selected respectively. Information in the next state is now used as that of present state for processing of the next symbol along the same track.

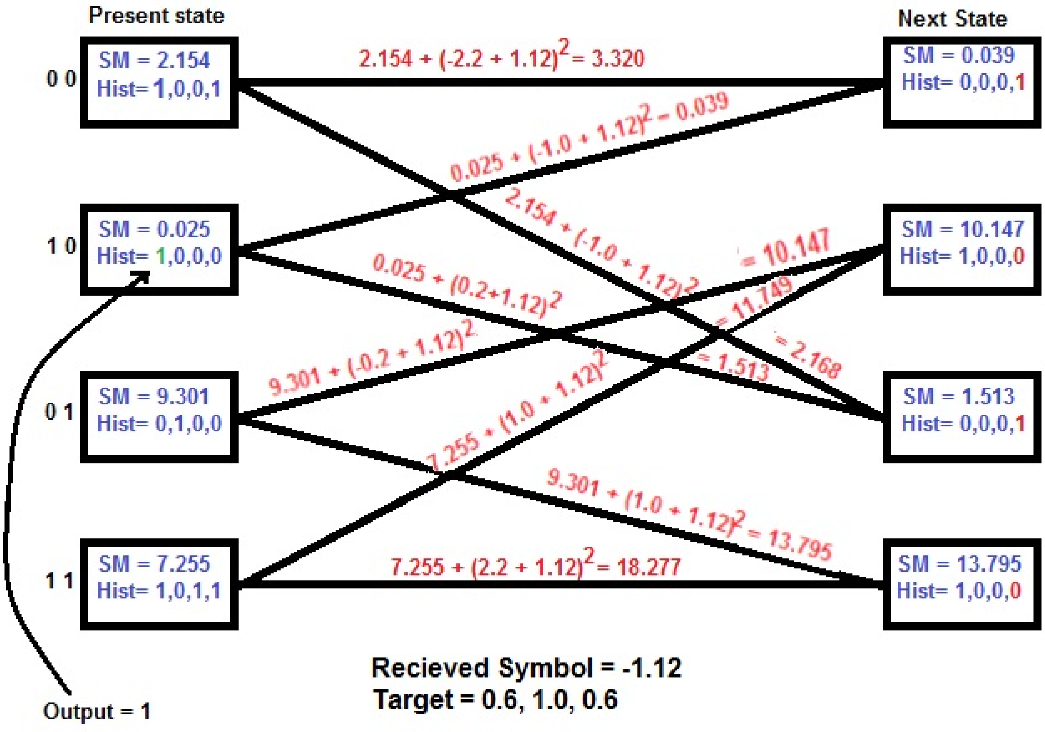


Figure 3.5: Example of 1D Viterbi Detection

This continues until the end of the tracks.

ML Across Track

In this method of detection, 1D ZF equaliser is used to cancel ISI along track. The ITI response is assumed to be the target across track and therefore does not need equalisation. This is because we assumed that the interference across track does not involve more than three tracks. Then VA ML detector is used across the tracks to detect the signal. This is from now on termed ML Across-Track.

Register exchange is also used in saving the history but in reading the data, a whole band of 8 bits are processed across track before the data is read. This means the best path will be searched for once after every band of adjacent bits across tracks are processed.

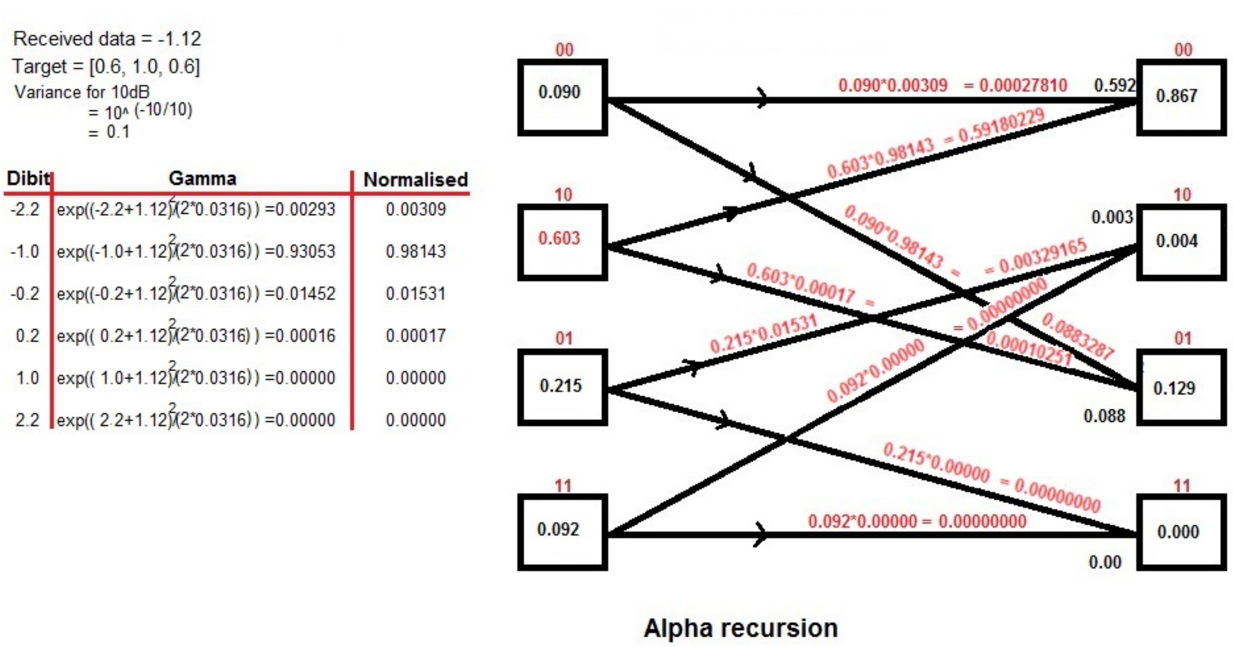
TBL is, therefore, the number of tracks. The first symbol of first track, the first symbol of second track, the first symbol of third track, and so on, are treated as successive symbols. A similar thing is done to second symbols, and so on to the end of the data. The difference to the example shown in figure 3.5 is that, no need to find the minimum SM to pick an output until all the adjacent symbols from all tracks are processed.

3.3.2 Maximum A-posteriori Probability Detector

The MAP detector used in our research is the BCJR algorithm. Normalised probabilities are used for forward recursive probability (α), backward recursive probability (β) and transition probability (γ) evaluated from equation 2.34, 2.35 and 2.36 respectively.

The α of all the states, for the first symbol, are initialised according to the known starting condition. In our situation, the known starting state is the first state, which will be given α of 1. The rest have equal value of α which is 0. The β's of the last symbol in a track are also initialised according to the knowledge of ending state. Figure 3.6 shows an example of alpha recursion for a PRML based on BCJR. The target is [0.6, 1.0, 0.6] at an SNR of 10 dB. The received signal at the instance is assumed to be -1.12

The table by the left of figure 3.6 shows the determination of γ. Normalisation can be done for γ by dividing by the total of the γ's as done in the example. This can be ignored to reduce computations. We assume the α of the states, at the instance of receiving the signal are: 0.090, 0.603, 0.215 and 0.092. Each branch is associated with a γ, corresponding to the dibit response of that branch. The γ is multiplied by the α of the state from which the branch emanates to get the branch probability, as shown in the figure. All branch probabilities that meet at the same state are then added to give the α of the new state for the



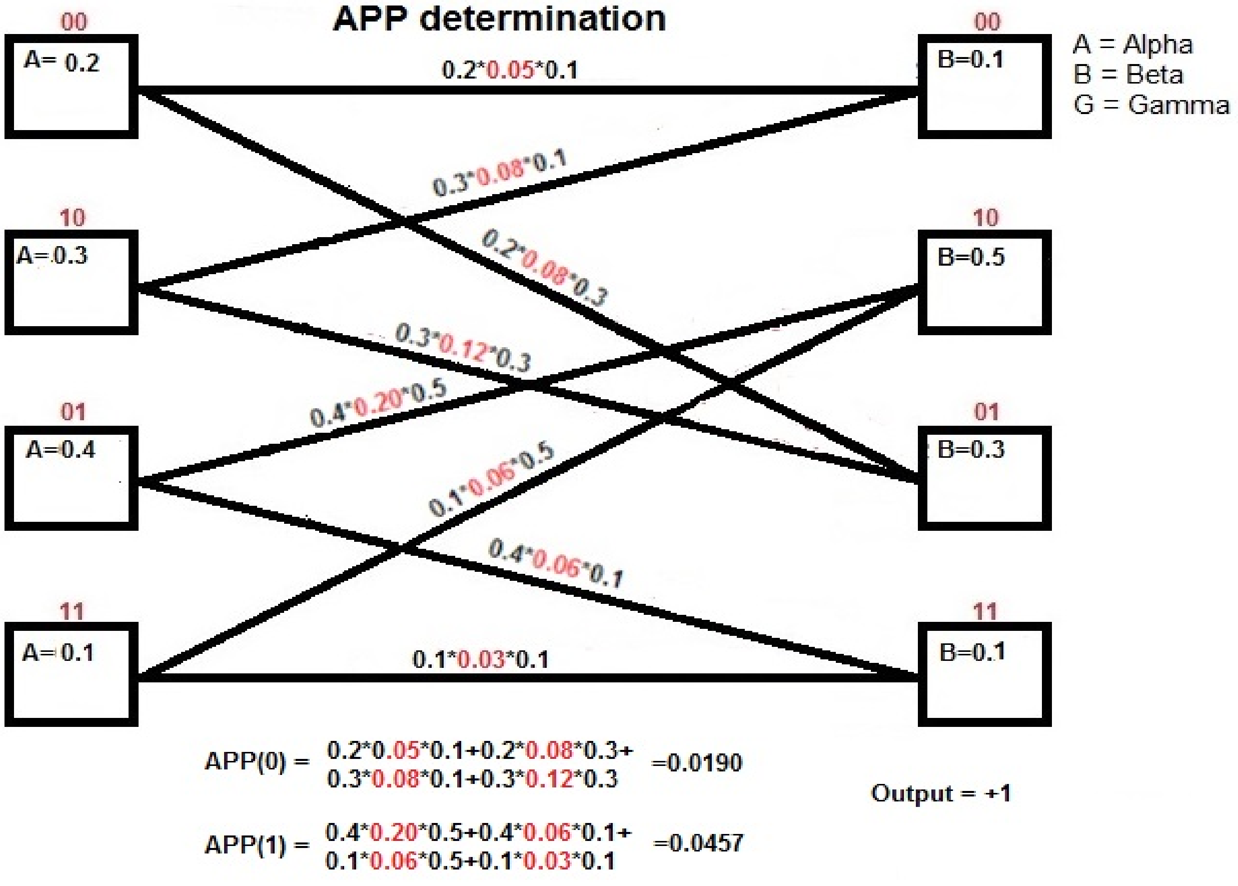
Received data = -1.12

Target = [0.6, 1.0, 0.6]

Variance for 10dB
= 10^ (-10/10)
= 0.1

Dibit	Gamma	Normalised
-2.2	$\exp((-2.2+1.12)^2/(2*0.0316))$ =0.00293	0.00309
-1.0	$\exp((-1.0+1.12)^2/(2*0.0316))$ =0.93053	0.98143
-0.2	$\exp((-0.2+1.12)^2/(2*0.0316))$ =0.01452	0.01531
0.2	$\exp((0.2+1.12)^2/(2*0.0316))$ =0.00016	0.00017
1.0	$\exp((1.0+1.12)^2/(2*0.0316))$ =0.00000	0.00000
2.2	$\exp((2.2+1.12)^2/(2*0.0316))$ =0.00000	0.00000

Figure 3.6: Example of 1D BCJR



APP(0) = 0.2*0.05*0.1+0.2*0.08*0.3+ 0.3*0.08*0.1+0.3*0.12*0.3 =0.0190

APP(1) = 0.4*0.20*0.5+0.4*0.06*0.1+ 0.1*0.06*0.5+0.1*0.03*0.1 =0.0457

Output = +1

Figure 3.7: Example of APP determination for 1D BCJR

next symbol. α should be normalised each time it is determined, to prevent values all going down toward zero. All the α's are saved up to the end of the track.

Beta recursion follows the same trellis except it goes backwards this time.

After all the γ, α, and β are determined, the APP of each symbol being a 0 or a 1 is determined. This is done by taking each α, multiply it by the γ of the branches that originate from that state, and then to the β of the states to which they transit to. This produces the branch APPs. This is illustrated in figure 3.7. In the example shown in the figure, red numbers on the branches are the γ of the branches. All branch APPs that originate from states that have zero at the end of their state-symbol (00,10) are added as the APP of 0 as shown in figure 3.7. The others are that end in 1 (01,11) are added as the APP of 1. The one with the highest APP is chosen as the output (i.e. 1 in the example).

3.3.3 Full 2D ML detector

To get rid of the use of ZF, which limits the amount of noise that can be handled, because of its noise amplification, we also implemented full 2D ML detector.

We implemented the Full 2D ML detector by first using PR equaliser along track, to shape the signal to a selected target. The response across track for two or three track interference is assumed to be the target across track and therefore needs no equalisation. Viterbi detectors are then used for the ML detection, both along and across track. One of the ML detectors is a 2D multi-bits Soft Output Viterbi Algorithm (SOVA) detector, used to first eliminate the effect of either ISI or ITI. A regular 1D (Soft or Hard Output) Viterbi Algorithm (VA) detector is then used in the other dimension, to detect the signal out of the interference left.

Joint-Track ML Along Track
In the first implementation, 2D SOVA is used along track as a joint track detector, to detect the signal by removing the ISI. The likelihood (BMs) of receiving all possible adjacent bits across tracks is saved from the state with the best SM (shortest distance) in the history, for use in the next VA detector. There are four BMs per symbol for a two-track interference and eight BMs per symbol for a three-track interference.

Following this procedure means, memory enough to save 4 or 8 floating numbers for all bits of each track is needed, which will be used in the next VA algorithm.

The saved BMs of adjacent symbols from adjacent tracks are then used as the DMs of a VA that is carried out across the tracks. BMs and SMs are then determined and trace-back conducted after processing all tracks of given bit positions. The final detected bits (in hard decision VA) or likelihood of bits (in soft decision VA) are determined from this process.

Multi-Bits ML Across Track
In the second implementation, 2D SOVA is used across track as a multi-bit detector. It is used to detect the signal by removing the ITI. The likelihood (BMs) of receiving all possible successive interfered bits along track, is saved from the state with the best SM (shortest distance) in the history, for use in the next VA detector. In this case, memory enough to save 8 floating numbers (for 3-bit target along track) for all tracks is needed. This means less memory will be needed since we have

only 8 tracks in our implementation.

The saved BMs, for symbols of the same track, are then used as the DMs of a VA that is carried out along the tracks. BMs and SMs are determined and trace-back conducted after processing each symbol. The final detected bits (in hard decision VA) or likelihood of bits (in soft decision VA) are determined from this process.

Example of 2D SOVA Implementation

The 2D SOVA, described above, is a novel procedure designed to facilitate the full concatenation of a 2D SOVA to another ML detector. This arrangement makes it possible to reduce the complexity of using full 2D detection on multiple tracks. Rather than using a single 2D detector to detect 8 tracks as presented in works like [68], it uses two ML detectors, one along and the other across track, to carry out the detection process.

If a single detector is to be used for joint detection of 8 tracks with a target of length 3 along track, a detector of complexity in the order of $2^{8*3} = (16,777,216)$ is needed. But if the method presented here is used, with three-track interference, then the complexity will be of the order $8 * 2^{3*3} = (1,024)$ for the 8 tracks. Example of the 2D SOVA procedure is explained below, and illustrated in figure 3.8.

The example in figure 3.8 assumes a target of [0.4, 0.8, 0.8, 0.4] along track. It also assumes two track interference, where the contribution from both tracks is the same ($ITI = [1,\ 1]$).

As usual, the histories and SMs must be initialised at the beginning of the process, according to the knowledge of starting points of the data. The example assumes a detection depth (trace-back length) of four symbols. These are seen as the 4 columns of the history in each of the Present and Next state. The leftmost column (in green) in the

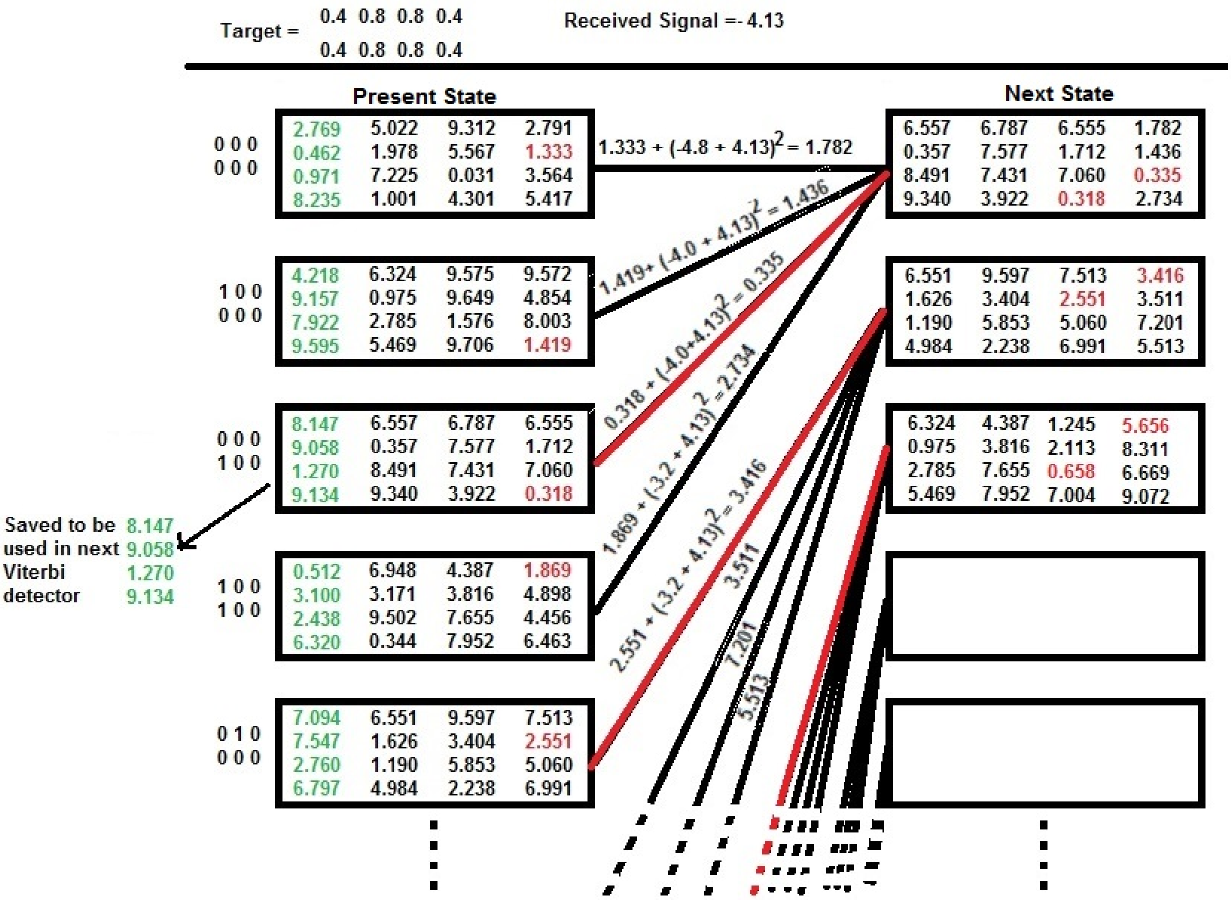


Figure 3.8: Example of 2D SOVA. State Metric=Red in each state. Green column in state with minimum state metric is the output.

present state is the oldest set of BMs, while the rightmost column is the newest. Figure 3.8 captures portion of the process at an arbitrary time when the received signal is assumed to be -4.13, and the history as shown in the figure.

The SM of each state is the minimum BM from the latest (rightmost) set of BMs. They are shown in red, in the Present state histories. To determine the output of the detector at this instance, the minimum among those SMs is determined. Assuming SMs of all the states not shown are higher than the ones shown, we can see the SM of the third state is the minimum. Therefore, the earliest column of BMs in green, from the third state, will be taken as the output. These 4 BMs represent the likelihood of 00, 10, 01, and 11. The two bit represent bits from two adjacent tracks interfered together.

Euclidean distance, of the received signal from the reference dibit responses of the branches, is determined as shown in the squared brackets on the branches. These are the DMs of the branches. Each DM is added to the SM of the state from which it comes from to get the BMs. The four BMs terminating in each state are now saved as the newest columns of the history for the Next state. The minimum of those new BMs in each state is the new SM for the state. History from the state where the new SM is derived is cut and appended to the history of the Next state (except the earliest history in green). These Next state metrics are now used in the next cycle, for the next received symbol, as the Present state metrics. The procedure continues until the end of the track or band of symbols is reached.

The four outputs of the 2D SOVA will now be used as DMs of a Viterbi detector applied across track. Across track means, the first symbol of the first track, the first symbol of second track, the first symbol of third track, and so on, are considered to be successive symbols. The same thing is done with second symbols of first, second and all other tracks. This continues until the end of the data is reached.

A similar arrangement can be implemented with 2D SOVA first applied across track then VA along track. The difference in procedure will be that there will be no need of finding output in each cycle. The final output across tracks can be found after all the symbols across tracks are processed.

As explained before, regular normalisation is needed to prevent overflow of registers. Register exchange is shown here because it is conceptually simpler. But trace-back method is preferred for hardware implementation, to reduce need of multiplexers for moving histories from one state to another.

3.4 Forward Error Correction

Forward Error Correction (FEC) is used in communication and memory channels to reduce the errors found in data by identifying and possibly correcting the errors obtained in the channel. In binary data, parity bits are the building blocks of FEC algorithms.

3.4.1 Coding

We implemented a system in which a single parity bit is added to a small block of data, and used to reduce the number of edges in the trellis during detection, and therefore reduce the space of possible errors to be made during detection. Two single parity systems separated by an interleaver, which is either block or Dithered Relative Prime (DRP) interleaver, are used in our implementation.

The first parity bit is an even parity scheme for a data of length 3 bits (xor of 3 consecutive bits). This makes the code length to be 4 bit and data rate reduced to 3/4. Total data bits per track is initially 2304 bits before adding any parity. An interleaver is then used to rearrange the data. Two types of interleavers are investigated. The matrix interleaver and the DRP interleaver.

In the matrix interleaver, the row of 3072 bits is converted into a matrix with three rows and 1024 columns. The data is then read a column after another. After which the second parity is added for each column of three bits.

The DRP interleaver investigated in this research consist of read and write dither of length $W = 8$ bits. Given, in a row of length L (3072), the position of the bit to be dithered in a row is "i", and the position to which it will be moved is "j" which range from 0 to $L - 1$. Then the relationship relating "i" and "j" can be given as

$$j = W\lfloor i/L \rfloor + [(a + b * i) \mod W] \tag{3.26}$$

where "a" and "b" are integer constants normally less than W. "b" is chosen to be a relative prime to W. That is "b'' and "W" have no common factor. Mod stands for modulo division.

The main randomiser, which applies to the row as a whole, is also evaluated using similar function as shown in equation 3.27 with "b'" being a relative prime of, and less than L.

$$j = (a' + b' * i) \mod L \tag{3.27}$$

A write dither similar to the read dither is also carried out before a second parity bit is then added after the interleaving. In our implementation, we added a single parity to a data block length of 3 again, but this time using odd parity. This is to make sure that no streams of all zero or all ones are possible for a very large number of bits. The odd parity on block length 3 is equivalent to a RLL code (0,6,3,4,1). This can, therefore, serve the purpose of clock synchroniser and FEC code during detection.

3.4.2 Detection and Decoding

BCJR is used in detection from the equalised channel signal. For the system implemented here, where code length is 4, a target length of at least 4 is needed to be able to simultaneously detect and decode the parity bit from the equalised signal. The parity bit added to the data modifies the trellis, such that at the point where the odd-parity bit comes into the equalised signal, the number of branches is reduced to 8 instead of 16, for a target length of 4. This is because only branches whose bit component satisfy the parity equation are possible. Figure 3.9 shows the way the trellis is modified by single odd parity in a code of length 4 and detector of target length 4 in 1D scenario.

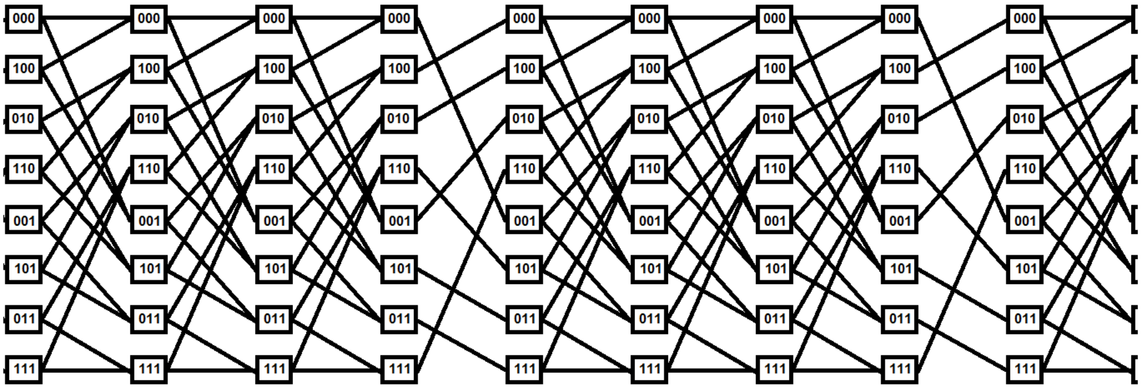


Figure 3.9: Trellis Modified by Odd Parity

At the fourth transition in figure 3.9, it can be seen that the valid transition from state [000] is to state [001]. This means, the component bits of the valid branch are [0001], which satisfy odd parity equation. The transition from [000] to [000] is not valid because, its component branch bits will be [0000], which does not satisfy odd parity equation. Therefore, the detector/decoder is designed to just select the valid branches whenever a parity bit is encountered. A soft output is saved in the history for further usage.

The history is de-interleaved to reverse the interleaving process. Deinterleavers are similar in structure to interleavers. They differ only in the constants "a" and "b" which are selected to reverse the rearrangement of the data.

The data is now non-convoluted soft information representing the probability of having a 0 or a 1 in a situation where there is no ITI. It may represent the probability of 00, 01, 10 or 11 which is convoluted across track in a situation where there is ITI from one adjacent track. These probabilities are now used in an MLSD or MAP detector along track, by utilising the first applied parity, which is decoded last.

In the MLSD case, the probabilities of any four block of data are multiplied according to possible sequence, considering even parity coding. For example, the likelihood of sequence 0000 is found by multiplying the probability of 0 for the first, second, third and fourth bit. The probability of 0011 is found by multiplying the probability of 0 for the first and second bit and the by probability of 1 for third and fourth bit. This continues for all the four-bit combinations that satisfy even parity equation. At the end, the values are compared to determine the sequence with the highest probability, and hence chosen as the output data.

The MAP detector goes further to add all probabilities of sequences found in the MLSD case, which have a bit in a state of 0 on one hand and those which have a bit in a state of 1 on another hand, for all the bit positions. For example, probabilities of 0000, 0011, 0101, 0110 are added to get the probability that the first bit is 0. Probabilities of 1001, 1010,1100,1111 are added to get the probability of the first bit being 1. The two are compared to determine the first bit. Similar thing is done for other three bit positions, which yields the final output.

3.5 FPGA Implementation

For determination and comparison of hardware requirement and to carry out more rigorous analysis of some of the detectors, FPGA implementation of the channel, ML along track, and ML across track were carried out. VHDL coding was used and implemented on a Tearasic Sockit FPGA board with a clock frequency of 50MHz. The board has a

Cyclone V FPGA IC on it. Altera Quartus II version 13.0 and version 14.1 are used for programming the board.

The VHDL code involves the main program (Top-level entity) where the “Channel” component and equalisers/Viterbi detector components were connected. Initialisations and Error calculations are performed in this program file (Entity ShingleA). The whole programme is pipelined to carry out the data generation, channel modelling, equalisation and detection on a certain window of data simultaneously.

3.5.1 Channel

The channel model generates random binary data, add ISI to it, add ITI and AWGN.

The Muller method of generating AWGN noise was used in the channel code [75]. The method uses equation 3.28 to generate the AWGN noise of unit variance and zero mean, where m_1 and m_2 are two uniform random numbers ranging from 0 to 1.

$$AWGN = cos(2\pi m_1)\sqrt{-ln(m_2)} \tag{3.28}$$

In place of Cosine, Sine function can be used. A polynomial approximation of Cosine function is used in the code, which is given in equation 3.29. The approximation is valid for “m” in the range 0 to 0.5. Consequently, for values of “m” in the range 0.5 to 1, the angle must be transformed by subtracting m from 1.

$$Cos(2\pi m) = 1 - 24m^2 + 32m^3 \tag{3.29}$$

To approximate $\sqrt{-ln(m)}$ inverse polynomials are used appropriate for various ranges of the function. Equation 3.30, equation 3.31 and equation 3.32 are used for “m” ranging from 10^{-5} to 10^{-2}, 10^{-2}

to 0.6 and 0.6 to 1.0 respectively.

$$\sqrt{-ln(m)} = 1.8674 + \frac{4.144e^{-3}}{5e^{-3} + m} + \frac{7.5e^{-5}}{1e^{-4} + m} \tag{3.30}$$

$$\sqrt{-ln(m)} = -0.465 + \frac{1.519}{0.7 + m} + \frac{0.01}{0.01 + m} \tag{3.31}$$

$$\sqrt{-ln(m)} = 1.821 + \frac{1.323}{-1.8 + m} + \frac{0.00167}{-1.01 + m} \tag{3.32}$$

Pseudo-random numbers of 16-bit data width are generated as m_1 and m_2 using linear congruential generator (multiply-with-carry (MWC)). The equations used for the MWC procedure are as given in equation 3.33 and equation 3.34.

$$m_n = (a * m_{n-1} + c_{n-1}) \mod b \tag{3.33}$$

$$m_n = (a * m_{n-1} + c_{n-1})/b \tag{3.34}$$

where "b" is 1 more than the maximum value of the random number which is preferably a prime number. m_n is the pseudo-random number generated at time n, and c_n is the value of quotient (carry) from the division of the sum-product expression by "b" at time n. If the value of "a" is appropriately chosen, the period of the random number is normally in the range $(ab/2) - 1$ [75]. In order to have AWGN with a higher period, different values of "a" were chosen for the two random numbers (m_1 and m_2). This results in a period of the AWGN being of the order $(a_1b/2-1)(a_2b/2-1)$, provided that the two periods have no common factor. For the codes presented in this research, $a_1 = 32739$ and $a_2 = 32718$ while $b = 65536$. The period of m_1 and m_2 are therefore 1,072,791,551 and 1,072,103,423 respectively. This makes the period of the AWGN to be about $1.15 * 10^{18}$.

Because the code is based on integer arithmetic, the equations of 3.29 up to 3.32 are scaled up to have the ms ranging from 0 to 65536 instead of 0 to 1, and the output scaled up to a scale of 2^{10} to 1.

Data generation is carried out by linear feedback shift register. The primitive polynomial used is $m^{20}+m^3+1$ with a period of about $2*10^6$.

The sequence in which the procedures are carried out to produce noisy data is as follows. A window of bits (eg. 31 bits per track) is created and initialised. The newest bits(same length as head response) are convolved to the head response along track to produced a symbol with ISI. This symbol will be passed on to process that will add jitter noise while at the same time the data register is shifted (eliminating oldest bits) and new bits for each track are generated and saved as the newest.

Similar registers are created for jitters. Transitions of the data along track are evaluated by subtracting the bits, then they are multiplied by white noise to form jitter. The newest transitions of the jitter are convolved to jitter response along tracks to generate jitter noise for each track and then added to the results from the process of ISI.

The symbols received from the process above are appropriately combined across track to add ITI. The results from these processes are then passed to where AWGN will be added. The result is passed to the detector while registers are shifted to give room for new data band. This concludes the channel modelling.

3.5.2 Across-Track Detector

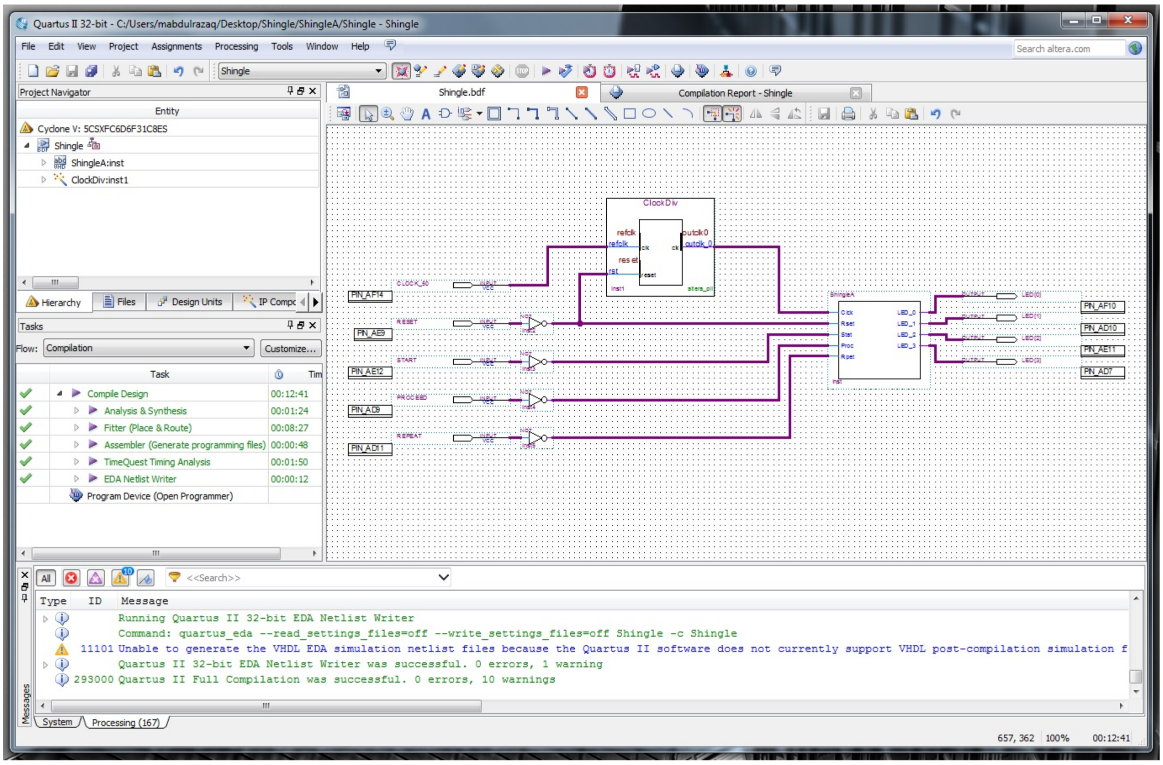
Across track ML detector was designed to cancel ISI by convolution of equaliser to the window of data passed into the detector. The results are moved into another register which performs Viterbi detection for each band of adjacent bits of data across all the tracks. The ITI amount determines the reference values with which to compare received data. The code presented in the appendix has two side-track ITI (three track interference). This means the constraint length of the detector is three and, therefore, will have 4 states with two transitions from each state.

3.5.3 Along-Track Detector

The VHDL code for the detector Along-track carries out the shaping of the data by convolution with shaping equaliser. The ITI cancellation is then carried out across track using ZF equalisers. Then Maximum Likelihood detection using Viterbi detector is carried out. A target length of 3 (with target [0.55 1.00 0.55] = [55 100 55] = [11 20 11]) is used in the code. This means the Viterbi algorithm will have 4 states each with two transitions. All these procedures are carried out on a portion of data in a pipelined fashion while keeping the original data until the output of a particular bit is available. The two are compared and that bit is moved out of registers while new data is generated.

3.5.4 Others

Because of multipliers and divisors in the channel model, the code runs slower than the clock of the board, therefore a phase locked loop (PLL) is used as seen in figure 3.10 to slow the clock speed to 10MHz. The FPGA implementation is designed to be controlled trough the onboard keys. The result is displayed in binary form through the LEDs on the board. The results are sent to output if 100 errors are recorded or 2^{22} bits along track are processed. 2^{22} makes sure up to 10^{-9} BER is reliably determined with about 100 errors. The output is streamed errors first, then the number of bits that produce the errors. From this values, BER is determined. The Quartus design and synthesis report can be seen in figure 3.10 and figure 3.11. A full procedure of the project



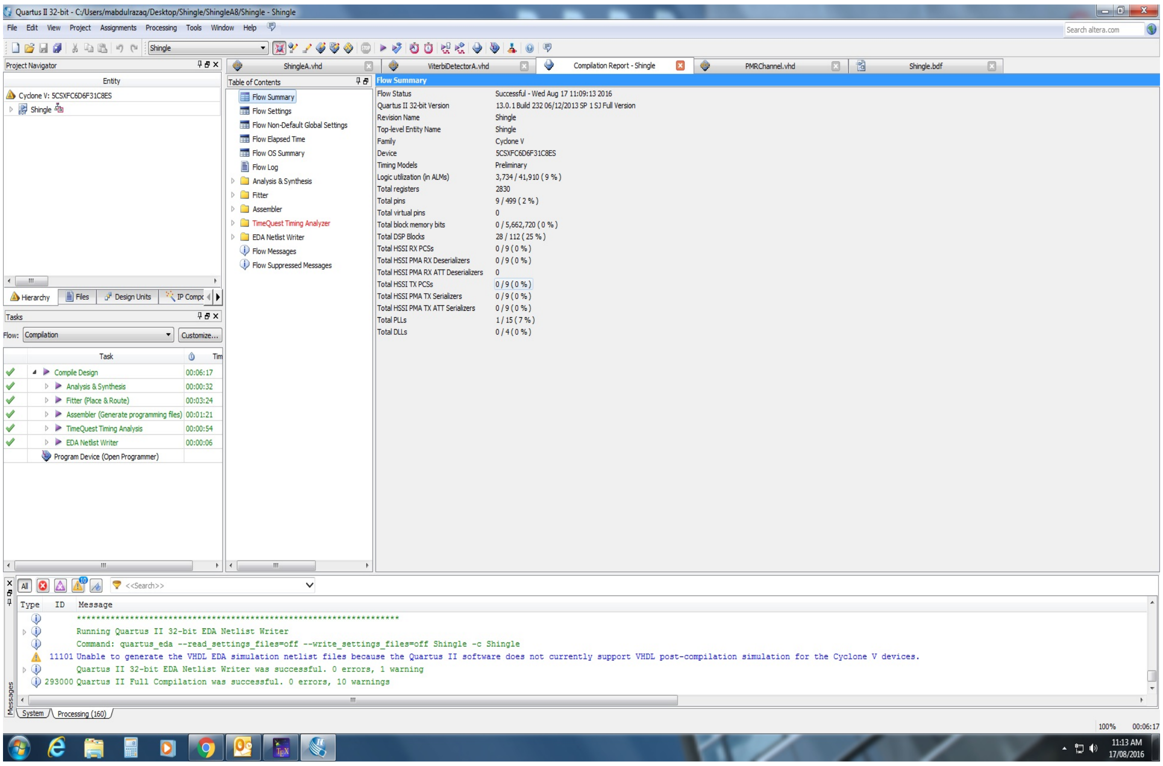
Quartus II 32-bit - C:/Users/mabdulrazaq/Desktop/Shingle/ShingleA/Shingle - Shingle
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Cyclone V: 5CSXFC6D6F31C8ES
Shingle
ShingleA:inst
ClockDiv:inst1
Shingle.bdf
Compilation Report - Shingle
ClockDiv
ShingleA
PIN_AF14
PIN_AE9
PIN_AE12
PIN_AD9
PIN_AD11
PIN_AF10
PIN_AD10
PIN_AE11
PIN_AD7

Flow: Compilation

Task	Time
Compile Design	00:12:41
Analysis & Synthesis	00:01:24
Fitter (Place & Route)	00:08:27
Assembler (Generate programming files)	00:00:48
TimeQuest Timing Analysis	00:01:50
EDA Netlist Writer	00:00:12
Program Device (Open Programmer)	

```
Type  ID     Message
             Running Quartus II 32-bit EDA Netlist Writer
             Command: quartus_eda --read_settings_files=off --write_settings_files=off Shingle -c Shingle
      11101  Unable to generate the VHDL EDA simulation netlist files because the Quartus II software does not currently support VHDL post-compilation simulation f
             Quartus II 32-bit EDA Netlist Writer was successful. 0 errors, 1 warning
      293000 Quartus II Full Compilation was successful. 0 errors, 10 warnings
```

Figure 3.10: FPGA Design Plan



Flow Summary

Flow Status	Successful - Wed Aug 17 11:09:13 2016
Quartus II 32-bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	Shingle
Top-level Entity Name	Shingle
Family	Cyclone V
Device	5CSXFC6D6F31C8ES
Timing Models	Preliminary
Logic utilization (in ALMs)	3,734 / 41,910 (9 %)
Total registers	2830
Total pins	9 / 499 (2 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	28 / 112 (25 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI PMA RX ATT Deserializers	0
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total HSSI PMA TX ATT Serializers	0 / 9 (0 %)
Total PLLs	1 / 15 (7 %)
Total DLLs	0 / 4 (0 %)

Flow: Compilation

Task	Time
Compile Design	00:06:17
Analysis & Synthesis	00:00:32
Fitter (Place & Route)	00:03:24
Assembler (Generate programming files)	00:01:21
TimeQuest Timing Analysis	00:00:54
EDA Netlist Writer	00:00:06
Program Device (Open Programmer)	

```
Type  ID     Message
             ******************************************************************
             Running Quartus II 32-bit EDA Netlist Writer
             Command: quartus_eda --read_settings_files=off --write_settings_files=off Shingle -c Shingle
      11101  Unable to generate the VHDL EDA simulation netlist files because the Quartus II software does not currently support VHDL post-compilation simulation for the Cyclone V devices.
             Quartus II 32-bit EDA Netlist Writer was successful. 0 errors, 1 warning
      293000 Quartus II Full Compilation was successful. 0 errors, 10 warnings
```

Figure 3.11: Synthesis Summary Report

creation, setting up and programming the board is presented in Appendix C. For further information about setting up and programming the board, [79] and [80] can be consulted.

Chapter 4

1D DETECTORS RESULT

The performance of Zero Forcing, MMSE equaliser, PRML Across-Track, and PRML Along-Track are presented in this chapter. The complexity of the various equalisation and detection techniques are stated. The relative advantages and disadvantages of the various equalisers and detectors are also discussed in this chapter.

4.1 ZF and MMSE Equalisers

ZF equaliser is effectively a special case of MMSE equaliser where the σ^2 variance, used to modify the equaliser (equation 2.19), is assumed to be zero. Therefore, the two are treated together here. There are a couple of factors that affect the performance of MMSE equalisers. Among them is the data density represented by T_{50}; the amount of ITI, and the length of the equaliser used to equalise the data.

In high-density SMR, jitter noise is more significant than white noise[37]. Therefore, we use a channel that has 80% jitter noise and 20% white noise. The equaliser taps can be of any length, but we will explore odd lengths here because the immediate PRML we present have odd targets (target length 3). Therefore, to get symmetric equaliser, we choose odd equaliser lengths.

To make sure we are making our result analysis using the best or close to best results, optimal equaliser lengths are first analysed for different conditions of ZF equaliser/detector. The effect of the length of the equalisers (equaliser taps) is shown in figures 4.1 a, b, and c for $T_{50} = 1.0$, $T_{50} = 1.5$ and $T_{50} = 2.0$ respectively. The channel has No ITI on the data.

Figure 4.1 (a), where $T_{50} = 1.0$, shows that an equaliser of 7 (7-taps) is adequate for optimal equalisation/detection. It can be seen in the fact that, longer equalisers do not give any better performance beyond that of length 7, as can be seen in the figure. But figure 4.1 (b) where $T_{50} = 1.5$, equaliser of length 7 shows inferior performance from other longer equaliser lengths. This is due to errors introduced by the equaliser itself. But equaliser of length 11 overcomes the errors and performs as good as higher order equalisers. Therefore as shown in figure 4.1 (b), length 11 is the optimal equaliser length for equalisation in a case where $T_{50} = 1.5$. In the case of $T_{50} = 2.0$ shown in figure 4.1 (c), an equaliser of 11 taps is adequate for values of up to 39dB SNR, with just a minute difference from 13 taps and above. Therefore, length 11 can be considered optimal within this range of SNR.

Because ITI is a linear combination of signals from adjacent tracks, the effect of equaliser along the track, on the central track will be the same as that on the side tracks. This means the optimal equaliser length in a situation where there is no ITI should be the same as when ITI is introduced. In next analyses of ZF equaliser, equalisers of 13 taps are used throughout, to make sure best results are used for all cases.

When ITI is introduced, an additional ITI cancelling equaliser (also ZF) is used across tracks as described in section 3.2.1. Three-track ITI, with similar interference from both side-tracks, is analysed and

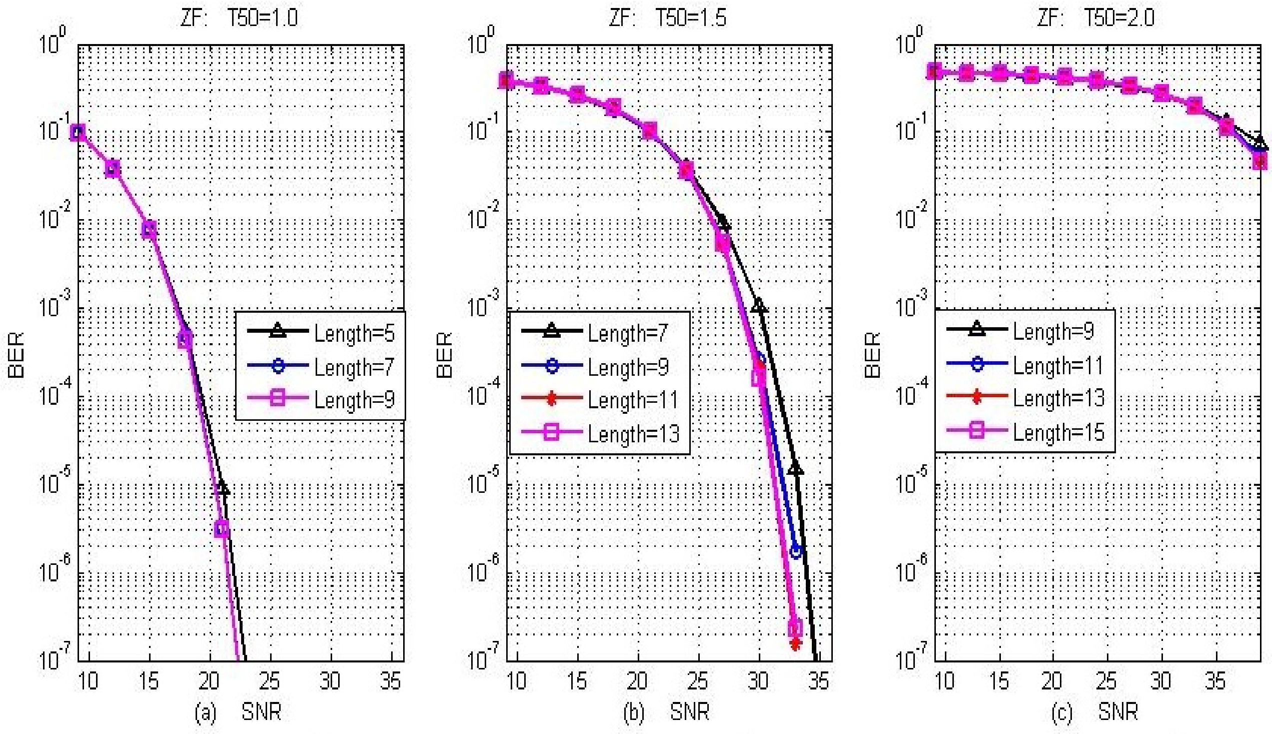


Figure 4.1: Effect of Equaliser length on ZF detector of different data density without ITI

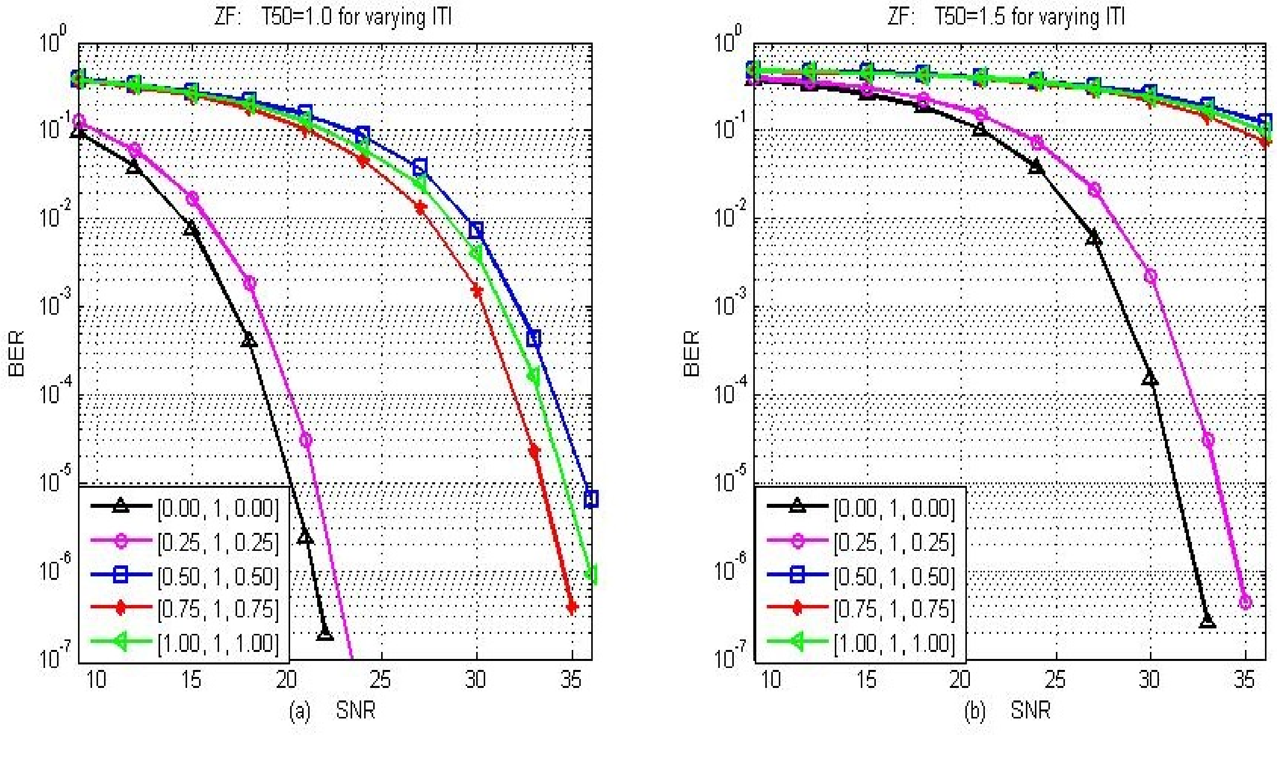


Figure 4.2: Effect of ITI in ZF detection

presented in figure 4.2 (a) and (b) for $T_{50} = 1.0$ and $T_{50} = 1.5$ respectively. Poorer performance is recorded as the ITI approaches 50% (0.5) from each side-track. When ITI is between 50% and 75% the detector recorded improvement on its raw BER performance. After that point, the ITI continued worsening up to 100% ITI (from each side-track).

From figures 4.1 and 4.2, it can be seen that there is deterioration of performance as the density along track (T_{50}) is increased (as expected). $T_{50} = 1.0$ is about 12dB better off than a situation where $T_{50} = 1.5$. About 15dB gain is recorded by density $T_{50} = 1.5$ over $T_{50} = 2.0$ in a situation where there is no ITI.

In an MMSE detector, the equaliser length required for best result is also analysed for situation where $T_{50} = 1.5$ and $T_{50} = 2.0$. Figure 4.3 (a) and (b) shows the results. It can be seen from figure 4.3 (a) that even an equaliser of length 9 is enough to optimise the result when $T_{50} = 1.5$. But for $T_{50} = 2.0$, a length of up to 13 is needed to get a good result. Therefore, from now on equaliser length of 13-taps along track is adopted for all other analyses.

When ITI is introduced in the channel for MMSE detection, and ITI cancellation equaliser is used, the performance recorded for $T_{50} = 1.0$, $T_{50} = 1.5$ and $T_{50} = 2.0$ is shown in figures 4.4 (a), (b) and (c) respectively.

The pattern of the performance is similar to that of ZF. The performance worsens until ITI of 50%. It then starts improving up to ITI of 75%, then continues reducing in performance up to when ITI is 100% from each of the side-tracks.

The trend of variation of the performance, as ITI changes, is more clearly shown in figures 4.5 (a) and (b) for a ZF and MMSE detector respectively at SNR of 18dB. The horizontal axes are the sum total of ITI from both sides of the central track. It can be observed from the

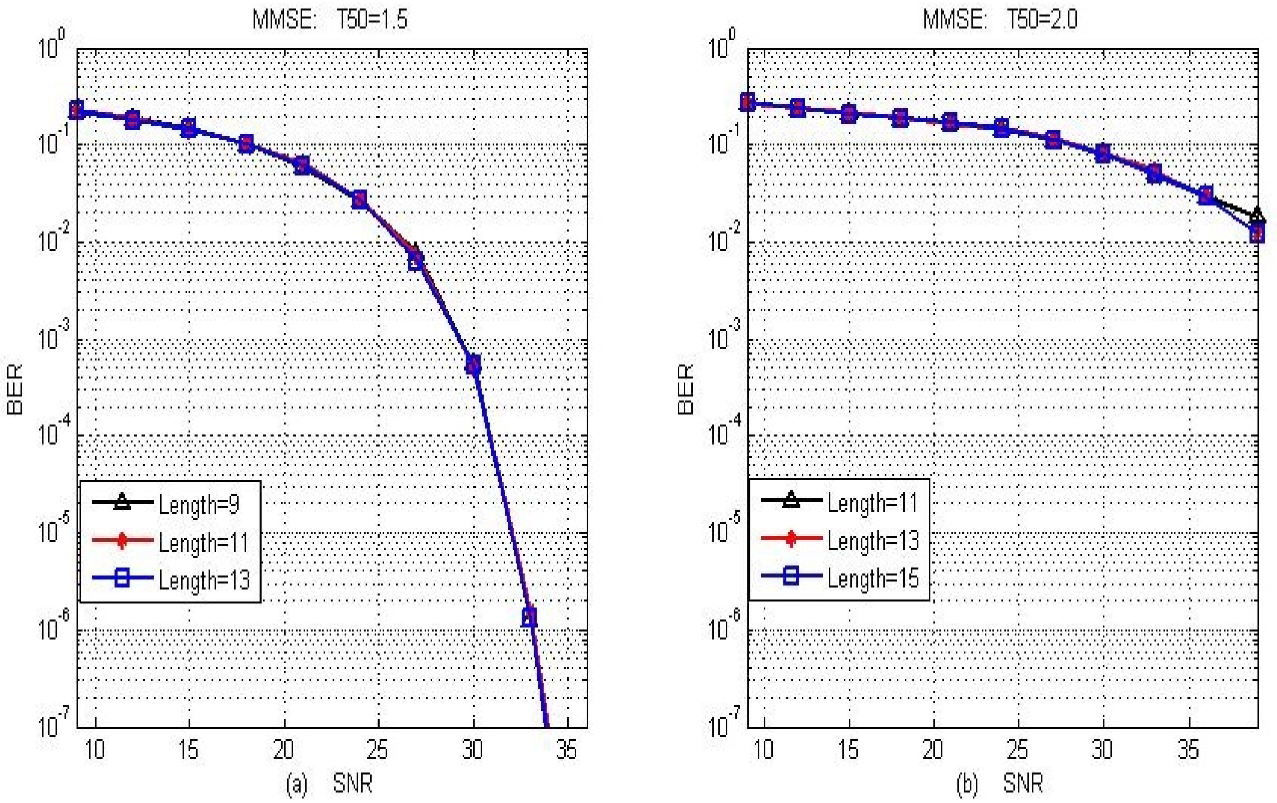


Figure 4.3: Effect of equaliser length on MMSE detector of different data density without ITI

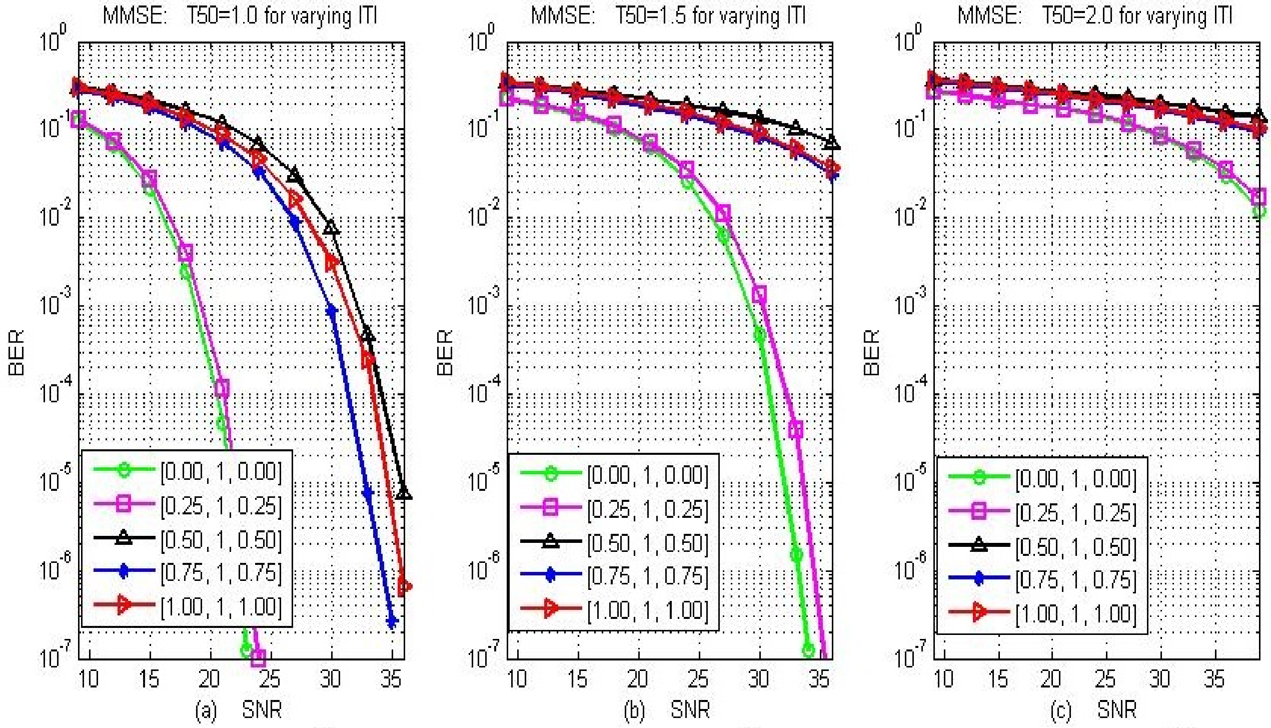


Figure 4.4: Effect of ITI in MMSE detection

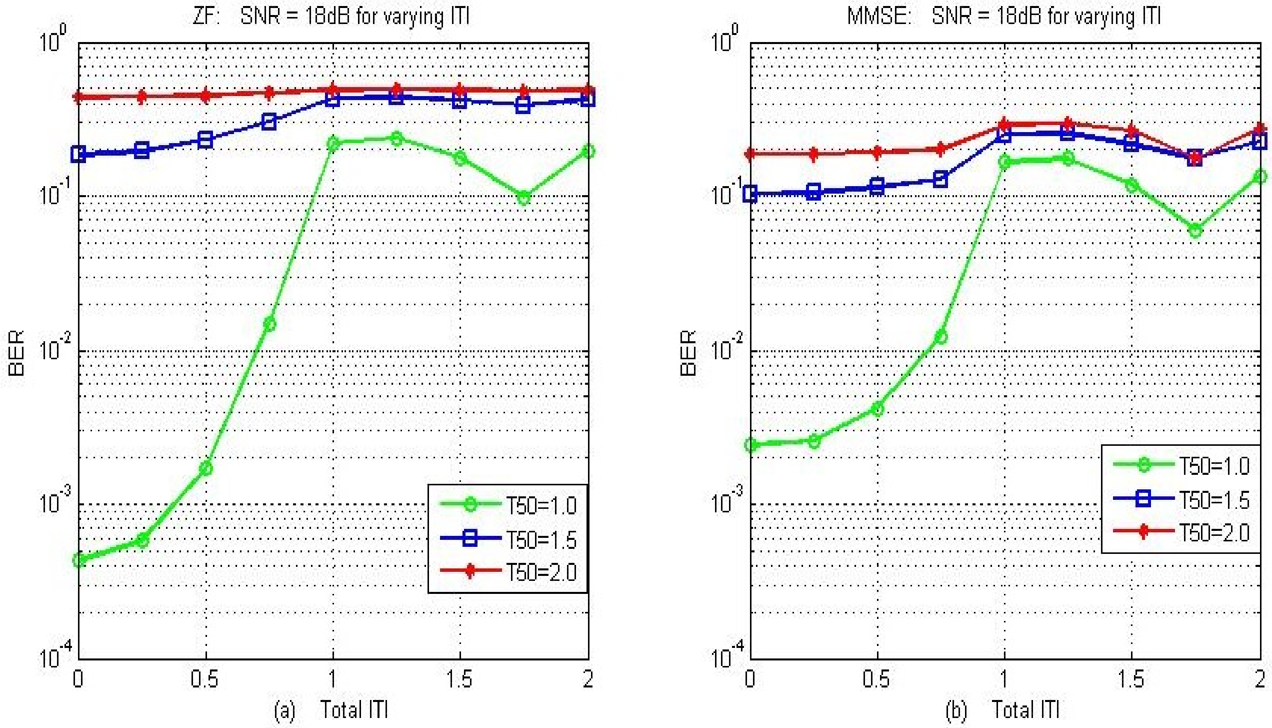


Figure 4.5: Performance as ITI varies for ZF and MMSE detector at 18 dB

figures that a local minimum in errors is achieved at total ITI of about 1.75 ($[0.875, 1.000, 0.875]$).

The pattern of performance as shown, when analysed closely, is strongly correlated to white noise amplification by ITI canceller. The power of the Additive White Gaussian Noise (AWGN) is amplified by sum square of the coefficients of the equalisers. The sum square of ITI canceller coefficients derived from equation 3.22 is plotted against the ITI from both side tracks in a three-dimensional graph as shown in figure 4.6.

Figures 4.5 (a) and (b) also show that at 18dB, ZF detector performs better than MMSE equaliser when the ITI is small while MMSE performs better for high ITI. A more detailed comparison can be seen in figure 4.7.

It can be seen that, when ITI, ISI and or noise is large, MMSE performs better than ZF. This can be seen from the larger difference in

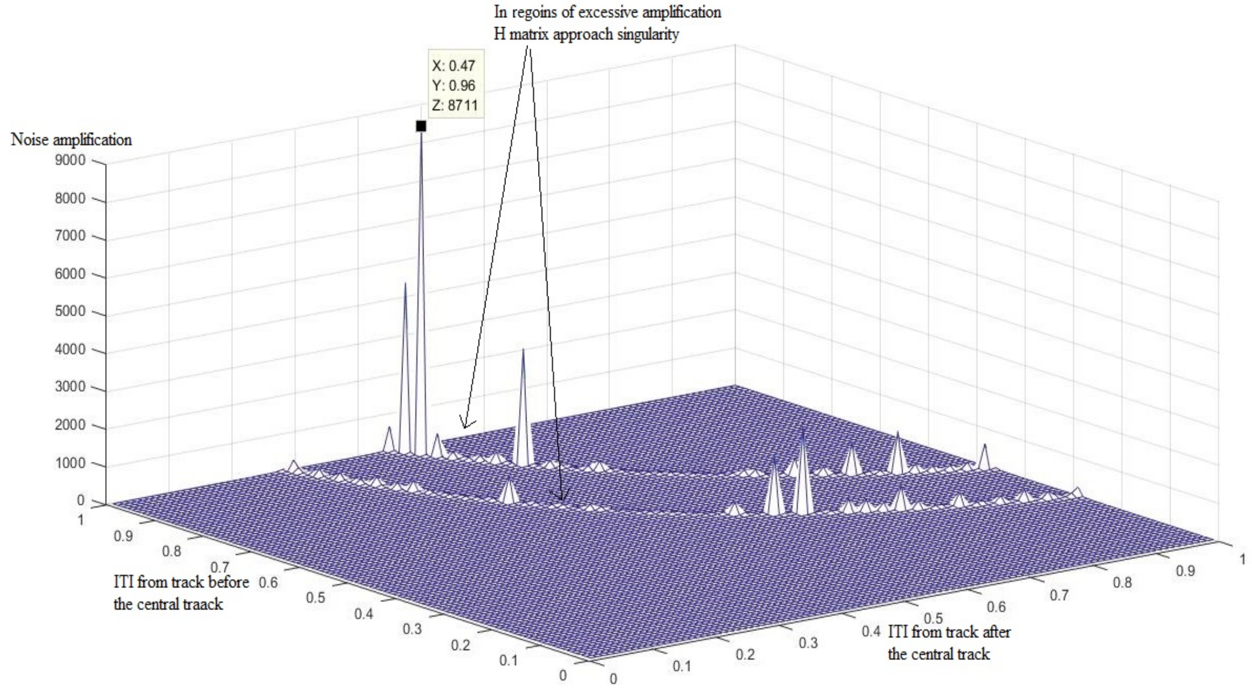


Figure 4.6: White Noise amplification by ITI canceller.

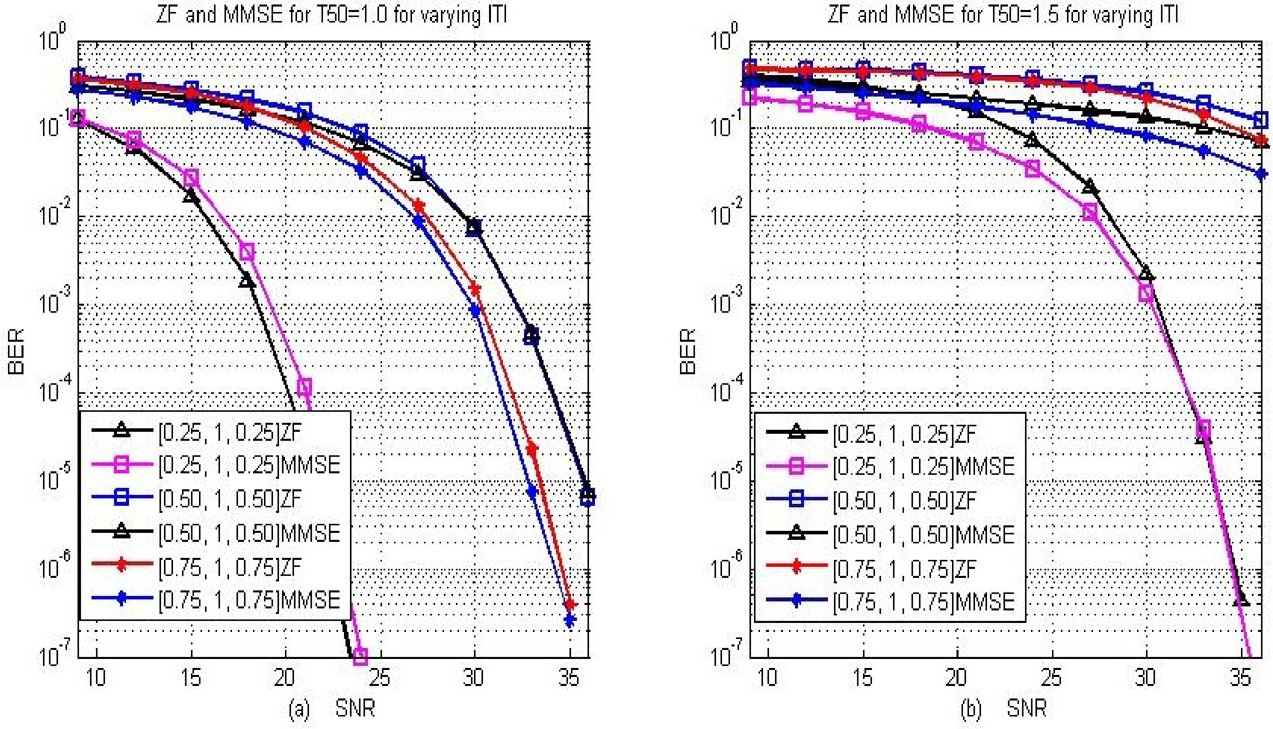


Figure 4.7: Comparing ZF and MMSE

performance between ZF and MMSE at $T_{50} = 1.5$, than between ZF and MMSE at $T_{50} = 1.0$, both for $ITI = [0.25\ 1.00\ 0.25]$ and lower SNR. But as SNR increases, the gap closes. This means MMSE will be more suitable to used only in case where there is large interference or noise.

The situations studied above are assumed to be for a read-head which is centrally located such that the read head has an equal amount of ITI from both sides of the central track. We can also assume the same size of read but skewed to one side such that, even though different amount of ITI comes from side tracks, the total sum of ITI is the same.

Figure 4.8 (a), (b) and (c) shows that for a total ITI = 0.5, the skewness of the position of read head does not affect the performance much. This is a situation where the position of head is known and used to determine the ITI canceller.

But figure 4.9 (a) and (b) shows better performance can be achieved in case where total ITI = 1.0. This is a situation where heavy noise amplification occurs and therefore skewing the read head facilitates the use of equaliser with less noise amplification. A gain of about 4dB is recorded by ITI [0.25, 1.0, 0.25] and [0.75, 1.0, 0.25] over ITI [0.5, 1.0, 0.5] at BER of 10^{-5}. Skewing read-head position, in a situation of heavier ITI, can change the equalisation task into a four-track interference problem, which is not within the scope of our investigations.

4.2 1D PRML Along-Track

In order to reduce the noise amplification of ZF and MMSE linear equalisers, ML detection is explored. PRML is employed and studied

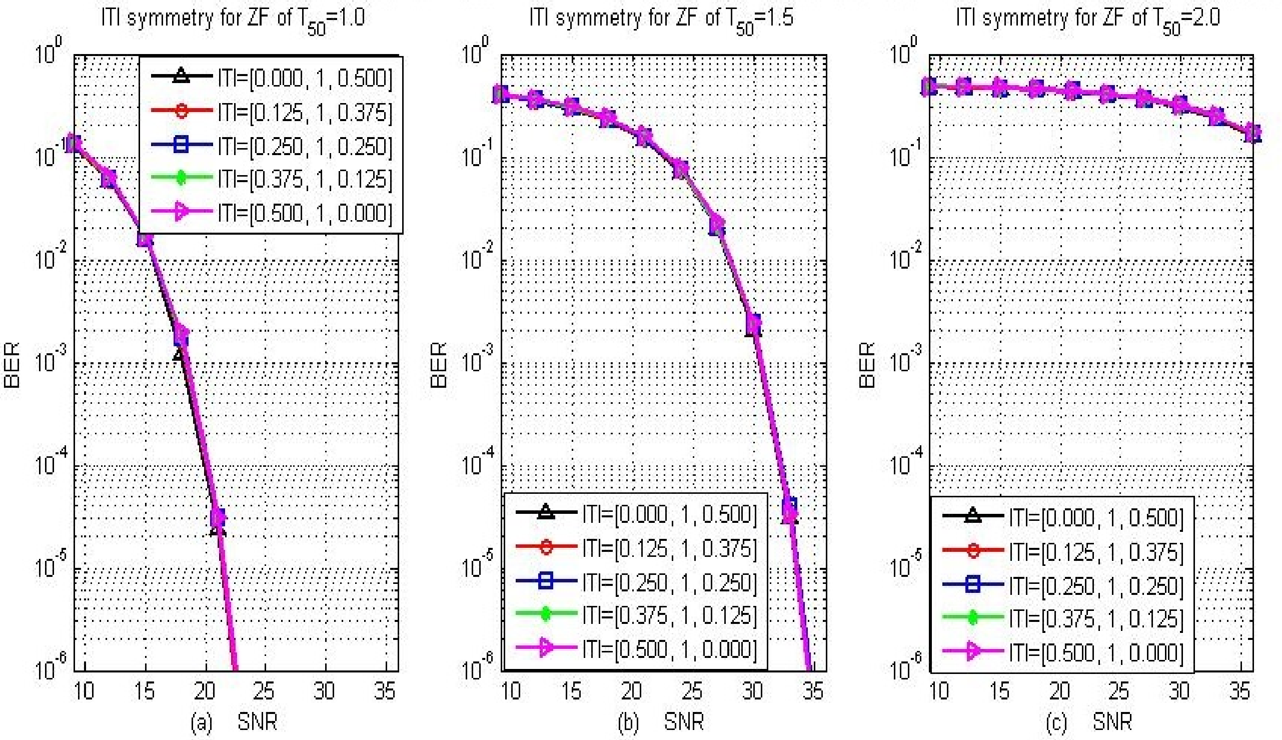


Figure 4.8: effect of ITI symmetry: total ITI=0.5

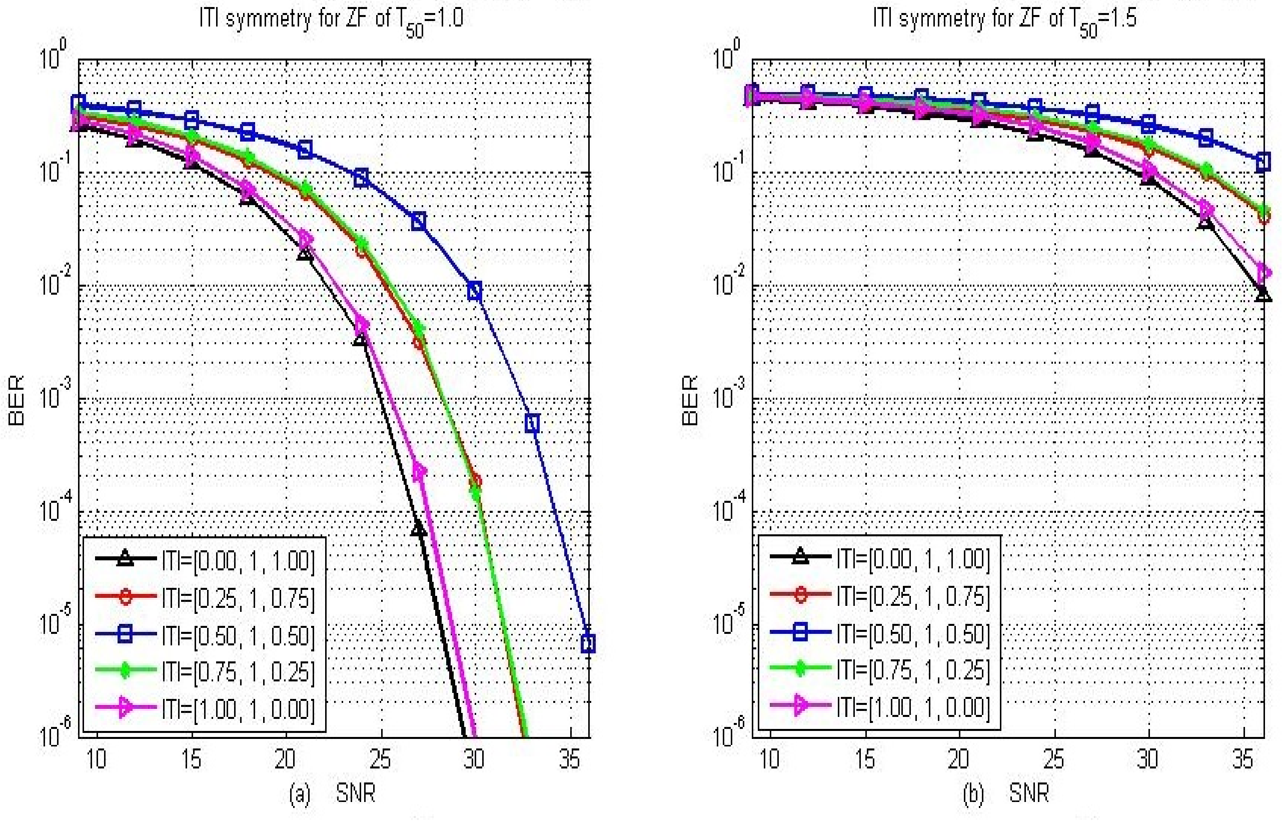


Figure 4.9: effect of ITI skewness: total ITI=1.0

as will be presented in the results here.

One approach to how we can eliminate the ITI as well as ISI is by using a linear equaliser to remove ITI, and then one-dimensional (1D) PRML to remove the ISI. This is called PRML along track here.

Selection of target is very important in order to get a good result that can be analysed rigorously. Targets of length 3 commonly used for PRML equalisation include PR2 target ([1 2 1]) and GPR targets such as ([4 6 4]) as used in [49]. GPR targets were investigated in this research. MMSE principle is used to determine the target, whose equaliser has the least noise amplification. It is investigated by calculating the rms of coefficients of the shaping equalisers. Since the shaping equaliser depends on T_{50}, it is also expected that the best target, based on rms, to depend on T_{50}. Figure 4.10 (a) shows the graph of noise amplification (rms of coefficients of shaping equaliser) for a data density at $T_{50} = 2.0$.

The middle coefficient of the target is set to 1, while the coefficient before it is "Target(0)", and the coefficient after it is termed "Target(2)". It can be seen that the noise amplification is minimum around the centre of the graph (figure4.10 (a)) when the target is symmetric. Further analysis of other cases also shows that the symmetric targets always produce the minimum noise amplification in our set up. Therefore, in figure 4.10 (b) we plot the noise amplification for targets that are symmetric. That is to say:
Target = [Target(0), 1, Target(2)]: where Target(0)=Target(2);

For low values of T_{50} the difference in amplification across the symmetric targets is small as seen in the figure 4.10 (b) and therefore small error in density calculations will not affect the result. Whereas high sensitivity will be present in situations where the value of T_{50} is large.

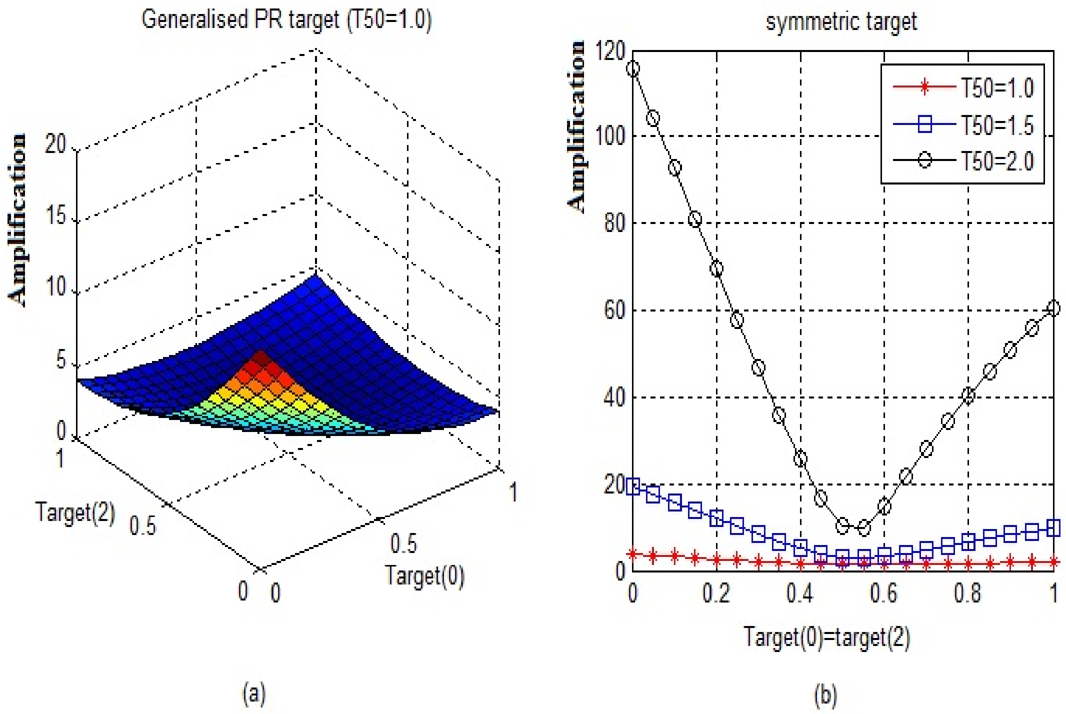


Figure 4.10: Noise Amplification for different Targets

The final detected signal is also affected by the ML detector, which will in turn be affected by the distance of dibit response of different symbols. The final error performance of the detector is analysed for the different targets after detection with the PRML. The results are presented in figure 4.11.

From figure 4.11 (a) for $T_{50} = 1.0$, it can be seen that small values of Target(0) and Target(2) performs better, even though from figure 4.10 (b) the minimum noise amplification is around 0.55. It can be deduced that this is because the difference in amplification of noise is not much, through all values of side target coefficient. Therefore, reducing the interference is more beneficial than reducing noise amplification.

Figure 4.11 (b) and (c) on the other hand show the noise amplification is very significant. That is why targets with least noise amplification give better performance. Targets with less ISI perform less. The best performance is found around when Target(0)=Target(2)=0.55 as can be observed from figure 4.10.

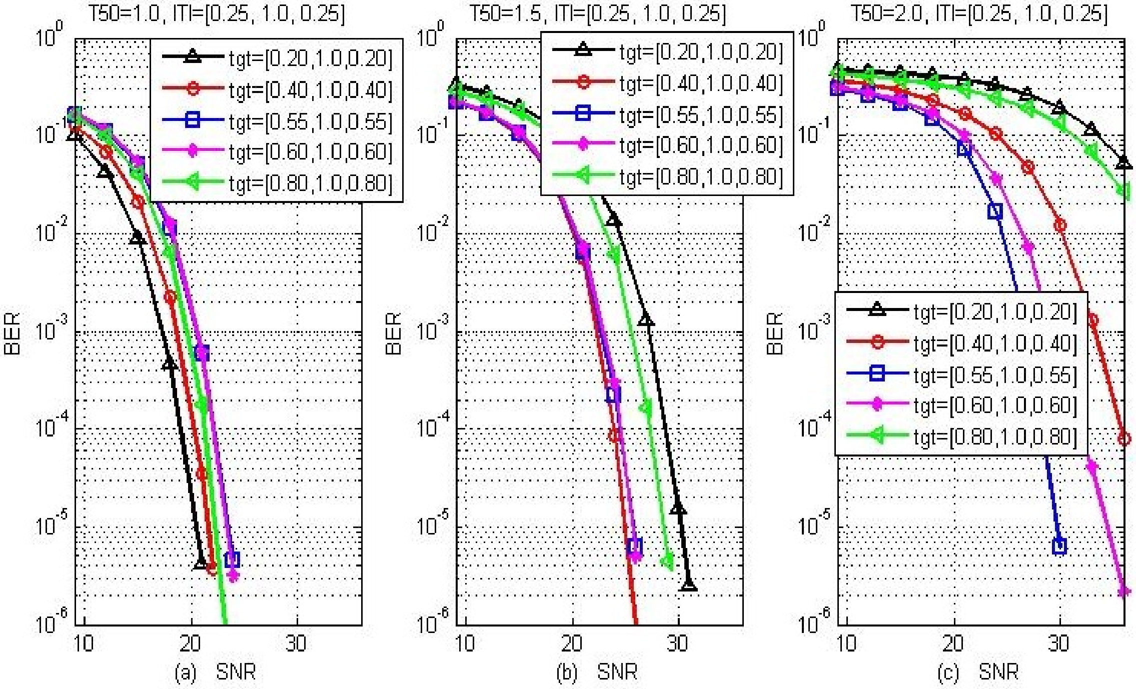


Figure 4.11: Performance of different Targets

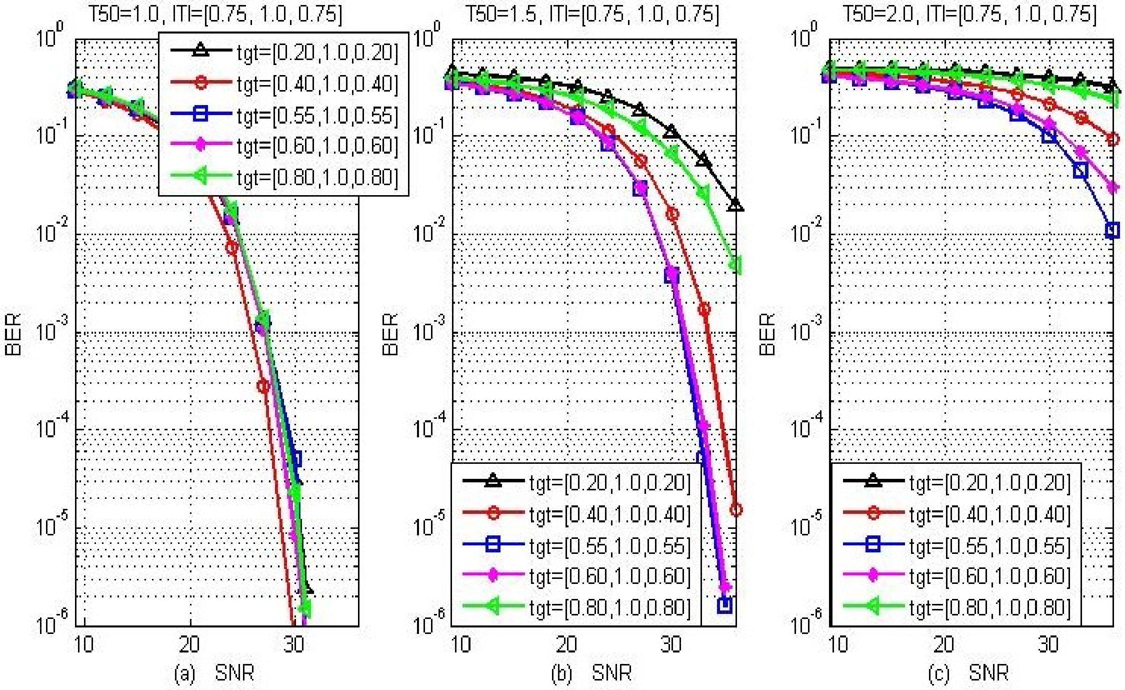


Figure 4.12: Performance of different Targets

The same pattern is recorded for a system with heavy ITI. Figure 4.12 shows the performance when ITI = [0.75 1.00 0.75].

Target [0.55 1.00 0.55], which is close to PR2 ([1 2 1]≡[0.5 1.0 0.5]) target, provides good separation between the values of different symbols, thereby helping the ML detector to provide good result at high density. Therefore, target[0.2 1.0 0.2] for $T_{50} = 1.0$ and [0.55 1.00 0.55] for higher T_{50} will be used henceforth for ML detection along track in this report, except where stated otherwise.

Since we are using Viterbi algorithm for the Maximum Likelihood (ML) detection, Trace-Back Length (TBL) also plays a role in the performance of the equaliser. In the results presented so far, a TBL of 10 bits is used. Very long TBL will increase the hardware requirement of the detector without necessarily providing much improvement. Therefore, in order to find the optimum TBL, a comparison of the performance of the detector was made for various TBL for different data density. Figure 4.13 (a), (b) and (c) shows the performance of different TBLs for $T_{50} = 1.0$, $T_{50} = 1.5$ and $T_{50} = 2.0$ respectively with heavy ITI of [0.75 1.0 0.75] present.

All the three situations presented here show that TBL of 8 units is adequate for a good result. Therefore, from this point on, it should be noted that the TBL of the result to be presented for PRML Along-Track is 8.

Using target [0.20 1.0 0.20] for $T_{50} = 1.0$ and target [0.55 1.00 0.55] for $T_{50} = 1.5$ and $T_{50} = 2.0$; and using TBL = 8, effect of varying the amount of ITI was investigated and figure 4.14 shows the results.

Similar property as ZF and MMSE is observed, where the performance reduces as the ITI increases until ITI of [0.5 1.0 0.5], then the

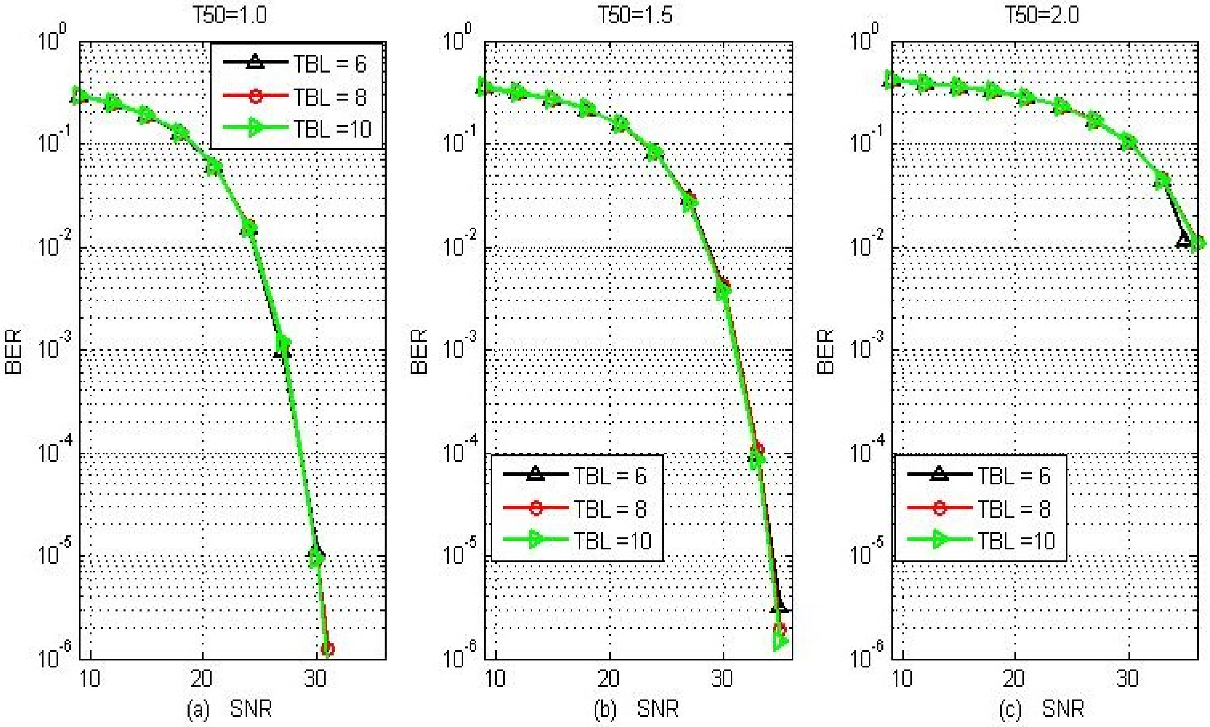


Figure 4.13: Performance of PRML along track for Trace back Lengths

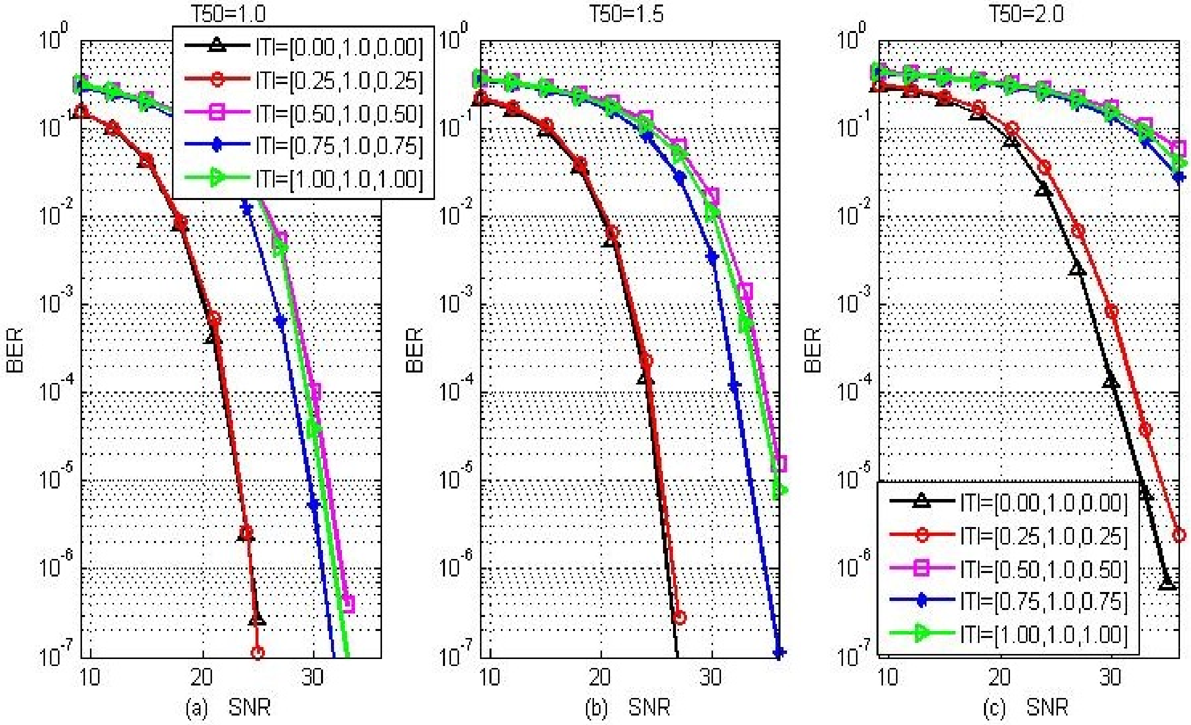


Figure 4.14: Performance of PRML along track for different ITI levels

performance increases up to [0.75 1.0 0.75] and then it continues reducing. This can be attributed to ITI cancellation as discussed under

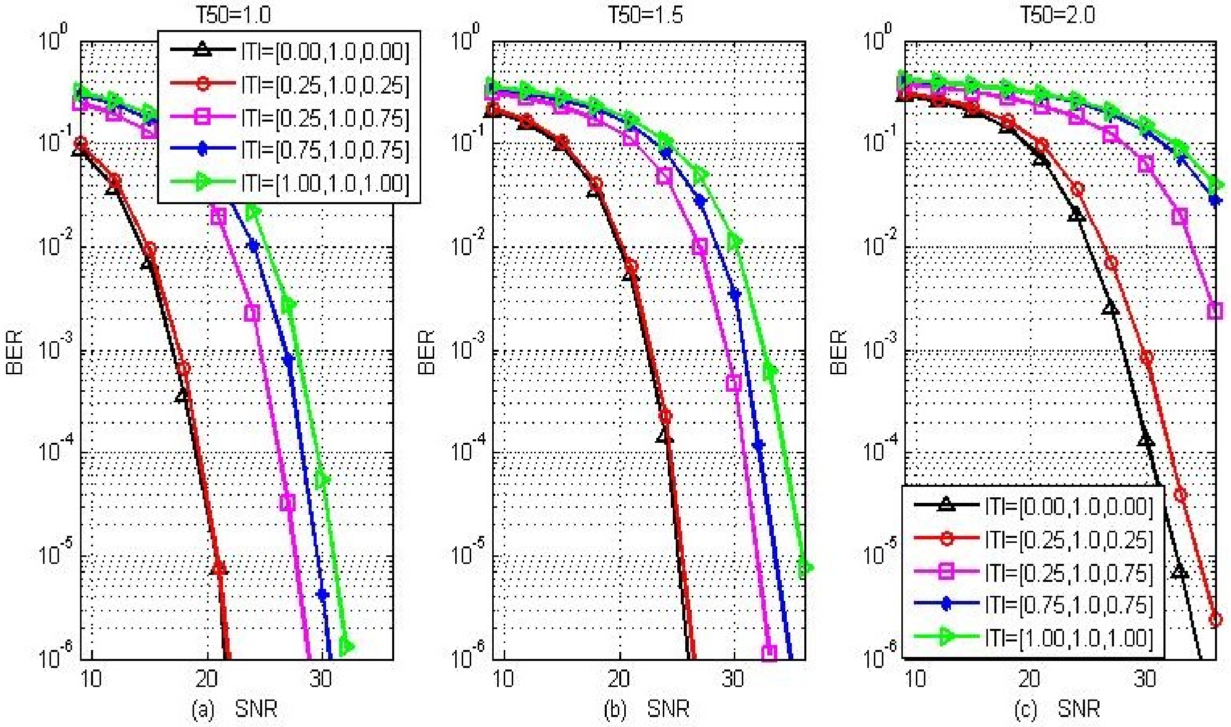


Figure 4.15: Performance of PRML along track for different ITI levels

results of ZF and MMSE. This is because PRML along track uses the same ITI canceller as the linear detectors. It is equally noticed that skewing the read head from the centre of the track helps in avoiding the heavy noise amplification at a point when ITI = [0.5 1.0 0.5] just as presented in the case of ZF and MMSE detectors. Figure 4.15 shows what the performance becomes, as read head of ITI = [0.5 1.0 0.5] is skewed to [0.25 1.0 0.75].

4.3 1D ML Across-Track

While PRML along track uses a linear equaliser for ITI cancellation and PRML for ISI cancellation, the equalisers can be applied the other way around. That is, to use Linear equaliser for ISI cancellation and ML for ITI cancellation. In this situation, the ISI cancellation equaliser is zero-forcing(ZF) not partial response(PR).

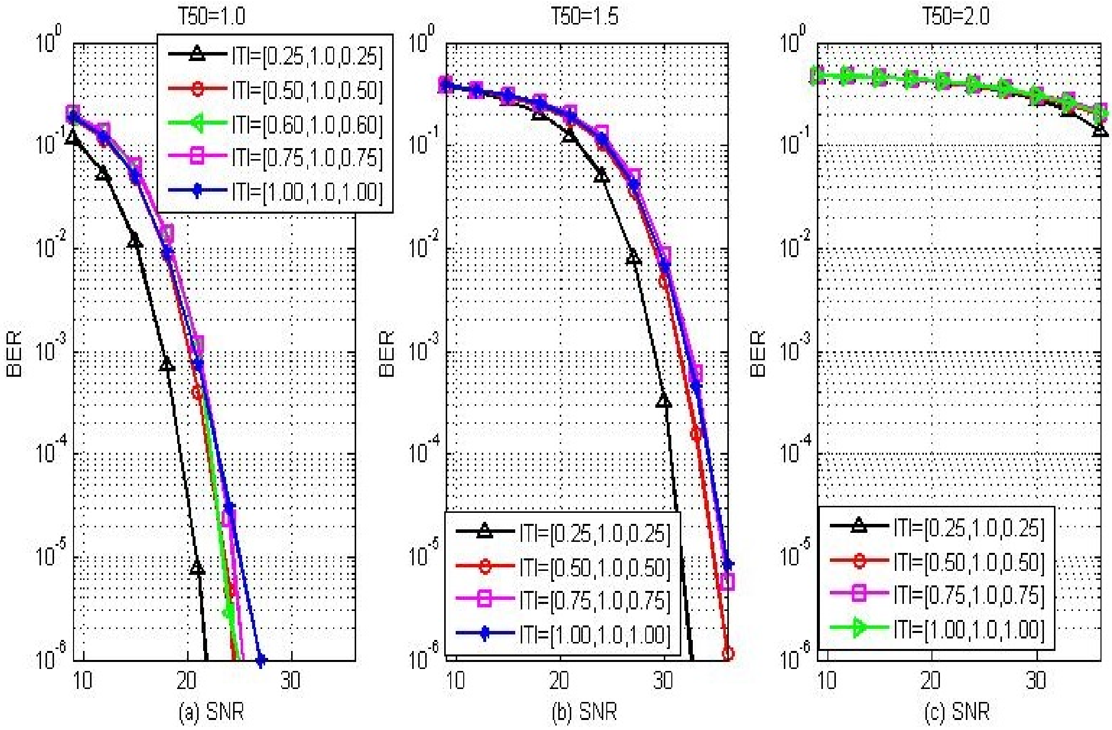


Figure 4.16: ML Across Track For Different ITI

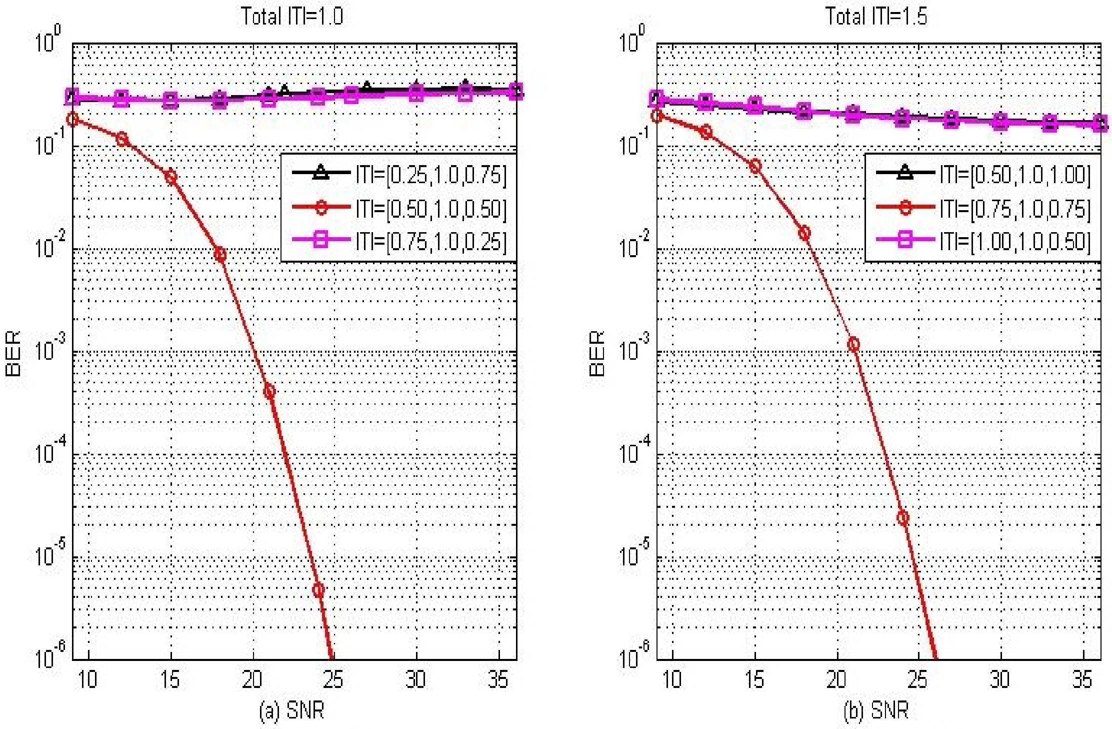


Figure 4.17: ML Across Track For Skewed ITI

The factors that may affect the performance of ML Across-Track detectors include, the length of the ZF equaliser (which is still maintained at 13 taps), T_{50} and ITI. Because only 8 symbols are to be equalised across the tracks, Traceback is not used. The whole symbols are processed before deciding the best sequence. Therefore, there will be no effect of TBL. And because ITI is the target, effect of different targets will be same as effect of ITI.

Figure 4.16 (a), (b) and (c) shows the effect of varying the ITI, for data density of $T_{50} = 1.0$, $T_{50} = 1.5$ and $T_{50} = 2.0$ respectively.

The performance degrades as amount of ITI increases as expected. In this situation, because the ITI canceller is not linear, the effect observed in previous cases is absent. We tried skewing the read head so that the amount of ITI from sides tracks is not the same. In this case, the performance is destroyed by the skewing the read-head from centre. Some results are shown in figure 4.17 (a) and (b) for total of ITI = 1.0 and 1.5 respectively. $T_{50} = 1.0$.

4.4 Comparison of the Detectors

A comparison of ZF, MMSE, PRML Along-track and ML Across-track were made to highlight the strengths and weakness of the detectors.

Figure 4.18 (a), (b) and (c) show the result of detection of signal with $T_{50} = 1.0$, with various degrees of ITI. The Total ITI (sum of ITI from side tracks) is maintained constant at 0.5, 1.0 and 1.5 in figure 4.18 (a), (b) and (c) respectively. In figure 4.18 (b) the ITI used for ZF, MMSE and PRML along track is [0.25 1.00 0.75]. This is because it was shown in section 4.1 and 4.2 that ITI of [0.5 1.0 0.5] has inferior performance to [0.25 1.00 0.75] due to excessive noise amplification.

The graph shows that for low ISI ($T_{50} = 1.0$) and low ITI (Total ITI

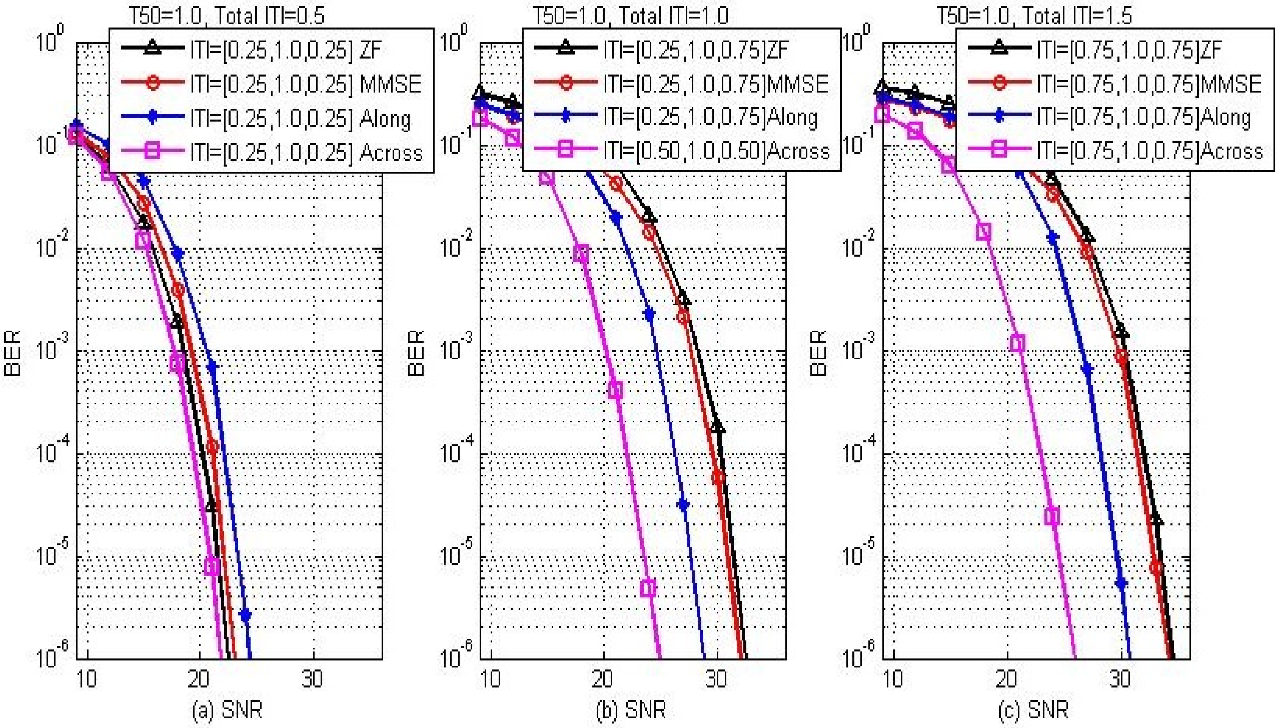


Figure 4.18: Comparison of Performance of the 4 detectors (T50=1.0)

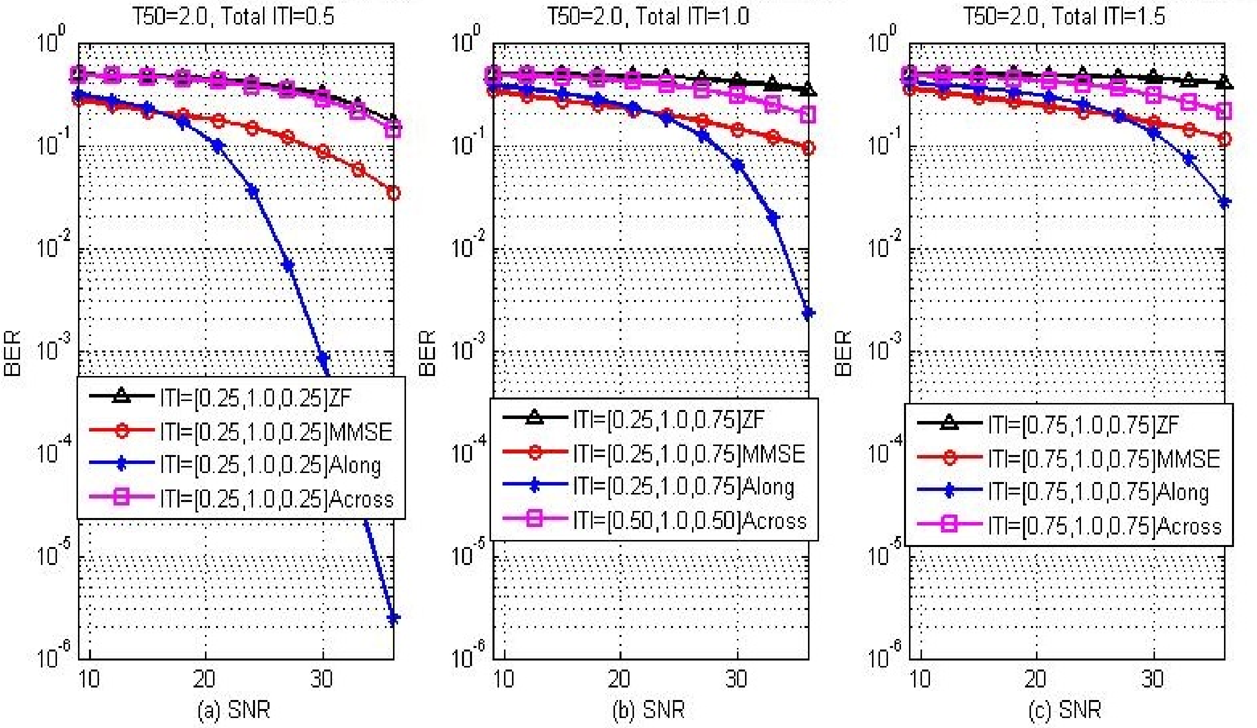


Figure 4.19: Comparison of Performance of the 4 detectors (T50=2.0)

= 0.5), PRML along track and ML across track perform almost the same with slightly better performance from PRML along track. ZF equaliser is close to 1 dB worse than the two while MMSE is about 2dB worse off. But when ITI increases (figure 4.18 (b) and (c)) ML across track gains performance over PRML along track and the rest. This is because it applies ML detection in the direction of increasing ITI. For a total ITI of 1.0, ML Across track has a gain of about 4dB over PRML Along track and about 7dB over ZF and MMSE at a BER of 10^{-6}. When total ITI = 1.5, ML Across track is about 6dB better than PRML along track and about 9dB better than ZF and MMSE for a BER of 10^{-6}.

Figure 4.19 (a), (b) and (c) shows similar information for high level of ISI ($T_{50} = 2.0$). When the ISI is high PRML along track performs far better than ML Across track, ZF and MMSE. In such situation, as shown in figure 4.19, MMSE also performs better than ML Across track and ZF with ZF having the worst performance of all.

A fairer comparison of the ML Across track and PRML Along track is to compare a situation in which there is similar density on both, with higher density across the width of the tracks for ML Across track detection, and higher density along the track for PRML Along track detection. We, therefore, assume a read head of the same size in the two situations:

- The first situation is where the bit width is the same as the read head width but the data density is $T_{50} = 2.0$ along track. This means there is negligible ITI across track (ITI=[0.0, 1.0, 0.0]) and therefore PRML Along track will be used for the detection of this signal.
- The second situation is where the bit width is half of read head width, but the bit is made double the first case along the track (That is $T_{50} = 1.0$). This case will have more ITI (ITI=[0.5, 1.0, 0.5]) therefore ML Across track will be used for the detection of

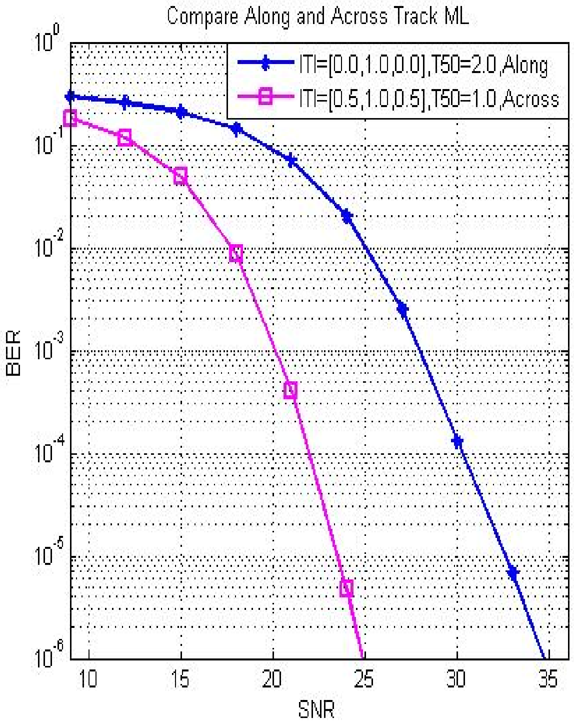


Figure 4.20: Comparison of Along and Across track ML

this signal.

Figure 4.20 shows the result of this comparison. It shows that much better result is achieved if the SMR is made denser across track and the density relaxed along track. A gain of about 10dB is achieved at a BER of 10^{-6}. But for the situation where the bits are wider along track ($T_{50} = 1.0$), it will take the read head longer (assuming the same head speed) to read the data, thereby reducing speed of the data access.

Results implying similar gain for using ML Across track over PRML along track are presented at “International Conference on Magnetism” which took place in Barcelona, Spain between 5-10 July 2015 (ICM2015). The title of the paper is “Two Dimensional Equalisation of Shingled Magnetic Recording Media”. Part of the results included are the performance comparison shown in figure 4.18 and 4.19 but only for the two 1D ML detectors. In the presentation, different SNR definition was used, which defines SNR as the ratio of signal energy to noise energy. This is different from the one used in this report which defines SNR as the ratio of the square of peak amplitude of the signal to rms of noise. Computational complexities of the two 1D ML detectors were also presented, which will be discussed in section 4.6. The publication also published a result for system with predominantly white noise unlike in this report where jitter noise is predominant.

4.5 FPGA results

Among the good performing results analysed so far, no error floor is noticed up to BER of 10^{-6}. The error floors were eliminated due to the assumption that guard bands have an all -1s saved on them and utilising the information in detection. It is also assumed in SMR, the last track of each sector has at least twice the width of a track. This is also used in the detectors. It should be noted that in normal magnetic medium, the guard bands that are unwritten to, will have random magnetisation. Therefore, if the all -1s is needed on the guard band, extra work has to be done to write at the beginning of any sector.

The error floor may rise in hardware situation with limited data sensitivity. The software model we implemented used 32 bit and 64 bit operating system. This accuracy may have very high complexity in hardware implementation such as FPGA. In order to rigorously investigate the position of the error floor further, more data need to be sampled with a reasonably accuracy. This led to the implementation of the ML Across-track and PRML Along-track detector on an FPGA board.

Figure 4.21 shows the performance of PRML Along track implemented using 16 bit FPGA input. The figures show that for 16 bit data there is not much difference in performance between FPGA 16 bit implementation and a software implementation whose result is pre-

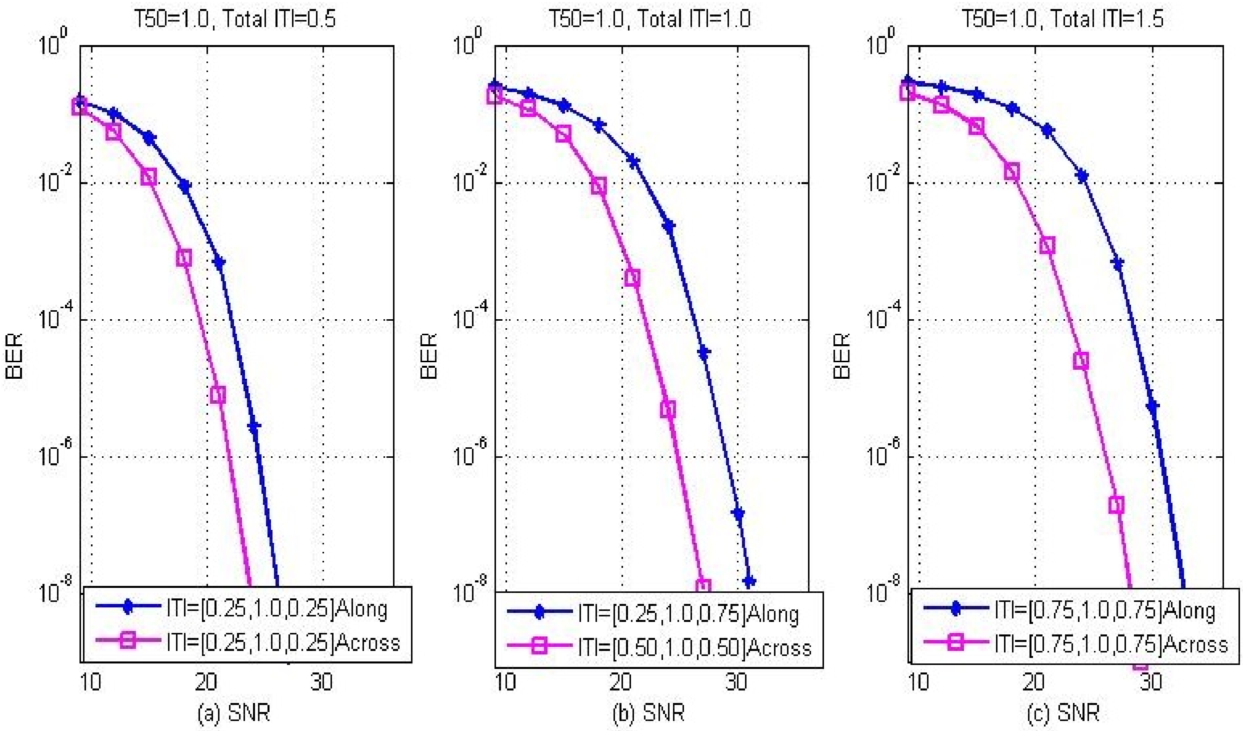


Figure 4.21: Comparison of Along and Across track ML

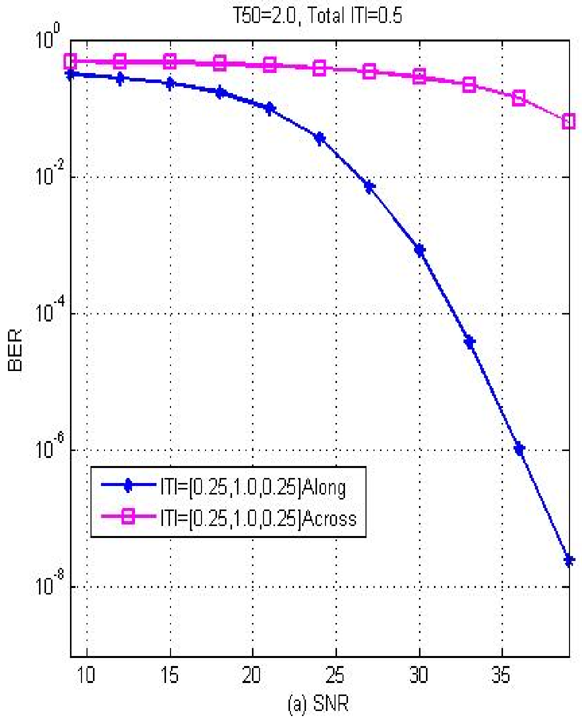


Figure 4.22: Comparison of Along and Across track ML

sented in figure 4.18. No error floor is also observed up to BER of 10^{-8}. Figure 4.22 also shows similarity at high density ($T_{50} = 2.0$) with no visible error floor.

4.6 Complexities

In the Viterbi detector, the amount of computation depends on the constraint length of the detector and the number of bits per symbol as explained. For binary input (0s and 1s or -1s and +1s) in an unrestricted situation, each state will have two possible sources. That means, two branch metrics have to be determined per state. In the computation of Branch Metric (BM) and the State Metric (SM), one multiplication and 2 additions are carried out. Therefore, the total number of multiplications needed for detection of a single bit is equal to the number of branches while the number of additions is 2 times the number of branches. After calculation of branch metrics, the two possible inputs (in 1D situation) must be compared to choose the surviving path in each state. This means the number of comparisons needed for choosing a surviving path is same as the number of states. To select the best path among all the possible states in a continuous run detector (the one used in ML Along track detector), the metrics for all the states has to be compared. This means, there must be one less than the number of states, in comparisons. But in a case where there is a definite state in which the sequence ends (the case for Across-track), the number of comparisons decrease to the number of possible finishing states.

For linear equalisers, all the coefficients are multiplied by the data. This means there will be an equal number of multiplications as the length of the equaliser. The results of the multiplications are then added together. Therefore, the number of additions needed is one less than the length of equalisers.

For this comparison, the two 1D ML detectors are assumed to work on a system with two side-track ITI. This means the ML Across track detector will have a constraint length of 3, thereby having 4 ($2^{(3-1)}$) states and a total of 8 branches. The PRML Along track detector is assumed to have a target length of 3, therefore, it also has 4 states. The length of the linear equaliser is assumed to be 13 for either ISI cancellation or target shaping. Table 4.1 shows the complexity of PRML Along track and ML Across track detectors. The values with unit "A" represents the number of additions. The ones with unit "M" represent the number of multiplications and the ones with unit "C" shows the number of comparisons. The numbers are the amount of computations carried out per bit of data produced.

	Shaping	ITI cancellation	ISI cancellation	total
Across track	0A, 0M, 0C	16A, 8M, 5C	12A, 13M, 0C	28A,21M,5C
Along track	12A, 13M, 0C	7A, 8M, 0C	16A, 8M, 7C	35A,29M,7C

Table 4.1: Computational complexities

The results show that, for similar size of detectors, ML Across-track will have less complexity and potentially better performance if T_{50} is chosen to be small. But the gain in fewer computations may not be justified if T_{50} is large. In general, the order of the complexity is still the same. That is, it is exponential with respect to target length.

Chapter 5

FULL 2D DETECTORS RESULT

The results so far presented show a situation with target lengths of 3 bits and an ITI of at most 3 track for 1D detectors. In this chapter, we are going to present full 2D detectors that utilise the concatenation of two ML detectors, to reduce the complexity of ML detector which will be needed, if a single 2D detector is to be used to process 8 tracks. Performance of using longer target length of four bits will also be investigated. This is expected to reduce the white noise amplification by the shaping equalisers. A reduction in the complexity of the Full 2D detectors is achieved when the ITI is reduced to a maximum of two tracks.

Both SOVA and BCJR based ML detectors were implemented, and their results presented in this section. At the end of the chapter, results of adding parity bits for error correction are presented.

5.1 Full 2D Detectors

To reduce the weakness of linear equalisers when interference increases, as shown in results presented, two versions of the implementation of Joint Multi-Bit (JMB) detectors were carried out. The first is the Joint-Track detector which carries out Multi-track SOVA along track and then VA across track. The second version carries out Joint-Bit SOVA across track, and then VA along track. Full explanation of the

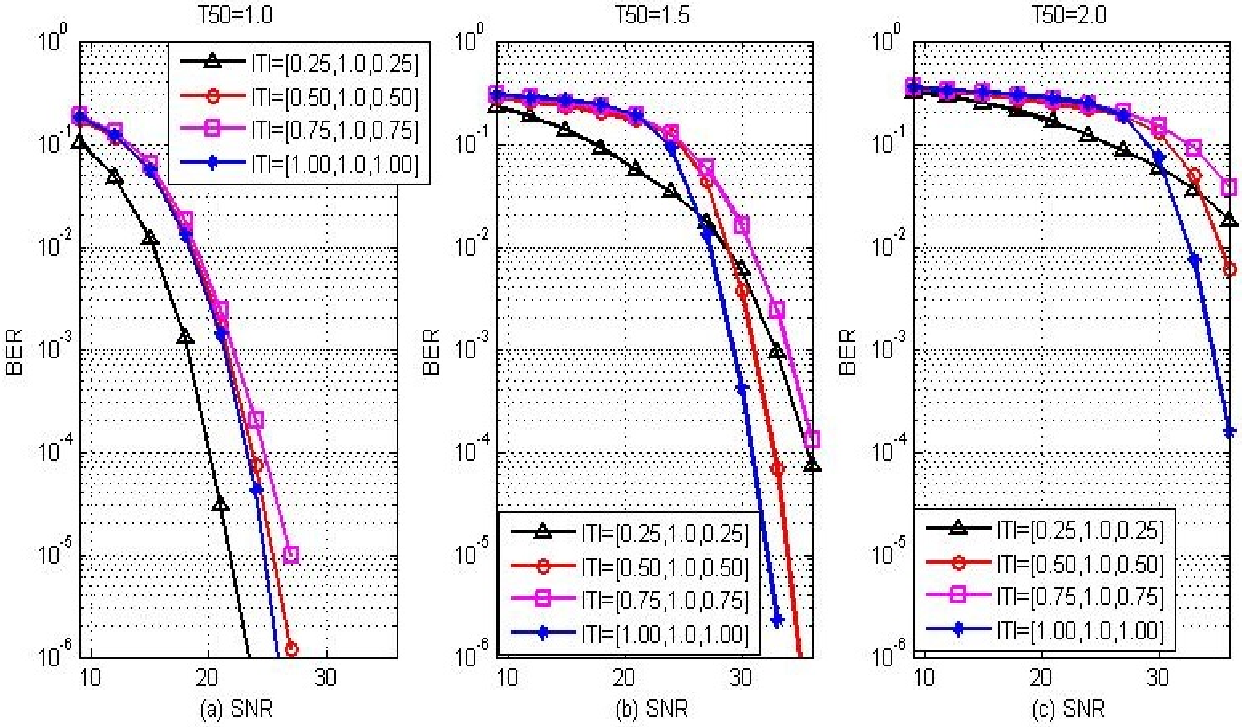


Figure 5.1: Performance of Full 2D Along track detector

detectors can be found in chapter three.

5.1.1 Full 2D Along track detector

Figures 5.1 (a), (b) and (c) show the performance of detector that uses full 2D SOVA to detect the signal along track. It removes the effect of ISI in the process and produces branch metrics that are used as distance metrics for the next process. Viterbi Algorithm is then used with the information gained from the 2D SOVA but now used across track. The final results are as displayed in the figures.

Figure 5.1 (a) shows that the performance of the detector is best when ITI is low ([0.25 1.0 0.25]) and ISI is low ($T_{50} = 1.0$). But as ITI increases, the performance shows that ITI [1.0 1.0 1.0] has the better performance than the rest which is followed by ITI [0.5 1.0 0.5]. ITI [0.25 1.0 0.25] has a gain of about 3dB over ITI [1.0, 1.0, 1.0] and almost 5dB compared to other ITIs. This can be attributed to the fact that the minimum distance among the different possible symbols in [1 1 1] and [0.5 1 0.5] is higher than in the rest, thereby making the SNR higher. The ITI of [1 1 1] captures more energy than the ITI of [0.5 1.0 0.5] and therefore has a better advantage. But in actual practice, it will be impossible to capture the energy of the side tracks without picking up more ITI from further tracks. It is, however, presented here just for analysis purpose.

Figure 5.1 (b) and (c) show the ITI[1 1 1] and [0.5 1.0 0.5] perform even better than the low ITI case of [0.25 1.0 0.25] when the condition are more extreme ($T_{50} \geq 1.5$). ITI[1 1 1] has a gain of about 2dB over [0.5 1.0 0.5] and much higher gain over other ITIs. As we have mentioned, ITI[1 1 1] and [0.5 1.0 0.5] have the advantage of high minimum separation between their symbols. Table 5.1 shows the possible symbols and their minimum separations for the four ITI conditions. It shows that ITI [1 1 1] has the best separation and therefore will have the least detriment to SNR.

ITI	Symbols(dibit)	Minimum separation	Ratio to peak amplitude
[0.25 1.0 0.25]	1.5, 1.0, 0.5,-0.5,-1.0,-1.5	0.5 (eg. 1.5-1.0)	0.33
[0.50 1.0 0.50]	2.0, 1.0, 0.0,-1.0,-2.0	1.0 (eg. 2.0-1.0)	0.50
[0.75 1.0 0.75]	2.5, 1.0, 0.5,-0.5,-1.0,-2.5	0.5 (eg. 1.0-0.5)	0.20
[1.00 1.0 1.00]	3.0, 1.0,-1.0,-3.0	2.0 (eg. 3.0-1.0)	0.67

Table 5.1: Possible Symbols and Minimum Separations

Results for Full 2D detector Along track are published from this research at Telecommunication Forum Conference, which took place at Belgrade Serbia, between 24^{th} to 26^{th} of November 2015 (Telfor 2015). The paper is titled “Concatenated 2D SOVA for Two Dimensional Maximum Likelihood Detection”. The result in the publication is presented using definition of SNR which considers jitter noise and white noise before ITI is added.

5.1.2 Full 2D Across track detector

In another implementation of the full 2D ML detection, the 2D SOVA is applied across track first to remove ITI effects, before Viterbi algorithm is then used along the tracks to detect the signal out of the ISI. The results obtained using this technique are presented in figure 5.2.

In this situation, the performance of low ITI situation (ITI=[0.25 1.0 0.25]) is significantly better for all densities observed in figure 5.2 (a), (b) and (c). But for higher amounts of ITI, the performances are very close in situation where T_{50} is 1.0 or 1.5, with small differences in the performance for $T_{50} = 2.0$ in favour of Lower ITI. ITI=[0.25 1.0 0.25] has a gain of about 10dB at $T_{50} = 1.0$ for BER of 10^{-6}. For $T_{50} = 1.5$, the gain reduced to about 6dB.

5.1.3 Comparison of Full 2D detector

Comparison of performance of the two Full 2D techniques is presented here in figure 5.3. For a very low density along-track ($T_{50} = 1.0$) and low ITI ([0.25 1.0 0.25]), the performance of the two methods is similar as shown in figure 5.3 (a). But when the T_{50} is increased to 2.0, the Version that uses SOVA across track outperforms the version which uses SOVA along track.

For higher ITI (figure 5.3 (b) and (c)) Using SOVA along track produces better performance than using SOVA across track for low ISI ($T_{50} = 1.0$). SOVA along track has a gain of about 6dB at BER of 10^{-6}. But when $T_{50} = 2.0$, the performance gain changes in favour of SOVA across track.

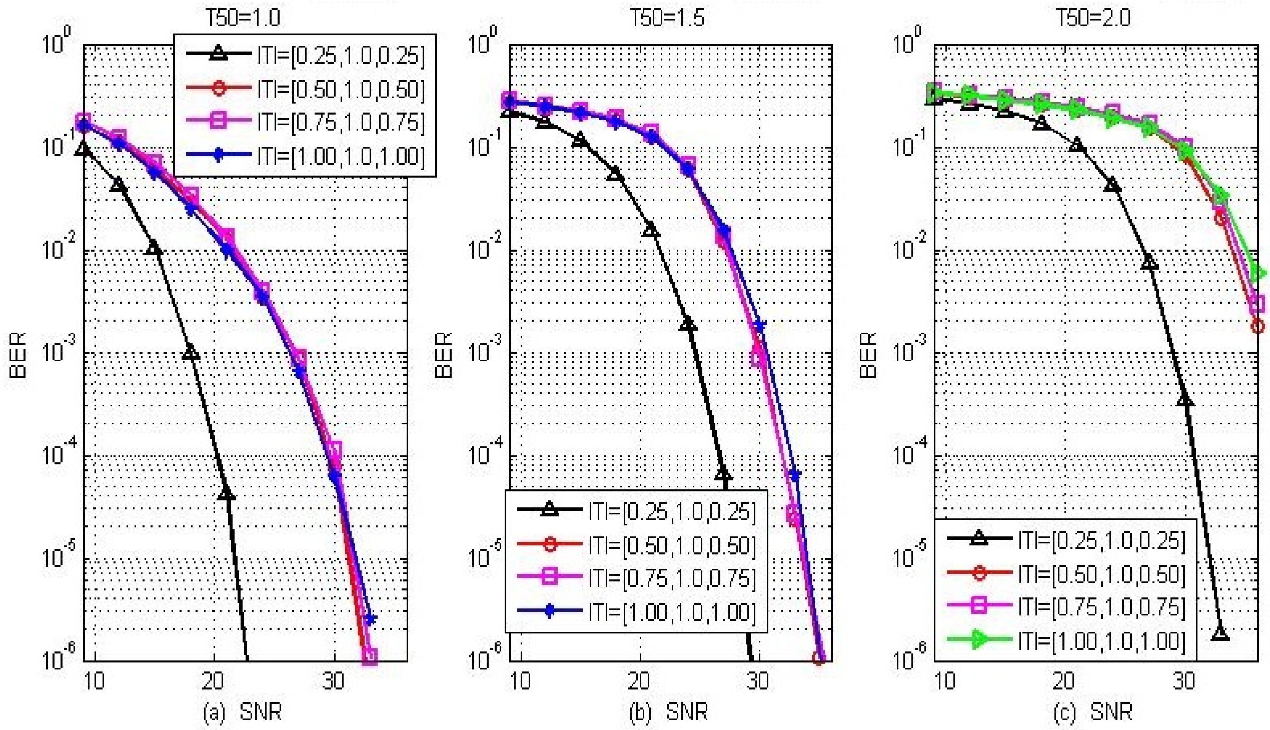


Figure 5.2: Performance of Full 2D Across track detector

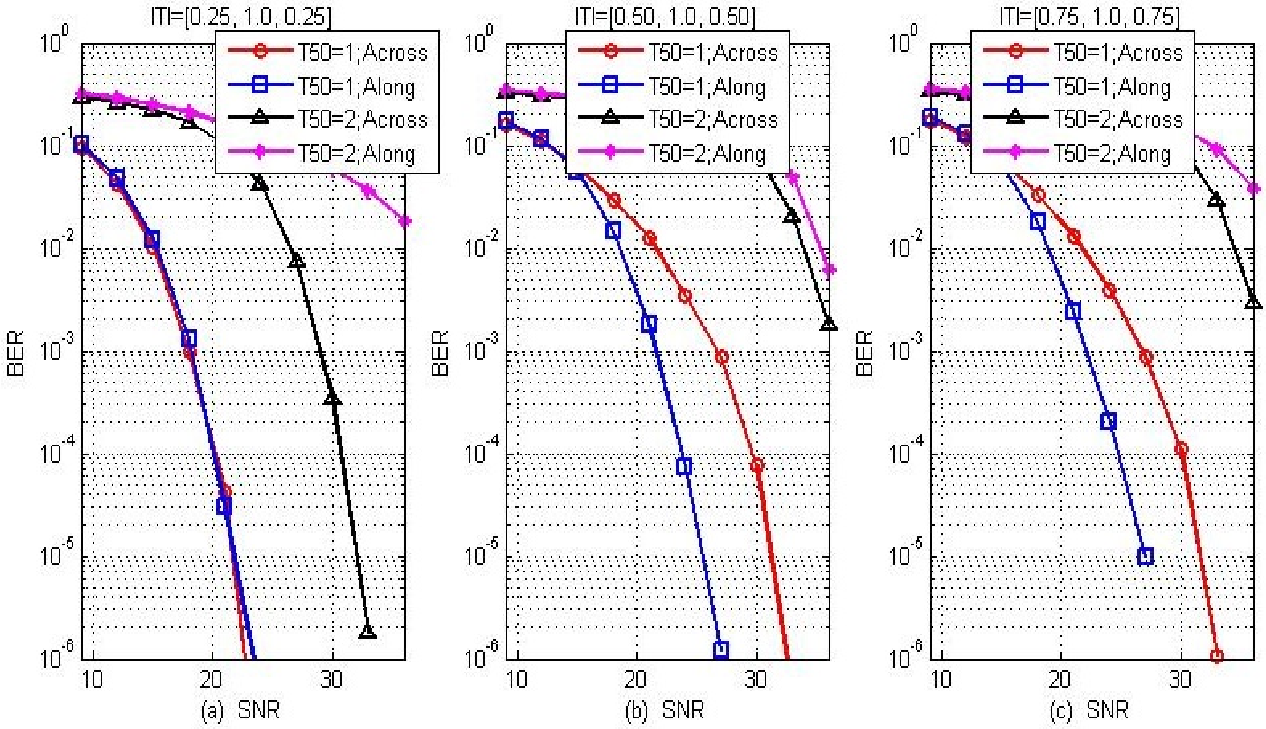


Figure 5.3: Comparison of Full 2D detectors

Results on this analysis are submitted to TELFOR journal in a paper titled, "Performance of 2D SOVA Along and Across Track in Shingled

Magnetic Recording Media" in November 2016, and a it is accepted for publication.

5.2 Comparison of 1D against 2D detectors

The aim of full 2D detection is to increase performance by eliminating excessive noise amplification in certain conditions of the detectors. A comparison of performance of the two 1D ML detectors implemented in this research, and the two 2D full ML detectors, is presented as follows.

Figure 5.4 shows graphs of 1D PRML along track, 1D ML across track, 2D SOVA along track and 2D SOVA across track. The data density along track is low ($T_{50} = 1.0$) for different ITI levels in the different graphs. The results show that for low ITI (Figure 5.4 (a)), the detectors have similar performance except for 1D PRML along track which is about 3dB worse than 1D across track. As the ITI increases, the performance of 2D SOVA across track falls off too, leaving 1D ML across track slightly better than 2D along. Therefore for $T_{50} = 1.0$, there is no benefit of using 2D detection, rather the performance reduces as against the best-performing 1D ML across track. 1D ML across track is about 7dB better than 2D SOVA across track.

When the T_{50} is increased to 1.50 (Figure 5.5), the 1D PRML along track and 2D SOVA across track that performed worst in the previous case now overtakes the remaining two in performance, with 1D PRML along track surprisingly having the best performance. 1D PRML along track is bout 2.5dB better that 2D SOVA across track, also about 6dB better than 1D ML across track. 2D SOVA Across track closes the gap as the ITI increases but still lagging a little behind 1D PRML along track. This means that it is equally not beneficial to use 2D detection when $T_{50} = 1.50$.

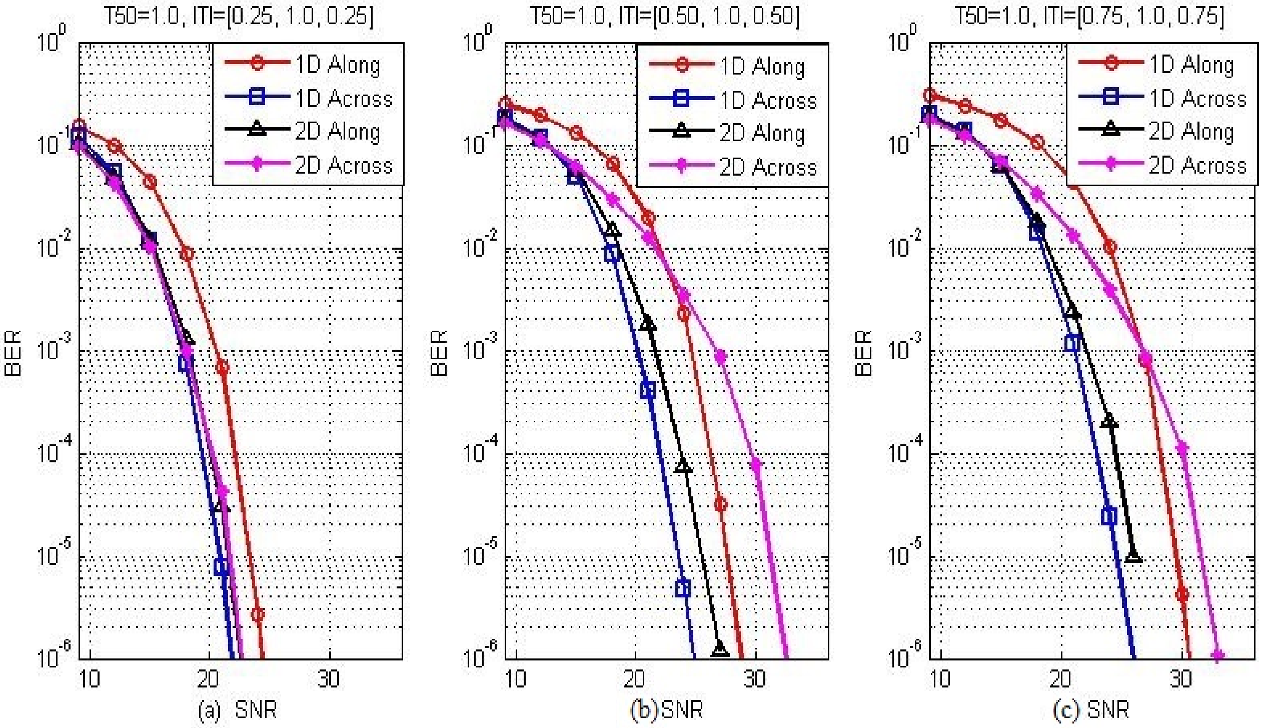


Figure 5.4: Comparison of 1D to Full 2D detectors at Low T_{50}

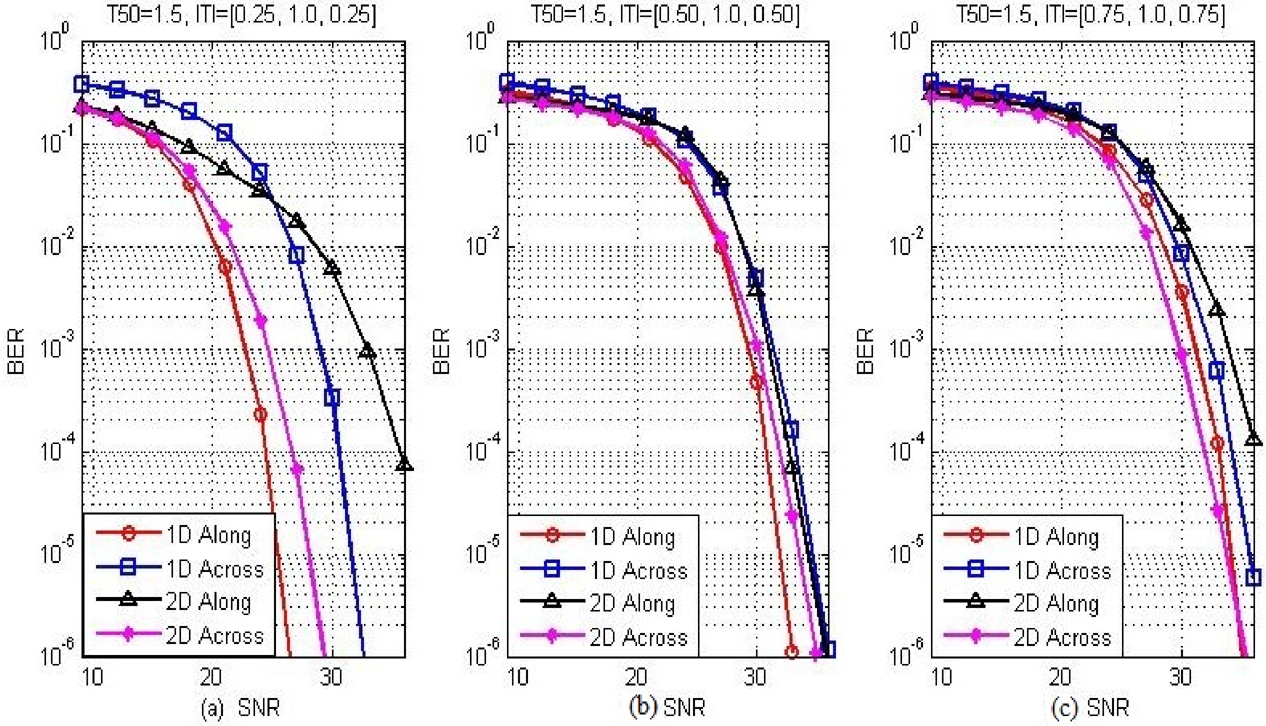


Figure 5.5: Comparison of 1D to Full 2D detectors at medium T_{50}

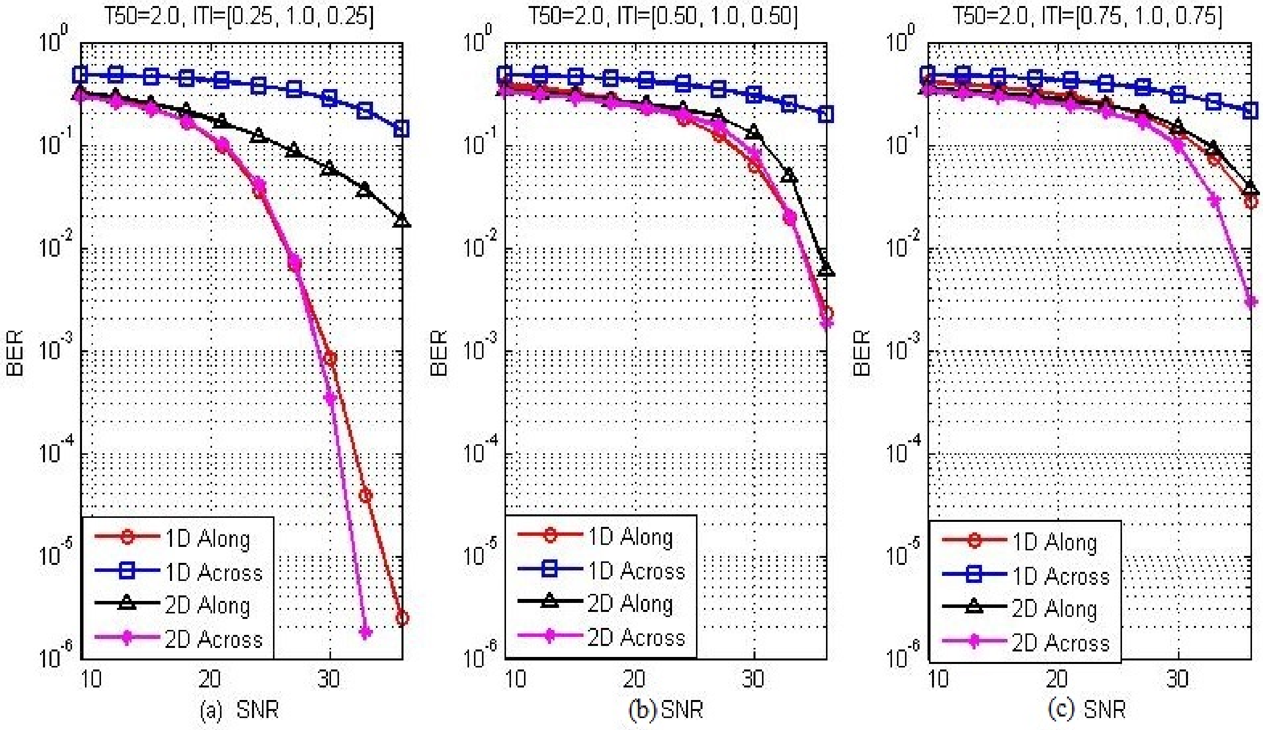


Figure 5.6: Comparison of 1D to Full 2D detectors at high T_{50}

Figure 5.6 shows graphs of the performance of the detectors for $T_{50} = 2.0$. In this situation 2D SOVA across track shows the best performance among all the detectors, with the difference in performance increasing at high ITI of [0.75 1.0 0.75]. 1D RPML along track still performs wonderfully good as the second best performing detector. 1D ML Across track has the worst performance of all for the conditions here.

These results show that an appreciable performance gain is achieved by using 2D detector (SOVA across) only for a situation where there is heavy ITI and heavy ISI at the same time. When ITI is low it is always sufficient to use 1D PRML along track as the detector, for the T_{50}s studied here. When ISI is low ($T_{50} = 1.0$) it is always best to use 1D ML across track as the detector.

5.3 Target Length of 4 Bits

In this scenario, we investigate two track ITI, with target length along track of 4. The read-head is assumed to capture both tracks involved equally (ITI=[1, 1]). This is to make sure that ML detection across track performs at its best. The target is symmetric, with the two middle values normalised to a value of 1.

Figure 5.7 (a), (b) and (c) shows the performance of different target for a SOVA based Full 2D ML detector using targets of length 4. It shows that target [0.2 1.0 1.0 0.2] ([1 5 5 1]) has the best performance among the tested targets for situations with low density along track (T_{50} =1.0 or 1.5). When the density along track is high (T_{50} = 2.0), target of [0.4 1.0 1.0 0.4] ([2 5 5 2]) exhibit the best performance.

Figure 5.8 displays the result of using BCJR in the 2D detector instead of SOVA. The results show similar performance to that of SOVA based detector where target [0.2 1.0 1.0 0.2] gives the best performance at T_{50} =1.0 and 1.5, while target [0.4 1.0 1.0 0.4] gives the best performance at T_{50} = 2.0.

From this point onward, the target of [0.2 1.0 1.0 0.2] will be adopted for T_{50} =1.0 and 1.5, while target [0.4 1.0 1.0 0.4] will be adopted for T_{50} = 2.0 in both the two types of detectors.

A comparison of the performance of SOVA based full 2D detector and BCJR based full 2D detector is shown, for various data densities, in figure 5.9. It shows that BCJR performs better than SOVA. This can be attributed to two factors. BCJR incorporate all probabilities of branches in its calculations, therefore, no information is discarded. Whereas SOVA, as implemented here, selects the best branch of each state and discard the remaining probabilities in further calculations.

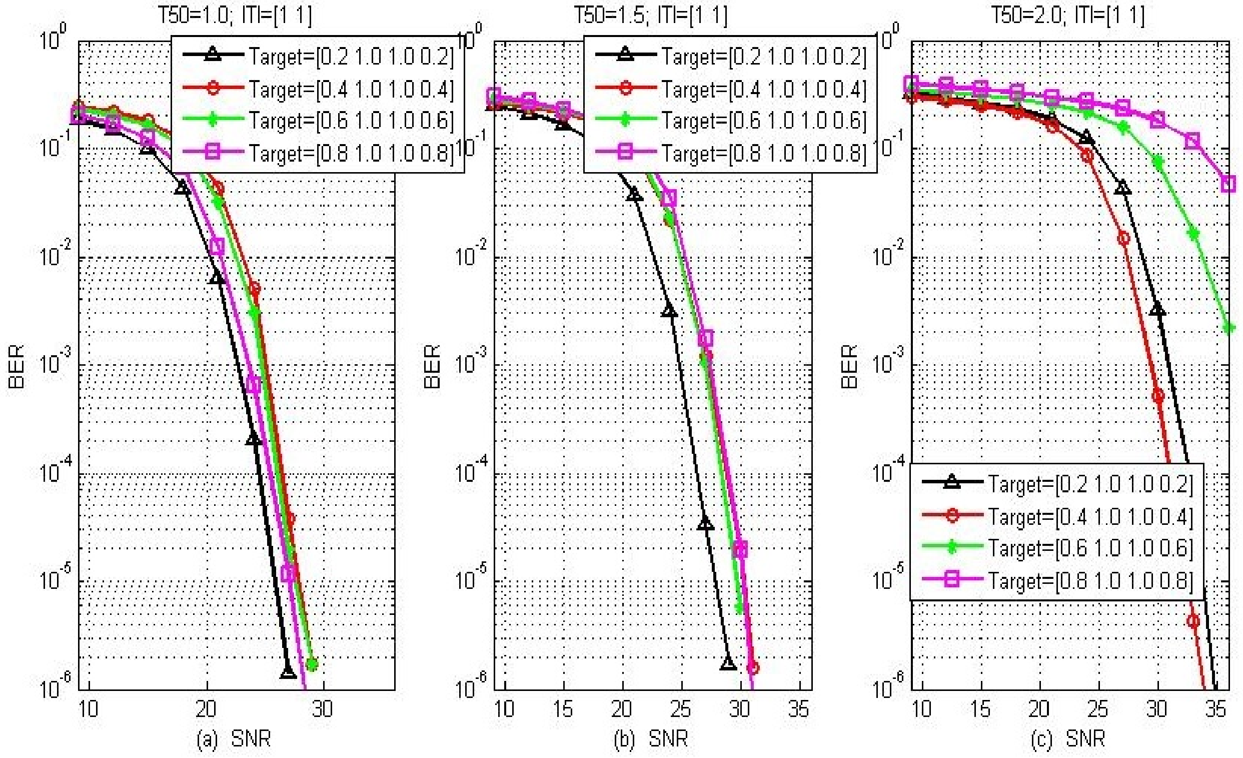


Figure 5.7: Performance of different Targets of length 4 using SOVA based ML detector

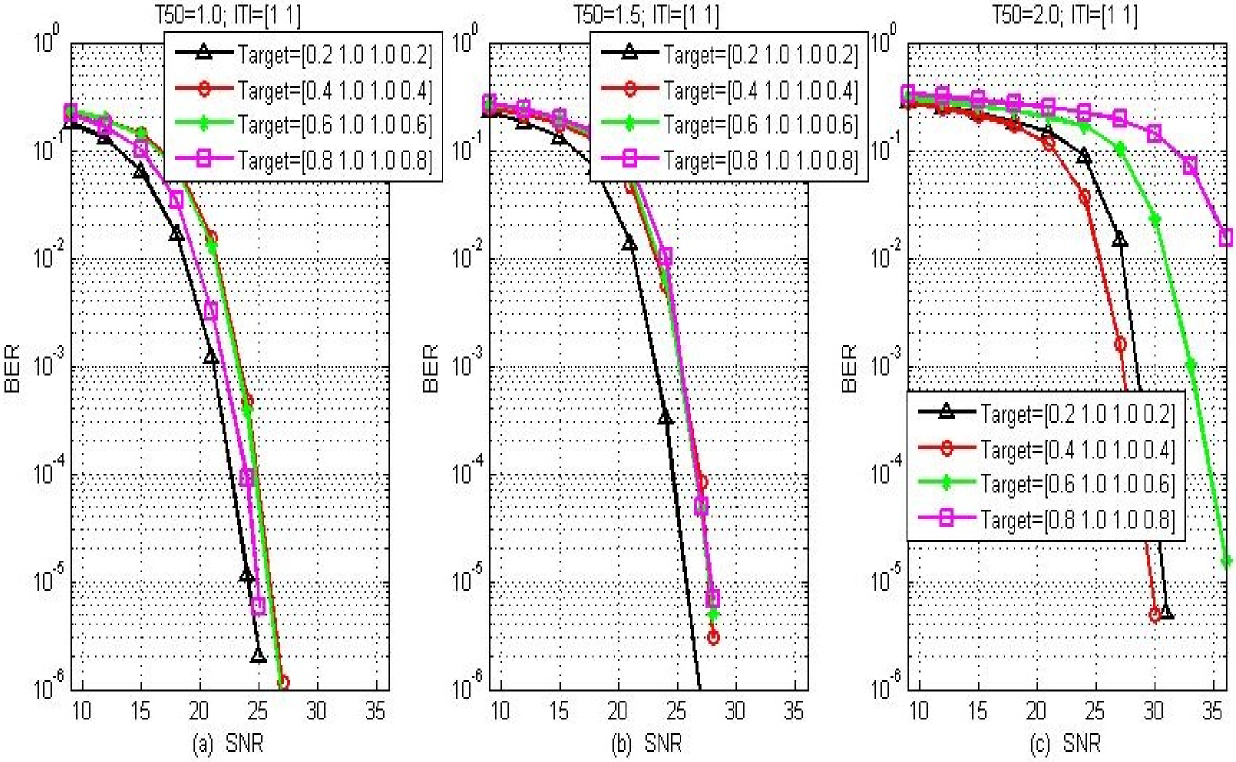


Figure 5.8: Performance of different Targets of length 4 using BCJR based ML detector

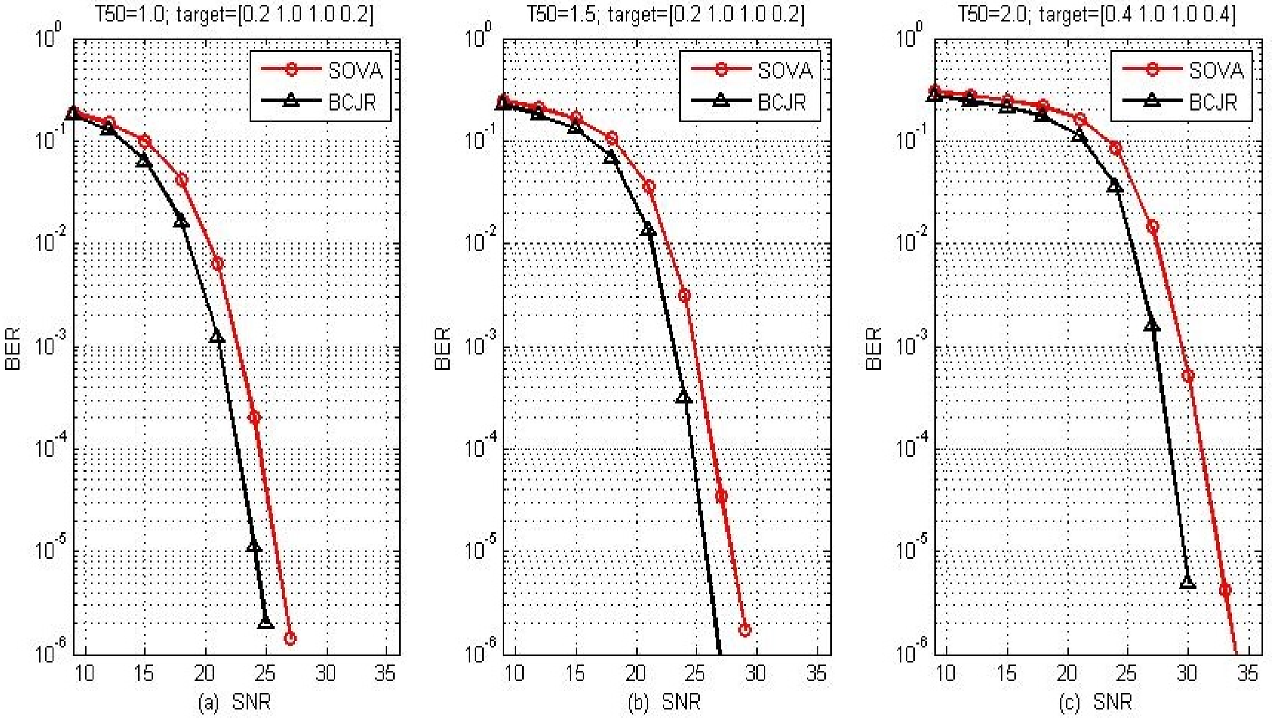


Figure 5.9: BER Performance comparison of SOVA and BCJR

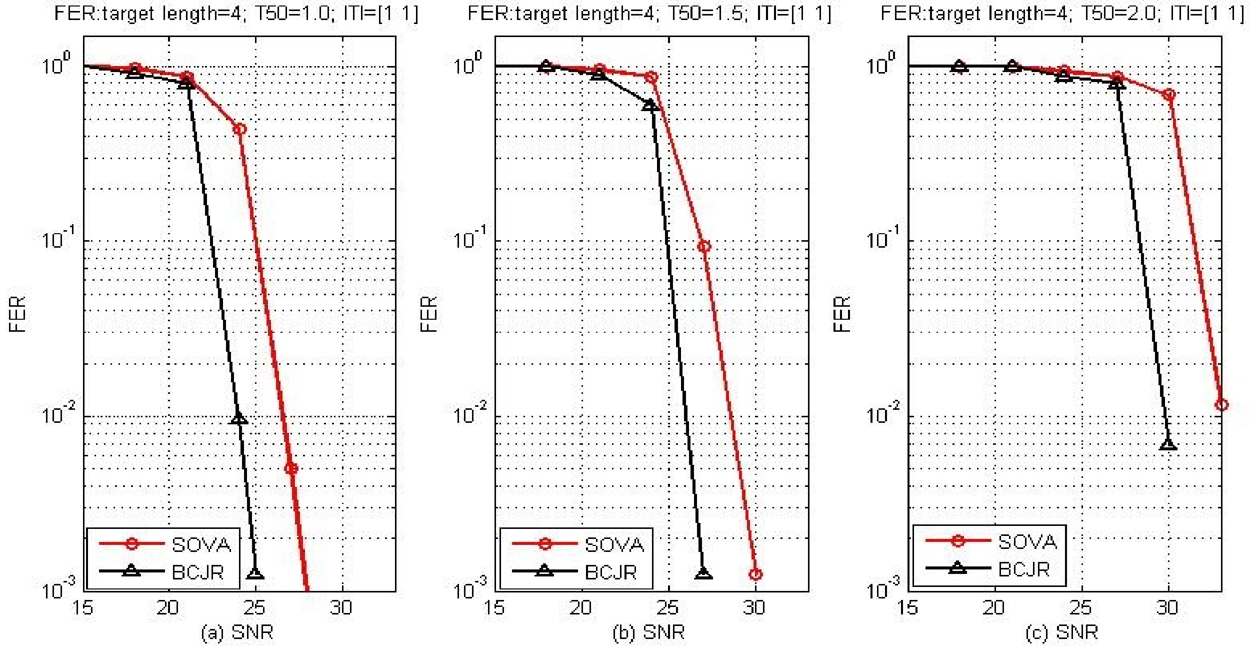


Figure 5.10: FER Performance comparison of SOVA and BCJR

The second factor is the fact that BCJR takes all bits, before and after the position of bit to be determined, in its calculations; whereas SOVA uses all symbols before the bit position to be determined and just a limited number of bits after it, given by trace-back length.

The gain as seen from the figures 5.9 (a), (b) and (c) is between 2 dB to 3 dB at BER of 10^{-5}. This gain is achieved at the expense of additional complexity. BCJR is about 3 times more computationally complex than SOVA because it needs separate calculations for forward recursion, backward recursion and A-Posteriori Probability (APP) determination. BCJR also requires more memory which is used to save both α, β and γ before APP is calculated. An extra latency is also incurred due to these extra processes.

Further comparison is made by comparing the frame error rates (FER) of the two types of detectors. Figure 5.10 show the FER of the SOVA based full 2D detector as compared to BCJR based full 2D detector. It shows a gain of about 3dB around the FER of 10^{-3}.

Part of these results are presented in a paper titled "Comparison of BCJR to SOVA in 2D detection of Shingled Magnetic Recording Media", and it is accepted to be presented in 40th International Conference on Telecommunications and Signal Processing (TSP) to take place between July 5-7, 2017.

5.4 Forward Error Correction

Two concatenated Single parity bit's system is used for forward error correction as described in section 3.4. This is applied on data of length 2304 per track which changes the number of bits per track to 4096 after all parity bits are added. This results into data rate of 9/16.

In our 2D implementation, 2D BCJR, constrained after every fourth symbol to account for the odd parity bit, is used for the initial detec-

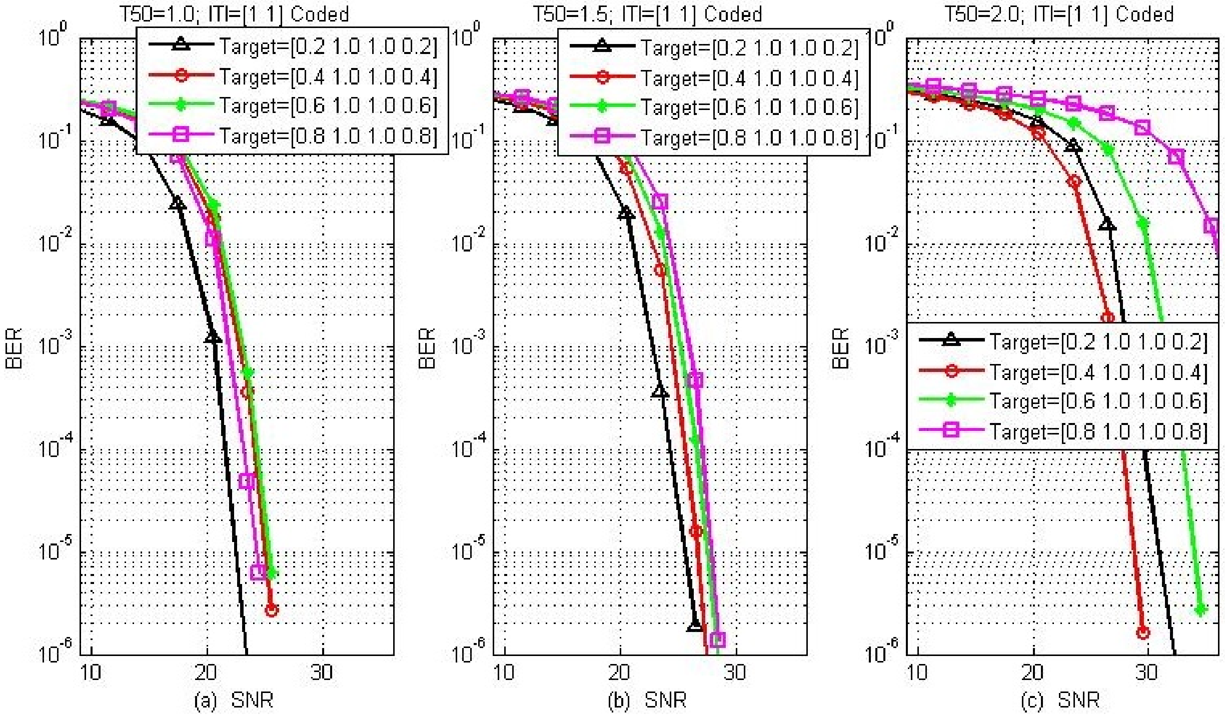


Figure 5.11: BER of Coded BCJR based 2D detector

tion along track. The parity symbols are then removed and the APP probabilities de-interleaved. The resulting APPs are passed through 1 parity bit 2D MAP decoder which produces results with ITI. Another MAP decoder is used across track to get the final decoded information. ITI of two track is still maintained and a target length of 4 bits.

Figures 5.11 show the performance of various targets in a coded system that used DRP interleaver to separate 2 single parity bits. It shows very similar property as compared to SOVA and BCJR based un-coded system. Therefore we are going to adopt the target selection of [0.2 1.0 1.0 0.2] for T_{50} =1.0 and 1.5, while target will be [0.4 1.0 1.0 0.4] for T_{50} = 2.0 as in the former cases.

A comparison of the performance of the coded BCJR and un-coded BCJR is shown in figure 5.12. The figures show only a marginal performance increase in terms of BER. The performance increase not significant especially when the additional complexity of the BCJR which

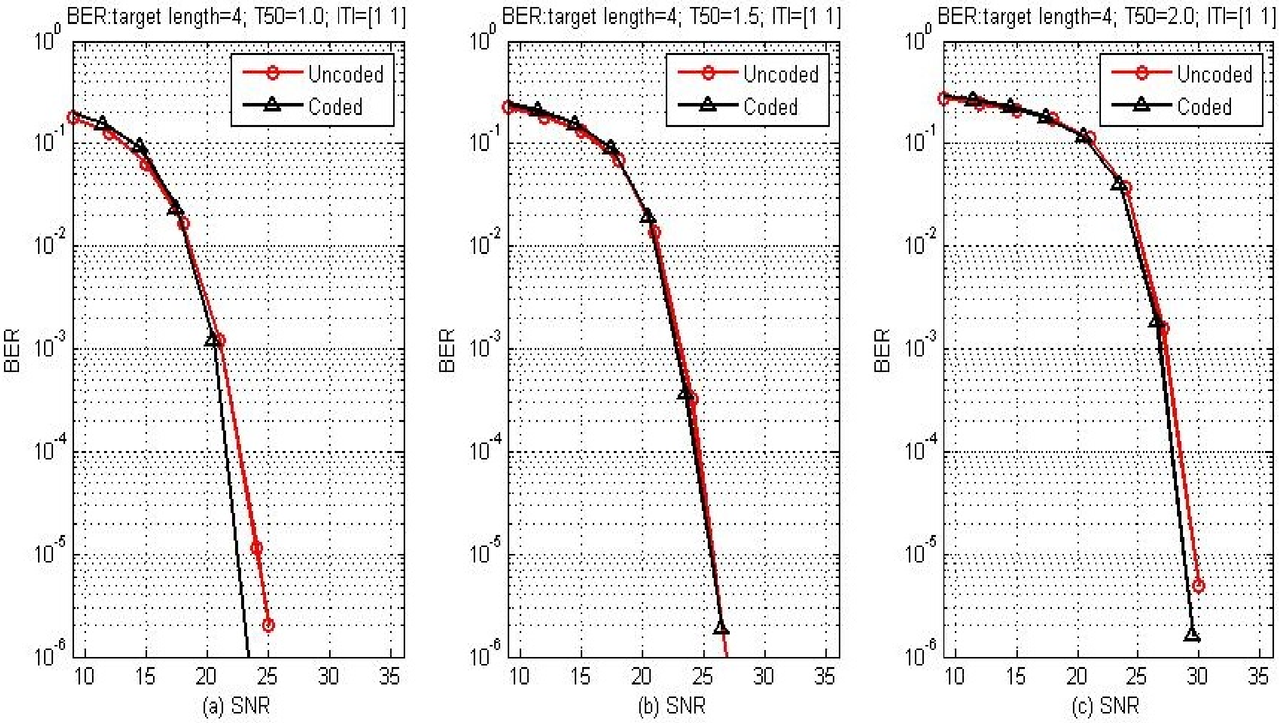


Figure 5.12: BER of Coded and Un-coded BCJR based 2D detector

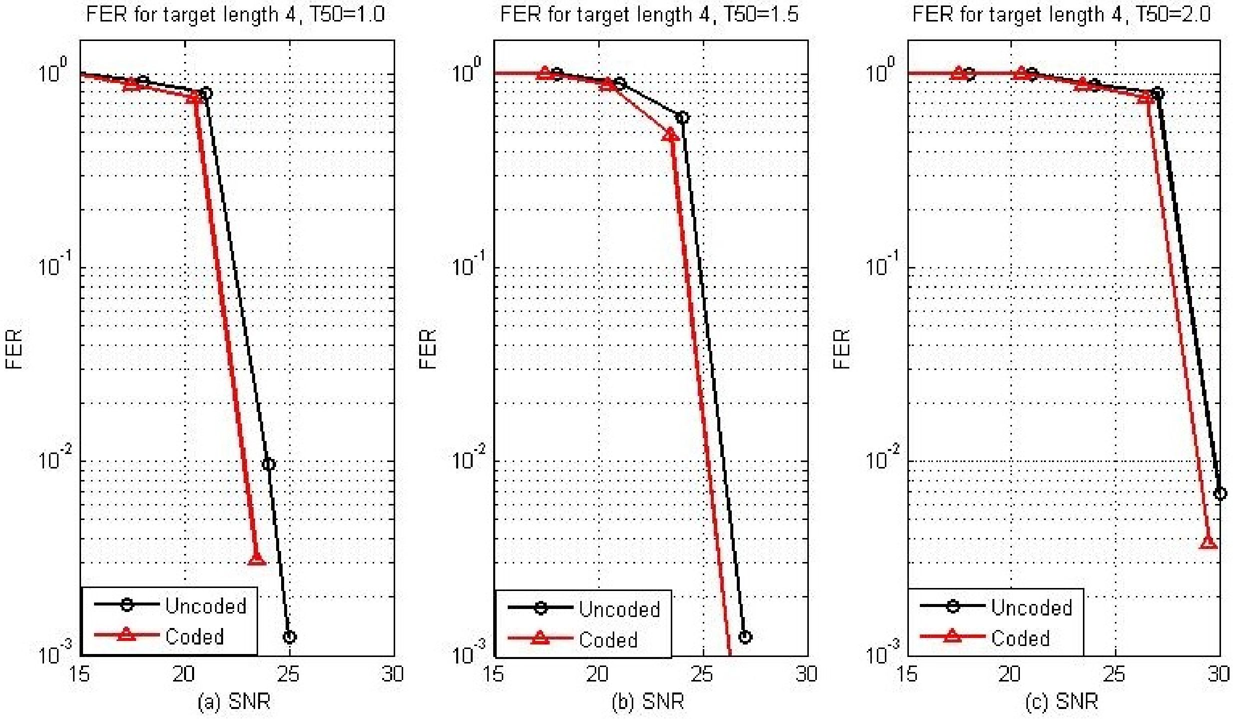


Figure 5.13: BER of Coded and Un-coded BCJR based 2D detector

now considers the parity constraint, the additional complexity due to MAP detector for the second parity bit and extra delay of processing the data is put into consideration.

Similar gains in the range of 1dB are also observed when the FER of the coded and un-coded 2D BCJR detectors are compared in figure 5.13.

The use of this coding procedure was pre-empted by another work, in which I participated. In the work, it was shown that, for a situation where there is no ITI and a 1D BCJR detector is used, a coding gain of up to 3dB can be achieved, using the same FEC system presented here. Results from the joint work are presented at "2016 Asia-Pacific Magnetic Recording Conference (APMRC)", with title "Decoding and Detection for Magnetic Recording Channel using Single Parity Coding". The conference took place at Yonsei University, Seoul South Korea, between 13th July to 15th July 2016.

5.5 Comparison to results from other literatures

Results of full 2D detection in an un-coded situation was presented in [57]. But the definition of SNR in that paper uses only AWGN variance ($\sigma_{noise} = \sigma_w$). In our definition, AWGN is just 20% of total noise. Therefore, we can transform their SNR by adding $10log_{10}(20/100)$ ($-7dB$) to it.

The specification in the paper is a 2D head response of

$$\begin{bmatrix} 0.1621 & 0.4026 & 0.1621 \\ 0.4026 & 1.0000 & 0.4026 \\ 0.1621 & 0.4026 & 0.1621 \end{bmatrix}$$

This is equivalent T_{50} of just less than 1.0 and ITI of [0.4026, 1.0, 0.4026]. For easy comparison, we will compare it to a SOVA situation with $T_{50} = 1.0$ and ITI of [0.5, 1.0, 0.5]. This is presented in figure 5.1 (a). The results presented in figure 4 of [57] achieved BER of 10^{-4} at 30dB. This is the same to BER at 23dB (30dB-7dB) of figure 5.1 (a) despite over-estimation of ITI and ISI in our case. This is also despite the fact that the paper we are comparing to, assumed spread of ISI is just three symbols for simplicity. When more rigorous error correction is applied to the detected signal in future research, the SNR at which acceptable BER is achieved can be determined and, therefore, achievable data density can be estimated after that.

Chapter 6

SUMMARY, RECOMMENDATIONS AND CONCLUSION

This chapter presents summary of the work done, summary of results obtained, contributions made by the work, conclusions derived from the work, and possible recommendations for future work.

6.1 Summary of Work done

The task of increasing the areal density of magnetic recording medium is a task in the hands of researchers and computing industry players. This is to satisfy the ever-increasing need for more storage capacity of computing devices. This task is facing a serious challenge of superparamagnetic limit. That is the capacity limit of the current magnetic medium technology.

Techniques for advancing the capacity beyond $1Tb/in^2$ were suggested and among them are, Bit Patterned Magnetic Recording medium, Energy Assisted Magnetic Recording medium, and Shingled Magnetic Recording (SMR) medium.

SMR is a very attractive proposition because it does not present any drastic change to the current HDD technology. It suggests using a slightly different write-head to write bits in overlapping shingles. This means parts of the previously written bits will be overwritten when neighbouring bits are written after them. This brings the bits very close together to the extent that the read-head has difficulty of reading the data without interference from neighbouring bits. Interference comes from both directions along-track and across-track.

This problem of Two-Dimensional (2D) interference necessitates the use of 2D techniques in coding, equalisations, detection and decoding, to achieve the potentials of SMR. 2D Magnetic Recording techniques are presented by researchers, which include the use of 2D equalisers, Partial Response Maximum Likelihood (PRML) detection, and full 2D joint track detections. The good performing ML alternatives are normally very complex. Getting a simpler less complex technique, or simplifying the existing ones, is a very important step in actualising the detection techniques. This is why this research is based on "Reducing the Complexity of Equalisation and Decoding of Shingled Writing".

So far the work conducted includes the creation of PMR channel model, with possible ITI of three tracks (Interference from both sides of the central track) and a target length of 3 bits during equalisation. The channel is also implemented with two track ITI (interference from one side track) with a target length of 4 bits during equalisation. Both of these channel models were implemented in software using MATLAB code, and an FPGA model for the situation of 3 track ITI and target of 3 bits.

MMSE equaliser/detector, which can be reduced to ZF was implemented. This is used to either cancel ITI or ISI in 1D PRML detectors or cancel both as in the ZF and MMSE detectors.

1D PRML detector along track, which uses linear equaliser along track to shape data, linear equaliser across track to cancel ITI, then

PRML along track to detect signal, was designed. The use of a linear equaliser to cancel ITI is to reduce the problem of the detection to a 1D problem. This helped in reducing the complexity of the ML detection greatly.

1D PRML detector across track was also made, which uses a linear equaliser to cancel ISI along track, and an ML detector across track to detect signal. This is to reduce the complexity of 1D PRML along-track and improve performance. These are achieved by using less linear equalisers. It has better strength across track.

Full 2D detectors based on 2D SOVA were also implemented. The detectors were implemented in two versions. After target shaping, one of the detectors uses 2D SOVA along track to cancel ISI, and then uses Viterbi Algorithm (VA) across track to cancel ITI. The other version uses 2D SOVA across track to cancel ITI, and VA along track to cancel ISI. These implementations are based on three track ITI channel and target of length 3 (3 by 3). The concatenation of the SOVA and VA is to reduce the complexity of using single 2D ML detection which is very high for large number of tracks.

Detectors are also designed for two-track channel interference with a target length of four bits (2 by 4). One of the detectors was based on 2D SOVA along track and VA across track. Another implementation was based on 2D BCJR along track with MAP detection across track. This is to investigate advantage of longer targets in the improvement of performance. It also gives less complex detection compared to “3 by 3” system discussed in the paragraph above.

Single parity encoders and interleavers were developed. They are used to apply two single parity bit systems, to the data. The two single-parity systems are separated by DRP interleaver. DRP interleaver is used to make sure the symbols are randomly spread as far as possible to their neighbouring bits after the first detection. This will randomise the noise when the second parity is to be detected. The single-parity system is aimed at simplifying the detection while concatenating the two is aimed at getting good performance.

A detector was also developed to detect data with the two concatenated parity bit systems separated by DRP interleaver. Constrained 2D BCJR was used along track to detect and remove last applied parity bits. The resulting data was de-interleaved, then 2D MAP decoder used on the inner parity bits along track, and finally another MAP detector is used across track to cancel ITI. The procedure provided a simple simultaneous detection/decoding procedure with the aim of getting improvement in performance.

6.2 Summary of Results

This section summarises the major results found or observed during the course of this research. The results are itemised here, with references to where they are presented.

- For a channel with low ITI and low ISI, it is adequate to use linear equalisers, such as ZF and MMSE, to detect the information (Figure 4.18 (a)).
- When there is high ITI and low ISI, using ZF equaliser to cancel ISI and 1D ML detector across track gives the best performance. A gain of about 5dB, by 1D ML across-track over 1D PRML along-track, and about 9dB over ZF and MMSE was recorded at BER of 10^{-6} (figure 4.18 (c)).

- In situations where there is low ITI but high ISI, a detector that uses a linear equaliser to cancel ITI and 1D PRML detector along track to cancel ISI is adequate to get a good result. A gain of more than 10dB is recorded by 1D PRML along-track over ZF, MMSE, and 1D ML across track at BER of 10^{-6} (Figure 4.19).

- The read head must be skewed from centre of the main track, when total ITI is about 1.0, in order to have a good result in ZF, MMSE and 1D PRML along track detector (Figures 4.8, 4.9).

- When data density is the main concern, with less concern on the speed of data access, 1D ML across-track detector is far better than 1D PRML along-track. The data is better compressed across track, relaxed along track, and 1D ML across track used. This performs better than compressing along track, relaxing across track and using 1D PRML along-track. A gain of about 10dB was recorded at BER of 10^{-6} (Figure 4.20).

- 1D PRML across track also has the advantage of less complexity. Though both 1D PRML detectors have exponential complexities, fewer computations and comparisons are needed for 1D ML across track, because only 1 equaliser is needed and fewer comparisons are done (Table 4.1). The results above are the basis of the publication of the paper titled, "Two Dimensional Equalisation of Shingled Magnetic Recording Media" in International Conference on Magnetism (ICM) 2015 in Barcelona.

- Concatenation of 2D SOVA along track with VA across track, and or 2D SOVA across track and VA along track, is presented, and they reduce the complexity of joint-track detection of 8 tracks from order of 2^{8*3} (16,777,216) to order of $8*2^{3*2}$ (1,024) (Section 3.3.3). This is presented in a paper titled, "Concatenated 2D SOVA for Two Dimensional Maximum Likelihood Detection" at TELFOR 2015 conference at Belgrade Serbia.

- ITI of [0.5, 1.0, 0.5] has better performance than other ITIs at high SNR and ISI, except for ITI[1.0, 1.0, 1.0] in 2D SOVA along track. But ITI[1.0, 1.0, 1.0] is not normally realisable without extra interference from other tracks (Figure 5.1).

- For 2D SOVA across track, less ITI is better. ITI[0.25, 1.0, 0.25] recorded more than 6dB gain over other ITIs at BER of 10^{-6} (Figure 5.2).

- 2D SOVA across track followed by VA along track performs better than using 2D SOVA along track followed by VA across track, at high data density along track. At low data density along track, 2D SOVA along track performs better with a gain of up to 6dB (Figure 5.3). This is accepted for publication in TELFOR Journal with the title, 'Performance of 2D SOVA Along and Across Track in Shingled Magnetic Recording Media".

- When 1D and 2D detectors are compared, it is found that it is always better to use 1D ML across-track for a situation with very low ISI (Figure 5.4).

- For medium and high ISI with low ITI, it is adequate to use 1D PRML along track over other 2D detectors (Figure 5.5 and figure 5.6 (a) & (b)).

- It is better to use 2D SOVA across track, only for simultaneously high ITI and ISI (Figure 5.6 (c)).

- For implementations with two track interference and target of length 4, suitable targets for the detection are found to be [0.2, 1.0, 1.0, 0.2] for $T_{50} \leq 1.5$, while for larger values of T_{50} target of [0.4, 1.0, 1.0, 0.4] performs better (Figures 5.7 and 5.8).

- 2D BCJR along track followed by MAP detector across track performs better than 2D SOVA along track followed by VA across track, with a gain of 2dB to 3dB at BER of 10^{-5}, with a considerable increase in complexity (Figures 5.9 and 5.10). A paper titled, “Comparison of BCJR to SOVA in 2D detection of Shingled Magnetic Recording Media”, is submitted for presentation to 40th International Conference on Telecommunications and Signal Processing (TSP).

- There is just small gain by using two concatenated single parity systems separated by a DRP interleaver, over an uncoded system. This suggests that it is not useful to do the coding (Figures 5.12 and 5.13).

6.3 Major Contributions

From the work done through this research to the results found and presented in this report, some contributions to knowledge and the area of research are made. The following list presents such contributions to knowledge by this work.

1. The merits of combining linear equalisation and ML detection for TDMR in shingled magnetic disk is highlighted. It is shown that for a large number of situations a much simpler 1D detector can be used to get better BER performance as compared to the more complex 2D detector.
2. It is shown that depending on the direction with high relative data density, the ML procedure can be applied either along track or across track.
3. It is found that, in order to achieve high data density with relatively low complexity using 1D detector, it is better to use a linear equaliser for cancelling ISI and use ML detector for cancelling ITI, through relaxing the data density along the track and increasing it across the track.

4. 1D ML across track has less computational complexity than 1D PRML along track.

5. 2D ML detector performance is better understood in its relation with 1D detectors. The conditions to have reasonable gain are found to be a situation where there is high ITI and ISI at the same time.

6. Simple FEC by two concatenated single parity systems can give a reasonable gain in situations with no ITI or low ITI but has negligible benefit when ITI is present.

7. Relative performance of BCJR/MAP and SOVA/VA full 2D detectors is presented. The extent of gain which BCJR/MAP offer over SOVA/VA is investigated. The extent of comparative complexity differences is also presented.

6.4 Recommendations for Future Work

In order to continue this research, to make sure the performances of the detectors are improved even more, the following recommendations can be pursued for future work:

1. In the VA, SOVA, MAP and BCJR detectors used in this research, it is assumed that there is uniform amount of noise in all symbols on the average. But since the noise is dominated by jitter, and jitter only occurs at transitions, then symbols like 000 will have no jitter in their principal junctions while symbols like 101 will have jitters in all their principal junctions. This means in a future work, this information can be included in determining the reference values, in SOVA and VA or the σ^2 in BCJR, which are suitable for each trellis branch. This may improve performances.

2. Because we now understand that combining linear equaliser with 1D PRML along track can perform very well in a lot of situations,

it will be good to investigate longer targets along track for this scheme. This may perform better than expected.

3. Further research on simple error corrections are advised. This may include using one parity bit system along track and the other across track, as against using them all along track separated by DRP as done in this research. This will reduce the complexity due to the task of interleaving and de-interleaving and potentially increase the strength of the detector across track.

4. In an attempt to improve the performance, iterative decoding can be investigated on the FEC implementation presented in this report, or the one advised in the item above. This will come in the form of passing information between the detector along track and across track. It means both detectors will, therefore, be 2D soft output detectors.

5. LDPC coding, normally used in FEC of SMR medium DSP, can also be implemented such that the detection and decoding are done at the same time without having to do one after the other. This may reduce complexity, reduce delay or reduce errors due to the passing of detection errors to decoders.

6.5 Conclusion

In conclusion, the overall findings of the study carried out is that: interchanging direction in which equalisers and detectors are applied can produce better performance in some situations; complex full 2D detectors are only needed in systems with very high ITI and ISI at the same time in order to get good performance; for low ITI situations, using ML across track is the best option and gives advantages of lower complexity; for low ISI using PRML across track is more beneficial; when both ITI and ISI are low it is adequate to use simple MMSE equalisers; the concatenation of SOVA/VA or BCJR/MAP greatly reduces complexity of full 2D joint track detection.

It is also shown that there is not much benefit in using simple single parity decoders as an error correction technique when ITI is present.

Bibliography

[1] "History of Hard Drives," *http : //en.wikipedia.org/wiki/History_of_hard_disk_drives*, (Accessed 19th May 2016).

[2] Y. Shiroishi, K. Fukuda, I. Tagawa, S. Takenoiri, H. Tanaka, and N. Yoshikawa, "Future options for HDD storage," *IEEE Transactions on Magnetics*, vol. 45, no. 10, October, 2009.

[3] P. Shah, P. Davey, M. Z. Ahmed, and M. A. Ambroze, "Multilevel Coding for Partial Response Channels," *DSNET MEETING, Manchester*, July 2005.

[4] "Disk Sector," $https : //en.wikipedia.org/wiki/Disk_sector$, (Accessed 15th April 2017).

[5] A. S. Hoagland and J. E. Monson, "Digital Magnetic Recording," Second edition, *John Wiley & Sons, Inc*, New York, USA, 1991.

[6] Toshiba Press Release, "Toshiba Leads Industry in Bringing Perpendicular Data Recording to HDD–Sets New Record for Storage Capacity With Two New HDDs," *http : //www.toshiba.co.jp/about/press/*2004_12/*pr*1401.*htm* (14th December 2004) Accessed 09 July, 2016.

[7] R. F. Harrington, "Introduction to Electromagnetic Engineering," Mineola, NY: Dover Publications, pp. 49. ISBN 0-486-43241-6, 2003.

[8] A. Taratorin, "Characterisation of Magnetic Recording Systems," *Guzik Technical Enterprises*, 1996.

[9] P. H. Siegel, “Recording codes for digital magnetic storage,” Technical Report 4720 (50198), *IBM Research Laboratory*, Almaden Research Center, USA, May 1985.

[10] H. Kobayashi, and D.T. Tang, “Application of partial-response channel coding to magnetic recording systems,” *IBM Journal on Research and Development*, vol. 14, pp. 368-375, November, 1970.

[11] C. K. Sann, R. Radhakrishnan, K. Eason, R. M.y Elidrissi, J. Miles, B. Vasic, A. R. Krishnan, “Channel Models and Detectors for Two-Dimensional Magnetic Recording (TDMR),” *IEEE Transaction on Magnetics*, vol. 46, no. 3, March, 2010.

[12] R. Wood, “The Feasibility of Magnetic Recording at 10 Terabits Per Square Inch on Conventional Media,” *IEEE Transactions on Magnetics*, vol. 45, no. 2, February, 2009.

[13] T. Wu , M. A. Armand , and J. R. Cruz, “Detection-Decoding on BPMR Channels With Written-In Error Correction and ITI Mitigation,” *IEEE Transactions on Magnetics*, vol. 50, no. 1, January, 2014.

[14] W. Chang and J. R. Cruz, “Inter-Track Interference Mitigation for Bit-Patterned Magnetic Recording,” *IEEE Transactions on Magnetics*, vol. 46, no. 11, November, 2010.

[15] Zhu Jian-Gang, et al. “Microwave Assisted Magnetic Recording,” *IEEE Transactions on Magnetics*, vol.44, no. 1, p.125-131, January, 2008.

[16] Y. Inaba, H. Nakata, D. Inoue, “New High Density Recording Technology: Energy Assisted Recording Media,” *FUJI Electric Review*, vol. 57 no. 2, 2011.

[17] M. H. Kryder, E. C. Gage, T. W. McDaniel, W. A. Challener, R. E. Rottmayer, G. Ju, Y-T. Hsia ; M. F. Erden, “Heat Assisted

Magnetic Recording," *Proceedings of the IEEE*, vol. 96, no. 11, pp. 1810 - 1835, November, 2008.

[18] AnandTech, "Breaking Capacity Barriers With Seagate Shingled Magnetic Recording," $http://www.seagate.com/em/en/tech-insights/breaking-areal-density-barriers-with-seagate-smr-master-ti/$, Accessed 20 May, 2016.

[19] S. Anthony, "HGST releases helium-filled 10TB hard drive; Seagate twiddles shingled finger," $http://arstechnica.co.uk/information-technology/2015/12/hgst-releases-helium-filled-10tb-hard-drive-seagate-twiddles-shingled-fingers$, Acessed 20 May, 2016.

[20] G. Gasior,"Shingled Platters Breathe Helium Inside HGST 10tb Hard Drive," $http://techreport.com/news/27031/shingled-platters-breathe-helium-inside-hgst-10tb-hard-drive$, The Tech Report (September 9, 2014), Accessed 19th May, 2015.

[21] K. A. s. Immink, "Run-length limited sequence," *Proceedings of the IEEE*, vol. 78, no. 11, November, 1990.

[22] I. McLoughlin, "Computer Pheripherals", *McGraw Hill*, 2011.

[23] R. S. LIM, "A (31,15) Reed-Solomon code for large memory system," *NASA-Ames Research Center*, Moffett Field, California, 1979.

[24] Y. Han, "LDPC coding for magnetic storage: low-floor decoding algorithm, system design, and performance analysis," PhD dissertation, *Department of Electrical and computer engineering*, University of Arizona, 2008.

[25] H.J. Richter, "Density limits imposed by the microstructure of magnetic recording media," *Journal of Magnetism and Magnetic Materials*, vol. 321, no. 6, pp. 467-476, March 2009.

[26] K. S. Chan, J. Miles, E. Hwang, B. V. K. Vijayakumar, J. G. Zhu, W. C. Lin, and R. Negi, "TDMR platform simulations and experiments," *IEEE Transactions on Magnetics*, vol.45, no. 10, October, 2009.

[27] R. M. Todd , E. Jiang , R. L. Galbraith , J.R.Cruz , and Roger W. Wood, "Two-Dimensional Voronoi-Based Model and Detection for Shingled Magnetic Recording," *IEEE Transactions on Magnetics*, vol. 48, no. 11, November, 2012.

[28] A. R. Krishnan, R. R. Redhakrishnan, B. Vasic, A. Kavcic, W. Ryan and F. Erden, "2-D Magnetic Recording: Read Channel Modeling and Detection," *IEEE Transaction on Magnetics*, vol. 45, no. 10, October, 2009.

[29] A. Fahrner, "On Signal Processing and Coding for Digital Magnetic Recording System," *Ph.D. dissertation*, Department of Telecommunications and Applied Information Theory, University of Ulm, Germany, 2004.

[30] D. T. Wilton, D. McA. McKirdy, H. A. Shute, J. J. Miles, and D. J. Mapps, "Approximate Three-Dimensional Head Fields for Perpendicular Magnetic Recording," *IEEE Transaction on Magnetics*, vol. 40, no. 1, January, 2004.

[31] P. Chaichanavong and P. H. Siegel, "Tensor-product parity code for magnetic recording," *IEEE Transaction on Magnetics*, vol. 42, no. 2, pp. 350-352, February, 2006.

[32] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Transactions on Information Theory*, vol. 2. pp. 284-287, March, 1974.

[33] F. J. MacWilliams and N. J. A. Sloane, “The Theory of Error-Correcting Codes,” Amsterdam, New York : North-Holland Pub. Co., 1996.

[34] C. Schlegel and L. Perez, “Trellis and Turbo Coding,” *Wiley-IEEE Press*, 2003, ch. 7.

[35] R. W. Wood and D. T. Wilton, “Readback responses in three dimensions for multilayered recording media configurations,” *IEEE Transaction on Magnetics*, vol. 44, no. 7, pp. 1874-1890, July, 2008.

[36] Forensicswiki, “Magnetoresistive (MR) Head Technology,” *A Quantum White Paper*, available at http //www.forensicswiki.org/w/images/0/06/Maxtor˙Magnetoresistive accessed in April 2017.

[37] Z. Wu, “Channel Modeling, Signal Processing and Coding for Perpendicular Magnetic Recording,” *PhD dissertation*, Electrical Engineering (Communication Theory and Systems), University of California, San Diego, 2009.

[38] T. Losuwan, C. Warisarn, and P. Kovintavewat, “A Study of 2D detection for Two-Dimensional Magnetic Recording,” *https : //www.researchgate.net/publication/*259475920_*A_Study_of_2D_de* *Dimensional_Magnetic_Recording*, July, 2012, Accessed July, 2016.

[39] C. Du, J. Zhang, E. H. Ong, “Timing Jitter Modeling and Minimization for a Servo Track Writer,” *IEEE Transactions on Magnetics*, vol. 43, no. 9, pp. 3769-3773, September 2007.

[40] G. Gibson, G. Ganger, “Principles of Operation for Shingled Disk Devices,” *Parallel Data Laboratory*,Carnegie Mellon University, Pittsburgh. April, 2011.

[41] G. Gibson, M. Polte. “Directions for Shingled Write and Two Dimensional Magnetic Recording System Architectures: Synergies with Solid-State Disks”, *Parallel Data Laboratory*, Carnegie Mellon University. January, 2009.

[42] A. Amer, JoA. Holliday, D. D.E. Long, E. L. Miller, J-F. Paris, T. S. J. Schwarz, “Data Management and Layout for Shingled Magnetic Recording,” *IEEE Transactions on Magnetics*, vol. 47, no. 10, October, 2011.

[43] C-I. Lin, D. Park, W. He, and D. H. C. Du. “H-SWD: Incorporating Hot Data Identification into Shingled Write Disks,” *IEEE 20th International Symposium on: Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 7-9 August, 2012.

[44] E. Hwang, R. Negi, and B. V. K. Vijayakumar, “Signal Processing for Near 10 Tbit/in2 Density in Two-Dimensional Magnetic Recording (TDMR),” *IEEE Transactions on Magnetics*, vol. 46, no. 6, June, 2010.

[45] P. Shah, M. Z. Ahmed, and Y. Kurihara. “New Method for Generalised PR Target Design for Perpendicular Magnetic Recording,” *The Eighth Perpendicular Magnetic Recording Conference, PMRC 2007*, October 15 - 17 2007.

[46] S. Gopalaswamy, B.V.K. Vijaya-Kumar, “Multi-channel equalization for PWM channels in magneto-optical recording,” *IEEE Transactions on Magnetics*, vol. 33, no. 5, September, 1997.

[47] B. M. Kurkoski, “Towards Efficient Detection of Two-Dimensional Inter-Symbol Interference channels,” *IEICE Transactions, Fundamentals*, vol. E91-A, no. 10, October, 2008.

[48] J. D. Coker, E. Eleftheriou, R. L. Galbraith, and W. Hirt. "Noise-Predictive Maximum Likelihood (NPML) Detection," *IEEE Transactions on Magnetics*, vol. 34, no. 1, January, 1998.

[49] N. Awad, "On Reducing the Decoding Complexity of Shingled Magnetic Recording System," *PhD dissertation*, University of Plymouth, May 2013.

[50] T. P. Krauss, M. D. Zoltowski, and G. Leus, "Simple MMSE Equalisers for CDMA Downlink to Restore Chip Sequence: Comparison to Zero-Forcing and Rake," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 3965-2868, 2000.

[51] A. Klein, G.K. Kaleh, P.W. Baier, "Zero forcing and minimum mean-square-error equalization for multiuser detection in code-division multiple-access channels," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 2, May, 1996.

[52] B. M. Hochwald and S. Ten Brink, "Achieving Near-Capacity on a Multiple-Antenna Channel," *IEEE Transactions on Communications*, vol. 51, no. 3, March, 2003.

[53] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bolcskei, "An Overview of MIMO Communications-A Key to Gigabit Wireless," *Proceedings of the IEEE*, vol. 92, no. 2, February, 2004.

[54] J. Wang, X. Liu, Z. Li, J. Si, "Soft-output MIMO detector for MMSE receiver with channel estimation error," Conference on textitCommunications in China (ICCC Workshops), 2016 IEEE/CIC International, 27-29 July, 2016.

[55] Y. Okamotoa, H. Osawaa, H. Saitoa, H. Muraokab, Y. Nakamurab, "Performance of PRML systems in perpendicular magnetic recording channel with jitter-like noise," *Proceedings of the fifth Per-*

pendicular Magnetic Recording Conference, vol. 235, no. 1-3, pp. 259-264, October, 2001.

[56] H. Ide, "A Modified PRML Channel for Perpendicular Magnetic Recording," *IEEE Transactions on Magnetics*, vol. 32, no. 5, 1996, pp.3965-3967.

[57] N. Zheng, K. S. Venkataraman, A. Kavcic, T. Zhang, "A Study of Multitrack Joint 2-D Signal Detection Performance and Implementation Cost for Shingled Magnetic Recording," *IEEE Transactions on Magnetics*, vol. 50, no. 6, June, 2014.

[58] G.D. Forney, "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Transactions on Information Theory*, vol. 18, no. 3, May, 1972.

[59] G. D. Forney, "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, March, 1973, pp. 268 - 278.

[60] B. Vasic, and E. M. Kurtas, "Coding and Signal Processing for Magnetic Recording Systems," *CRC Press*, United States of America, 2005.

[61] C. Fleming, "A Tutorial on Convolutional Coding with Viterbi Decoding," *Spectrum Applications*, http://home.netcom.com/c̃hip.f/viterbi/tutorial.html, 2011, Acessed July 2016.

[62] J. Hagenauer, P, Hoeher, "A Viterbi Algorithm with Soft Decision Outputs and its Applications," in *Proceedings of Global Telecommunications Conference and Exhibition*, pp. 1680-1686, 1989.

[63] R. Wood, "Shingled Magnetic Recording and Two-Dimensional Magnetic Recording," presented at *IEEE Magnetics Society*, Santa Clara Valley Chapter, Hitachi GST, San Jose, California, 19 October,2010.

[64] R. J. McEliece. “On the BCJR Trellis for Linear Block Codes,” *IEEE Transactions on Information Theory*, vol. 42, no. 4, July, 1996.

[65] E. F. Haratsch, “Advanced Read Channel Technologies,” Lecture at *LSI Corporation*, November 16-17, 2011.

[66] S. Mita, V. T. Van, and F. Haga, “Reduction of Bit Errors Due to Intertrack Interference Using LLRs of Neighboring Tracks,” *IEEE Transactions on Magnetics*, vol. 47, no. 10, October, 2011.

[67] M. Yamashita, Y. Okamoto, Y. Nakamura, H. Osawa, and H. Muraoka, “Performance Evaluation of Neuro ITI Canceller for Two-Dimensional Magnetic Recording by Shingled Magnetic Recording,” *IEEE Transactions on Magnetics*, vol. 49, no. 7, July, 2013.

[68] N. Zheng, and T. Zhang, “Design of Low-Complexity 2-D SOVA Detector for Shingled Magnetic Recording,” *IEEE Transactions on Magnetics*, vol. 51, no. 4, April 2015.

[69] J. G. Proakis, “Digital Communications”. Fourth Edition. *McGraw-Hill Book Company*, 2007.

[70] D. J. Costello, A. E. Pusane, S. Bates, and K. Sh. Zigangirov, “A Comparison Between LDPC Block and Convolutional Codes,” Procedings on Information Theory and Applications’ Workshop, 2006.

[71] M. Z. Ahmed “Crosstalk-Resilient Coding for High Density Digital Recording,” PhD dissertation, *Department of Communication and Electronic Engineering, Faculty of Technology*, University of Plymouth, UK. June, 2003.

[72] T. Wu, M. A. Armand, X. Jiao. “On Reed-Solomon Codes as Outer Codes in the Davey-MacKay Construction for Channels with Insertions and Deletions,” *8th International Conference on*

Information, Communications and Signal-Processing (ICICS 2011). pp. 1-5, December, 2011.

[73] R. Galbraith and T. Oenning, “Iterative Detection Read Channel Technology in Hard Disk Drives,” White Paper by *HGST*, November, 2008.

[74] L. M. M. Myint, P. Supnithi, and P. Tantaswadi, “An Inter-Track Interference Mitigation Technique Using Partial ITI Estimation in Patterned Media Storage,” *IEEE Transactions on Magnetics*, vol. 45, no. 10, October, 2009.

[75] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, “Numerical Recipes in C: The Art of Scientific Computing,” Second Edition. *Cambridge University Press*, Cambridge, New York, Port Chester, Melbourne, Sydney, 2002.

[76] S. Crozier and P. Guinand, “High-Performance Low-Memory Interleaver Banks for Turbo-Codes,” *Communications Research Centre*, Ottawa, Canada, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=957178, pp. 2394 - 2398, Accessed July 2016.

[77] Don H. Johnson, “Signal-to-noise ratio,” *Scholarpedia*, available at http://www.scholarpedia.org/article/Signal-to-noise˙ratio, 2006, Accessed in April 2017.

[78] S.J. Greaves, H. Muraoka, Y. Kanai, “Simulations of Recording Media for 1Tb/in2,” textitJournal of Magnetism and Magnetic Materials, 320, pp. 2889-2893, 2008.

[79] Altera, “My First FPGA Design Tutorial,” July, 2008, Available at *https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/tt/tt_my_first_fpga.pdf*, Accessed in July, 2016.

[80] Tearasic Sockit, “Sockit User Manual,” 2013, Available at *https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=167&No=816* Accessed in July, 2016.

[81] A. R. Krishnan, R. Radhakrishnan, and B. Vasic, “LDPC Decoding Strategies for Two Dimensional Magnetic Recording,” *IEEE Global Communications Conference*, Honolulu, Hi, USA, 30th November to 4th December, 2009.

Appendices

Appendix A

Published Papers

Removed.

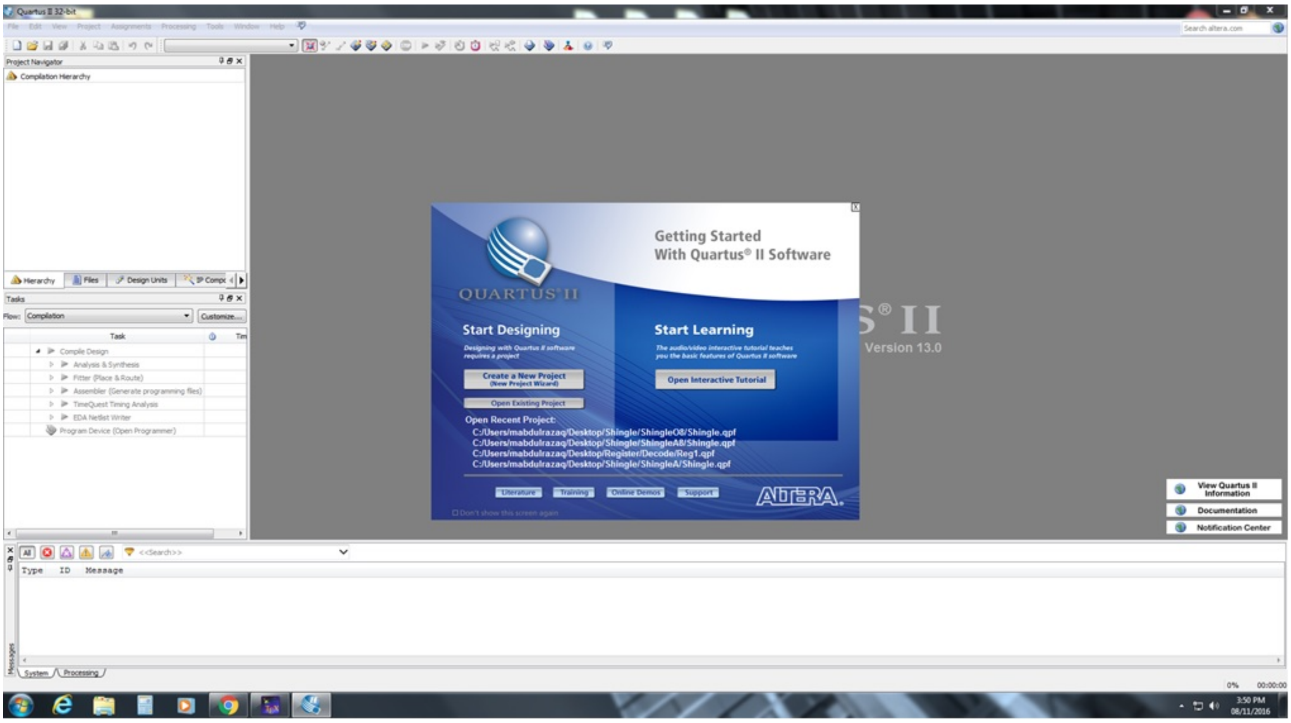
Appendix B

Codes

Removed.

Appendix C

FPGA Programming



Getting Started With Quartus® II Software

Start Designing

Designing with Quartus II software requires a project

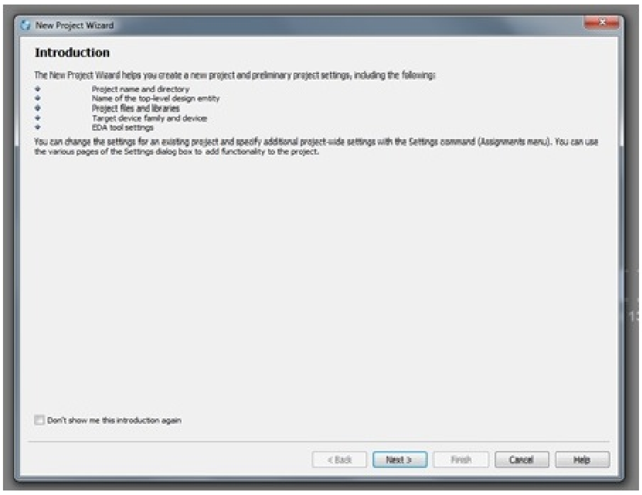
Start Learning

The audio/video interactive tutorial teaches you the basic features of Quartus II software

Open Recent Project:

C:/Users/mabdulrazaq/Desktop/Shingle/ShingleO8/Shingle.qpf
C:/Users/mabdulrazaq/Desktop/Shingle/ShingleA8/Shingle.qpf
C:/Users/mabdulrazaq/Desktop/Register/Decode/Reg1.qpf
C:/Users/mabdulrazaq/Desktop/Shingle/ShingleA/Shingle.qpf

Altera Quartus II 14.1 is going to be used to demonstrate the procedure for creating a project and programming the FPGA board. The FPGA board in use is TearAsic SockIt which has Altera Cyclone V 5CXFC6D6F31C8NES on it.
It is assumed that the codes are already written and all saved in a directory where the project will be saved.
When the software is started, the welcome screen shown above displays. The new project button will be available on the welcome screen. Click on it to start.



New Project Wizard

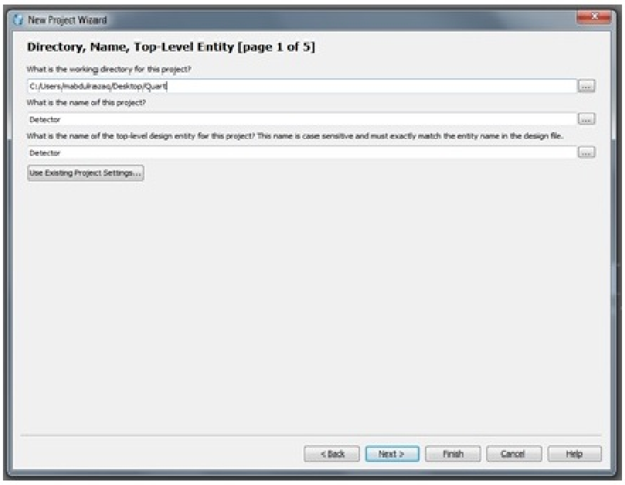
Introduction

The New Project Wizard helps you create a new project and preliminary project settings, including the following:

- Project name and directory
- Name of the top-level design entity
- Project files and libraries
- Target device family and device
- EDA tool settings

You can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project.

□ Don't show me this introduction again



New Project Wizard

Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?

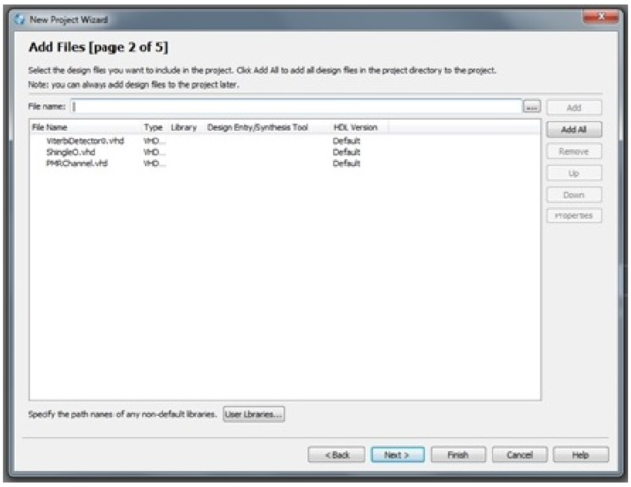
C:/Users/mabdulrazaq/Desktop/Quart

What is the name of this project?

Detector

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

Detector



New Project Wizard

Add Files [page 2 of 5]

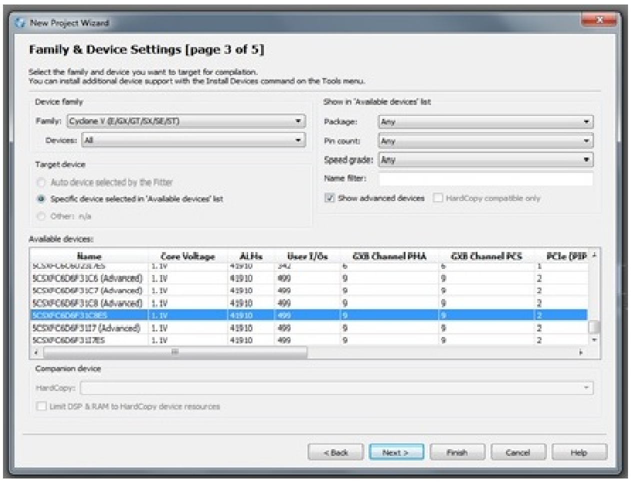
Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project.

Note: you can always add design files to the project later.

File name:

File Name	Type	Library	Design Entry/Synthesis Tool	HDL Version
ViterbiDetector0.vhd	VHD...			Default
Shingle0.vhd	VHD...			Default
PMRChannel.vhd	VHD...			Default

Specify the path names of any non-default libraries.



New Project Wizard

Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

Device family

Family: Cyclone V (E/GX/GT/SX/SE/ST)

Devices: All

Target device

○ Auto device selected by the Fitter

● Specific device selected in 'Available devices' list

○ Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: Any

Speed grade: Any

Name filter:

☑ Show advanced devices □ HardCopy compatible only

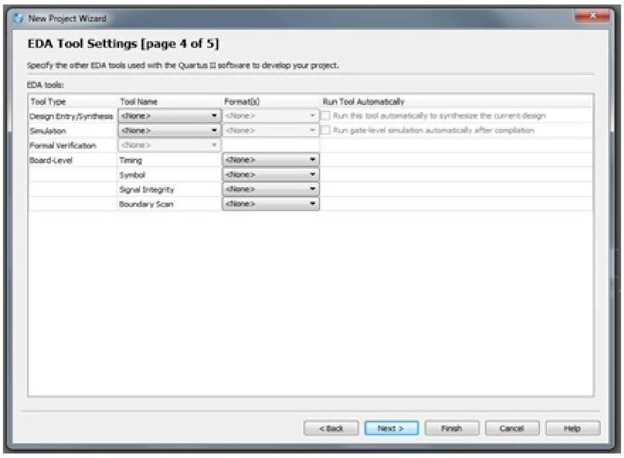
Available devices:

Name	Core Voltage	ALMs	User I/Os	GXB Channel PMA	GXB Channel PCS	PCIe (PIP
5CSXFC6C6U23I7ES	1.1V	41910	342	6	6	1
5CSXFC6D6F31C6 (Advanced)	1.1V	41910	499	9	9	2
5CSXFC6D6F31C7 (Advanced)	1.1V	41910	499	9	9	2
5CSXFC6D6F31C8 (Advanced)	1.1V	41910	499	9	9	2
5CSXFC6D6F31C8ES	1.1V	41910	499	9	9	2
5CSXFC6D6F31I7 (Advanced)	1.1V	41910	499	9	9	2
5CSXFC6D6F31I7ES	1.1V	41910	499	9	9	2

Companion device

HardCopy:

□ Limit DSP & RAM to HardCopy device resources



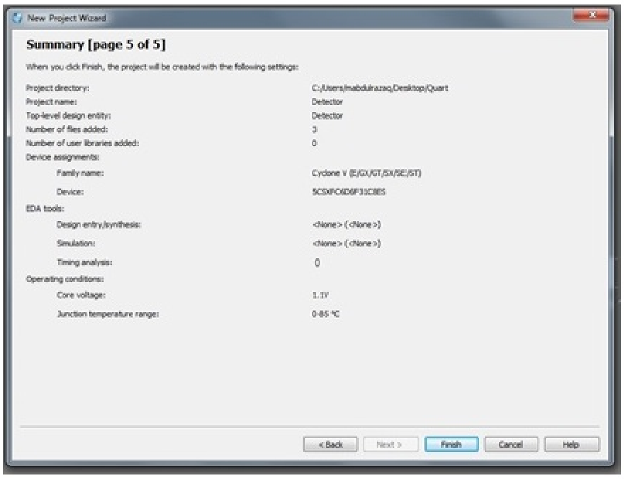
New Project Wizard

EDA Tool Settings [page 4 of 5]

Specify the other EDA tools used with the Quartus II software to develop your project.

EDA tools:

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synthesis	<None>	<None>	□ Run this tool automatically to synthesize the current design
Simulation	<None>	<None>	□ Run gate-level simulation automatically after compilation
Formal Verification	<None>		
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	



New Project Wizard

Summary [page 5 of 5]

When you click Finish, the project will be created with the following settings:

Project directory:	C:/Users/mabdulrazaq/Desktop/Quart
Project name:	Detector
Top-level design entity:	Detector
Number of files added:	3
Number of user libraries added:	0
Device assignments:	
Family name:	Cyclone V (E/GX/GT/SX/SE/ST)
Device:	5CSXFC6D6F31C8ES
EDA tools:	
Design entry/synthesis:	<None> (<None>)
Simulation:	<None> (<None>)
Timing analysis:	0
Operating conditions:	
Core voltage:	1.1V
Junction temperature range:	0-85 ℃

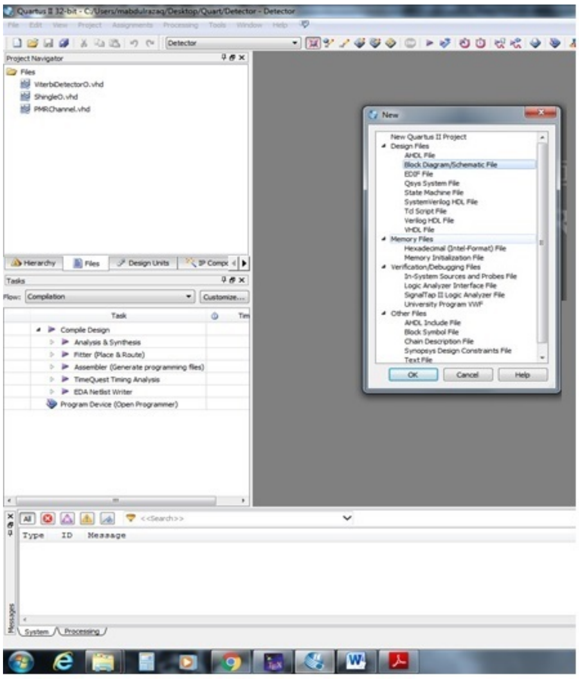
On the Introduction screen, click “Next”.
On directory screen, choose directory where project is to be saved and Project name.
On Add file screen, select all the program files for the project and click

"Add". Click "Next" after that.

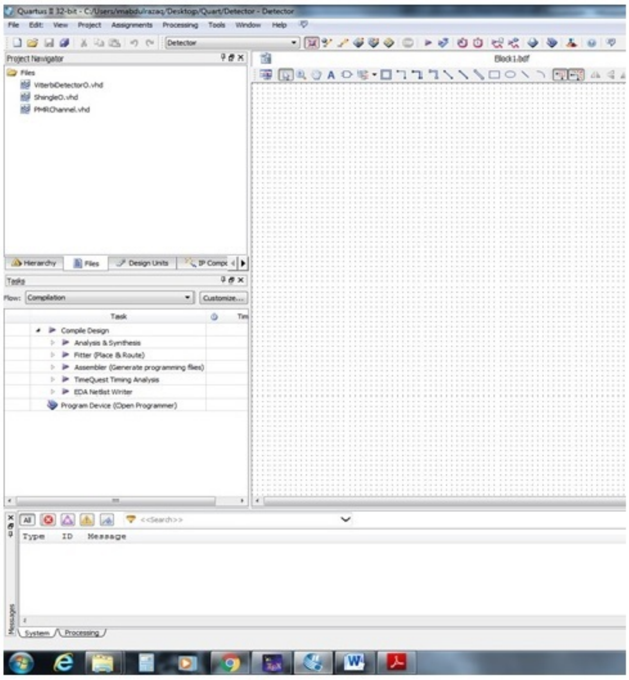
Family & Device Settings page appears. In the device family section, select Cyclone V option (or appropriate device family). In the available device section, select your device number or the closest to it (5CXFC6D6F31C8NES). Click on next.

On the EDA tool settings page, you can just click "Next".

A Summary page appears with the details of the option chosen. Click "Finish" to finish creating the new project.



A



B

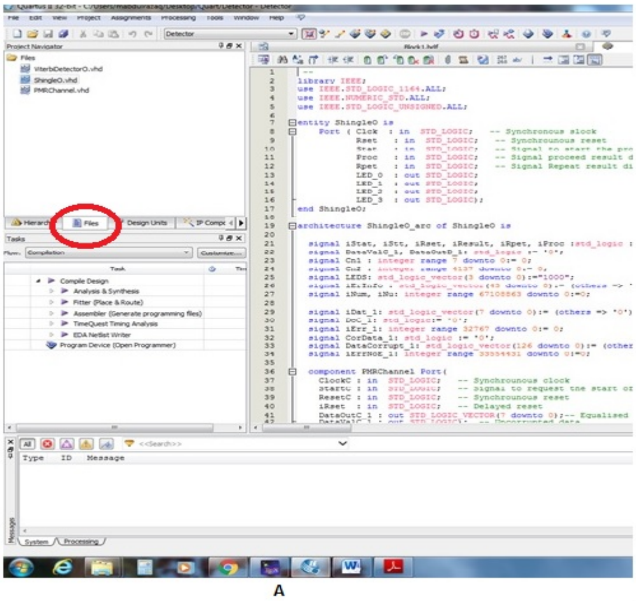
A window showing Menus, tools, project area, files, hierarchy opens. Select "File" menu on the Menu bar. Select "New" and a window pops up with multiple options.

Select "Block Diagram/Schematic File" then click "OK".

A spotted window appears in the project area as shown in B above.

On the Project Navigator pane by the side, Click on Files tab. Double click the top-level VHDL file(s). Top level file is the file that contains other file and will feature on the block diagram.

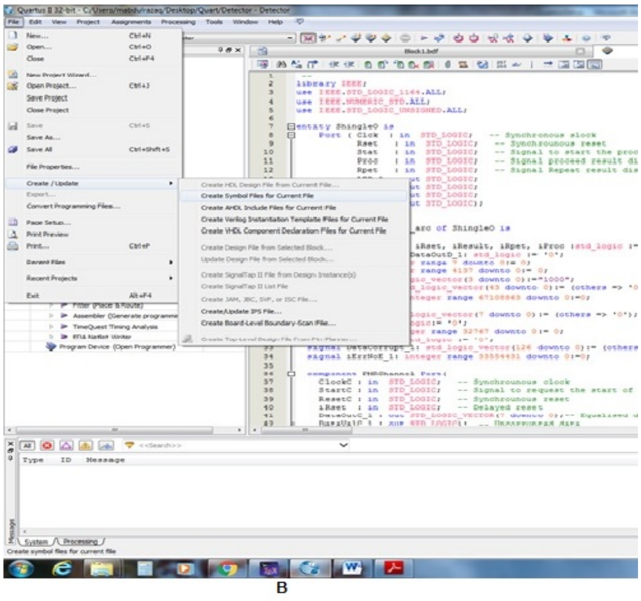
While the file is open as shown in A below, select "File" on the menu bar, navigate to “Create/Update” and select the option “Create Symbol File for Current File”.



```
--
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Shingle0 is
    Port ( Clck  : in  STD_LOGIC;    -- Synchronous slock
           Rset  : in  STD_LOGIC;    -- Synchrounous reset
           Stat  : in  STD_LOGIC;    -- Signal to start the pro
           Proc  : in  STD_LOGIC;    -- Signal proceed result d
           Rpet  : in  STD_LOGIC;    -- Signal Repeat result di
           LED_0 : out STD_LOGIC;
           LED_1 : out STD_LOGIC;
           LED_2 : out STD_LOGIC;
           LED_3 : out STD_LOGIC);
end Shingle0;

architecture Shingle0_arc of Shingle0 is

  signal iStat, iStt, iRset, iResult, iRpet, iProc :std_logic :
  signal DataValC_1, DataOutD_1: std_logic := '0';
  signal Cn1 : integer range 7 downto 0:= 0;
  signal Cn2 : integer range 4137 downto 0:= 0;
  signal LEDS: std_logic_vector(3 downto 0):="1000";
  signal iErInfo : std_logic_vector(43 downto 0):= (others => '
  signal iNum, iNu: integer range 67108863 downto 0:=0;

  signal iDat_1: std_logic_vector(7 downto 0):= (others => '0')
  signal DoC_1: std_logic:= '0';
  signal iErr_1: integer range 32767 downto 0:= 0;
  signal CorData_1: std_logic := '0';
  signal DataCorrupt_1: std_logic_vector(126 downto 0):= (other
  signal iErrNoE_1: integer range 33554431 downto 0:=0;

  component PMRChannel Port(
    ClockC : in  STD_LOGIC;    -- Synchrounous clock
    StartC : in  STD_LOGIC;    -- Signal to request the start of
    ResetC : in  STD_LOGIC;    -- Synchrounous reset
    iRset  : in  STD_LOGIC;    -- Delayed reset
    DataOutC_1 : out STD_LOGIC_VECTOR(7 downto 0);-- Equalised
    DataValC_1 : out STD_LOGIC); -- Uncorrupted data
```

Type ID Message

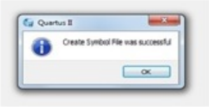
Project Navigator
Files
ViterbiDetector0.vhd
Shingle0.vhd
PMRChannel.vhd
Files
Design Units
Tasks
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming files)
TimeQuest Timing Analysis
EDA Netlist Writer
Program Device (Open Programmer)
Block1.bdf
System
Processing
A



Quartus II 32-bit - C:/Users/mabdulrazaq/Desktop/Quart/Detector - Detector
File
Edit
View
Project
Assignments
Processing
Tools
Window
Help
New... Ctrl+N
Open... Ctrl+O
Close Ctrl+F4
New Project Wizard...
Open Project... Ctrl+J
Save Project
Close Project
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
File Properties...
Create / Update
Export...
Convert Programming Files...
Page Setup...
Print Preview
Print... Ctrl+P
Recent Files
Recent Projects
Exit Alt+F4
Create HDL Design File from Current File...
Create Symbol Files for Current File
Create AHDL Include Files for Current File
Create Verilog Instantiation Template Files for Current File
Create VHDL Component Declaration Files for Current File
Create Design File from Selected Block...
Update Design File from Selected Block...
Create SignalTap II File from Design Instance(s)
Create SignalTap II List File
Create JAM, JBC, SVF, or ISC File...
Create/Update IPS File...
Create Board-Level Boundary-Scan File...
Create Top-Level Design File From Pin Planner...
Create symbol files for current file
B

When the symbol file is created successfully, a notification window appears as shown in A below. (If other files need to be on the schematic, symbol files should be created for them too).
On the Schematic, right-click and select “Insert” then “Symbol”. A symbol window, as shown in C below, pops up with created symbols under “project” and others available in the software.

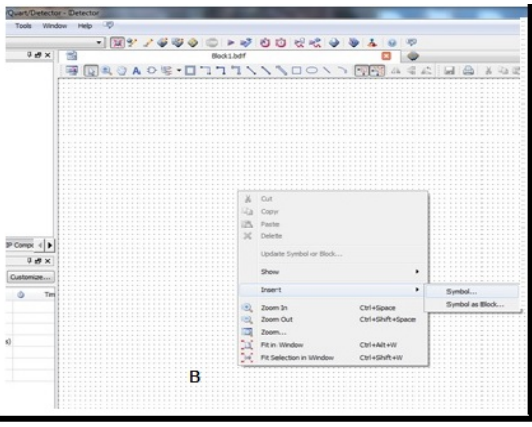
Select the appropriate symbol and click “OK” to place it on the schematic file.
At this point, we can save the schematic using “Ctrl+S” or other forms. Name should be given to th schematic (different from top level files) and saved.
Right-click on the schematic, choose “Insert”, then “Symbol”. From the Quartus Library, under primitive, choose logic and select “not” gate and click “OK” to insert it on the schematic. Put the required number of logic gates.



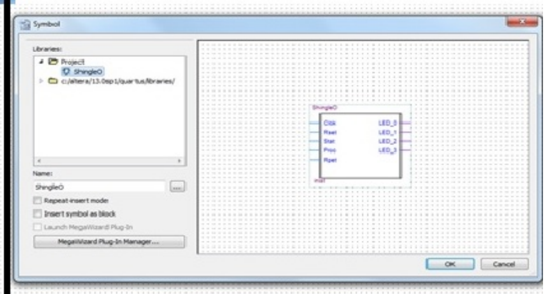
Quartus II

Create Symbol File was successful

A



B



Symbol

Libraries:

- Project
  - Shingle0
- c:/altera/13.0sp1/quartus/libraries/

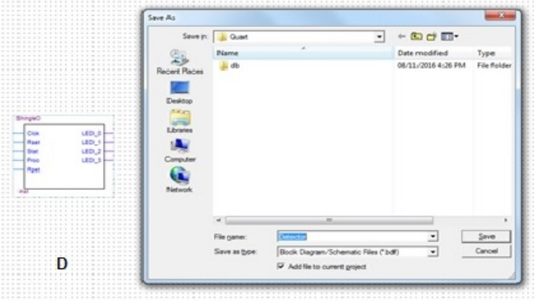
Name: Shingle0

□ Repeat-insert mode

□ Insert symbol as block

□ Launch MegaWizard Plug-In

C



Save As

Save in: Quart

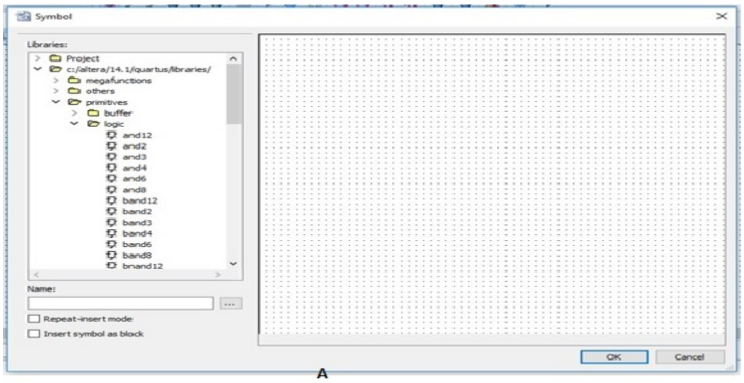
Name	Date modified	Type
db	08/11/2016 4:26 PM	File folder

File name: Detector

Save as type: Block Diagram/Schematic Files (*.bdf)

☑ Add file to current project

D



Symbol

Libraries:

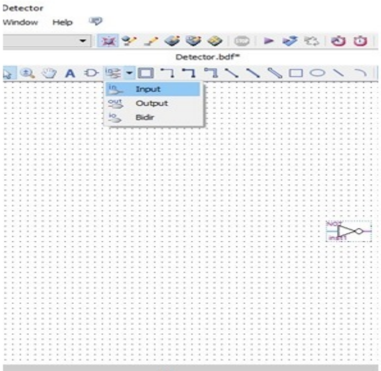
- Project
- c:/altera/14.1/quartus/libraries/
  - megafunctions
  - others
  - primitives
    - buffer
    - logic
      - and12
      - and2
      - and3
      - and4
      - and6
      - and8
      - band12
      - band2
      - band3
      - band4
      - band6
      - band8
      - bnand12

Name:

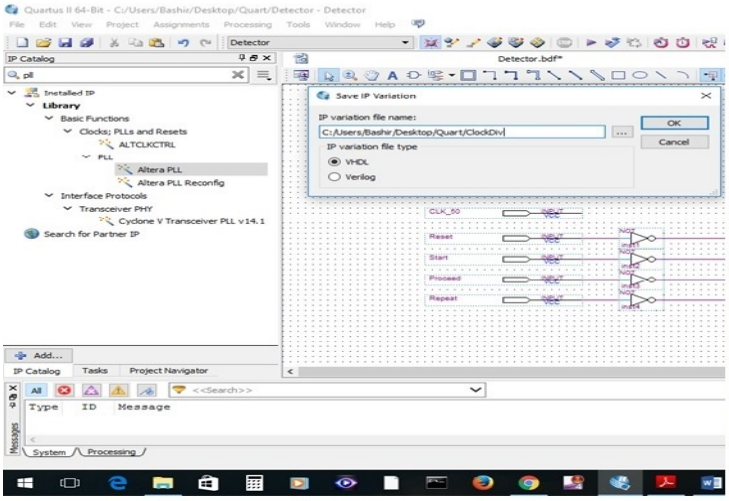
□ Repeat-insert mode

□ Insert symbol as block

A



B



Quartus II 64-Bit - C:/Users/Bashir/Desktop/Quart/Detector - Detector

Save IP Variation

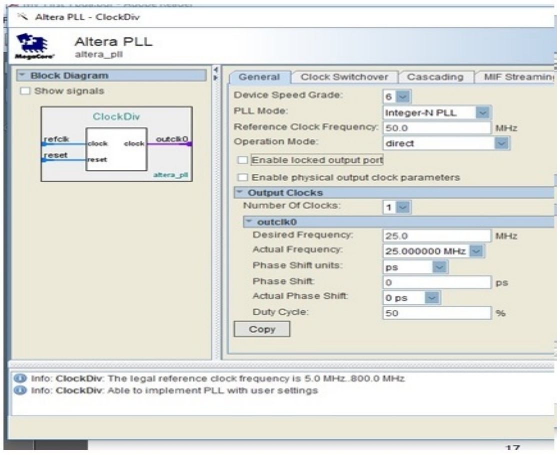
IP variation file name: C:/Users/Bashir/Desktop/Quart/ClockDiv

IP variation file type

● VHDL

○ Verilog

C



Altera PLL - ClockDiv

Altera PLL
altera_pll

Device Speed Grade: 6

PLL Mode: Integer-N PLL

Reference Clock Frequency: 50.0 MHz

Operation Mode: direct

□ Enable locked output port

□ Enable physical output clock parameters

Output Clocks

Number Of Clocks: 1

outclk0

Desired Frequency: 25.0 MHz

Actual Frequency: 25.000000 MHz

Phase Shift units: ps

Phase Shift: 0 ps

Actual Phase Shift: 0 ps

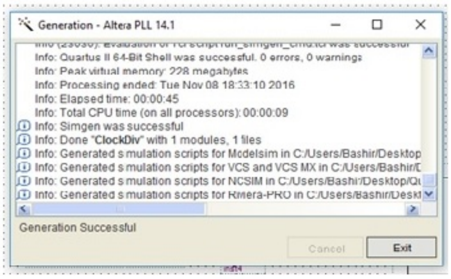
Duty Cycle: 50 %

Info: ClockDiv: The legal reference clock frequency is 5.0 MHz..800.0 MHz

Info: ClockDiv: Able to implement PLL with user settings

D

On the toolbar above the schematic, select “input”, to add input pins or ‘output” for output pins.
From the toolbar, select connecting wires to connect logic gates to pins and to other symbol inputs and outputs.
To add PLL for clocking modification, select “IP Catalog” tab on the side pane, type “PLL” in the search box (this can be found using “Insert” on the schematic in some versions of the software). Select “Altera PLL”, give it a name, and VHDL as the language of the file, then click “OK”. A window shown in D above pops up. For Cyclone V, select “speed grade” 6, reference clock frequency 50 MHz, un-check “Enable Locked Output port”, device frequency 5 MHz (this can be changed later after compilation shows different suitable frequency), click “OK”.

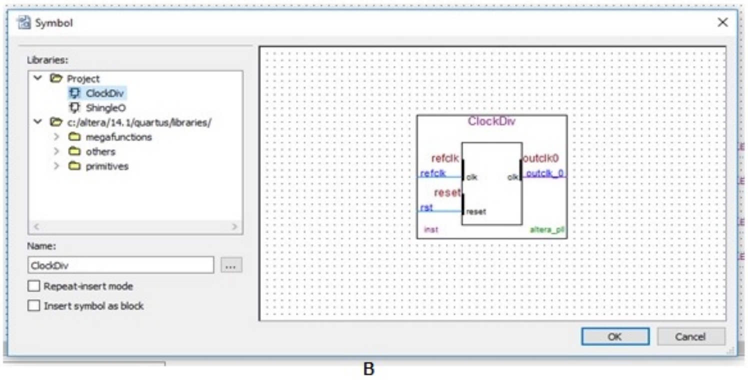


Generation - Altera PLL 14.1

```
Info: Quartus II 64-Bit Shell was successful. 0 errors, 0 warnings
Info: Peak virtual memory: 228 megabytes
Info: Processing ended: Tue Nov 08 18:33:10 2016
Info: Elapsed time: 00:00:45
Info: Total CPU time (on all processors): 00:00:09
Info: Simgen was successful
Info: Done "ClockDiv" with 1 modules, 1 files
Info: Generated simulation scripts for Modelsim in C:/Users/Bashir/Desktop
Info: Generated simulation scripts for VCS and VCS MX in C:/Users/Bashir/C
Info: Generated simulation scripts for NCSIM in C:/Users/Bashir/Desktop/Qu
Info: Generated simulation scripts for Riviera-PRO in C:/Users/Bashir/Deskt
```

Generation Successful

A



Symbol

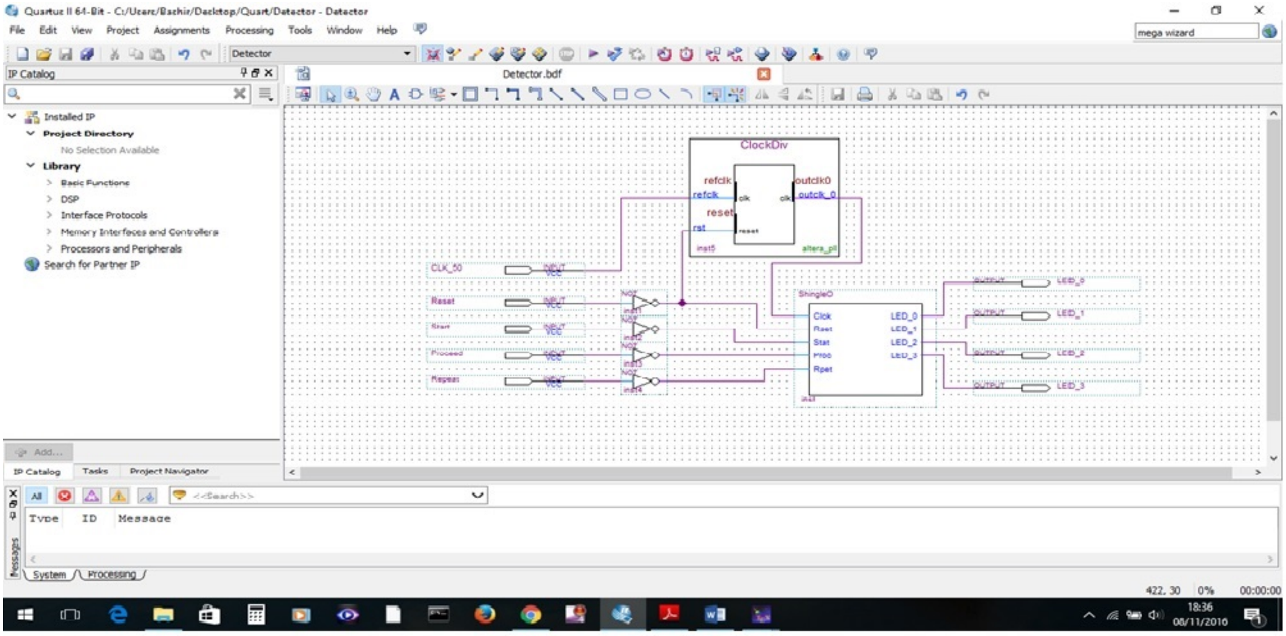
Libraries:
- Project
  - ClockDiv
  - ShingleO
- c:/altera/14.1/quartus/libraries/
  - megafunctions
  - others
  - primitives

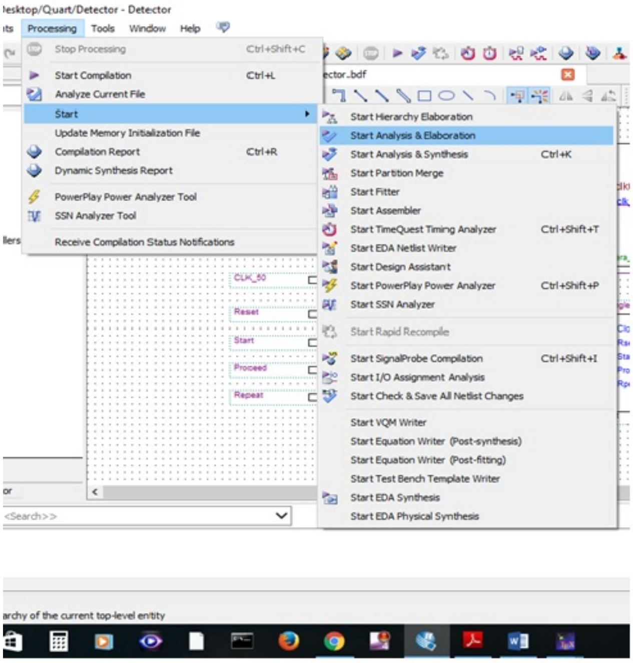
Name: ClockDiv

□ Repeat-insert mode
□ Insert symbol as block

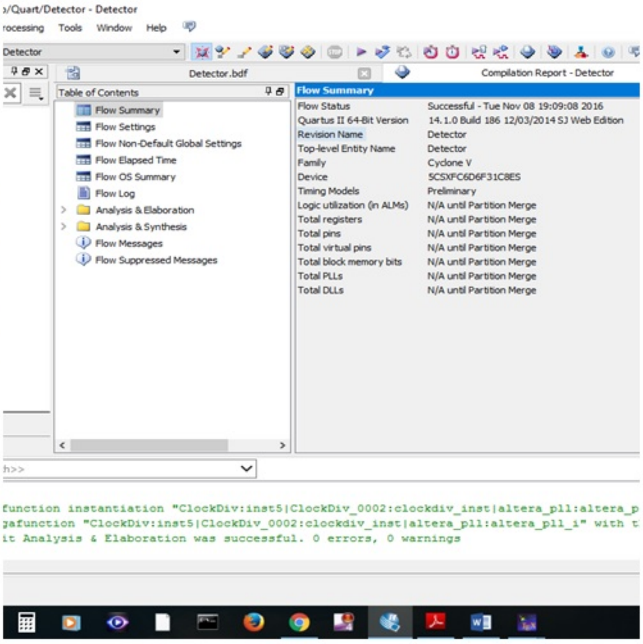
B

A window showing the process of generating the PLL shows as in A above. After a successful generation, click exit. We can now go to the “Insert” option under Project and Insert the PLL created just like other symbol files.
Connect the circuit in the appropriate order as shown in the figure below. Double-click on the names of pins to modify the names to suitable ones.
After all components are placed, connected and named, the project is now ready to be assigned to pins of the Cylone V IC on the SOCKIT board.




Processing Tools Window Help
Stop Processing Ctrl+Shift+C
Start Compilation Ctrl+L
Analyze Current File
Start
Update Memory Initialization File
Compilation Report Ctrl+R
Dynamic Synthesis Report
PowerPlay Power Analyzer Tool
SSN Analyzer Tool
Receive Compilation Status Notifications
Start Hierarchy Elaboration
Start Analysis & Elaboration
Start Analysis & Synthesis Ctrl+K
Start Partition Merge
Start Fitter
Start Assembler
Start TimeQuest Timing Analyzer Ctrl+Shift+T
Start EDA Netlist Writer
Start Design Assistant
Start PowerPlay Power Analyzer Ctrl+Shift+P
Start SSN Analyzer
Start Rapid Recompile
Start SignalProbe Compilation Ctrl+Shift+I
Start I/O Assignment Analysis
Start Check & Save All Netlist Changes
Start VQM Writer
Start Equation Writer (Post-synthesis)
Start Equation Writer (Post-fitting)
Start Test Bench Template Writer
Start EDA Synthesis
Start EDA Physical Synthesis

A

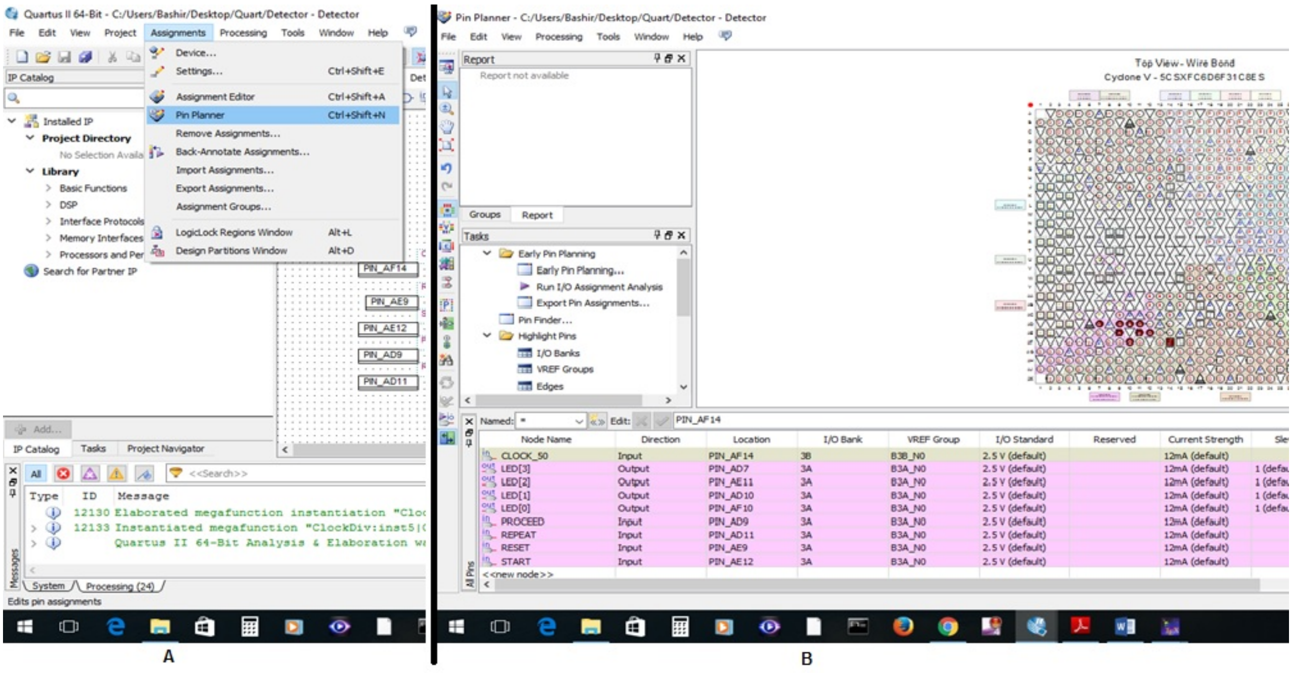


Flow Summary	
Flow Status	Successful - Tue Nov 08 19:09:08 2016
Quartus II 64-Bit Version	14.1.0 Build 186 12/03/2014 SJ Web Edition
Revision Name	Detector
Top-level Entity Name	Detector
Family	Cyclone V
Device	5CSXFC6D6F31C8ES
Timing Models	Preliminary
Logic utilization (in ALMs)	N/A until Partition Merge
Total registers	N/A until Partition Merge
Total pins	N/A until Partition Merge
Total virtual pins	N/A until Partition Merge
Total block memory bits	N/A until Partition Merge
Total PLLs	N/A until Partition Merge
Total DLLs	N/A until Partition Merge

```
function instantiation "ClockDiv:inst5|ClockDiv_0002:clockdiv_inst|altera_pll:altera_p
gafunction "ClockDiv:inst5|ClockDiv_0002:clockdiv_inst|altera_pll:altera_pll_i" with t
it Analysis & Elaboration was successful. 0 errors, 0 warnings
```

B

To prepare the project for pin assignment, on the Menu bar, select "Processing", "Start" then "Start Analysis & Elaboration". After a successful analysis and elaboration, a report should show up in the project area as shown in figure B above.


Quartus II 64-Bit - C:/Users/Bashir/Desktop/Quart/Detector - Detector
File Edit View Project Assignments Processing Tools Window Help
Device...
Settings... Ctrl+Shift+E
Assignment Editor Ctrl+Shift+A
Pin Planner Ctrl+Shift+N
Remove Assignments...
Back-Annotate Assignments...
Import Assignments...
Export Assignments...
Assignment Groups...
LogicLock Regions Window Alt+L
Design Partitions Window Alt+D
PIN_AF14
PIN_AE9
PIN_AE12
PIN_AD9
PIN_AD11
A
Pin Planner - C:/Users/Bashir/Desktop/Quart/Detector - Detector
File Edit View Processing Tools Window Help
Report
Report not available
Groups Report
Tasks
Early Pin Planning
Early Pin Planning...
Run I/O Assignment Analysis
Export Pin Assignments...
Pin Finder...
Highlight Pins
I/O Banks
VREF Groups
Edges
Top View - Wire Bond
Cyclone V - 5CSXFC6D6F31C8ES
B

Type ID Message

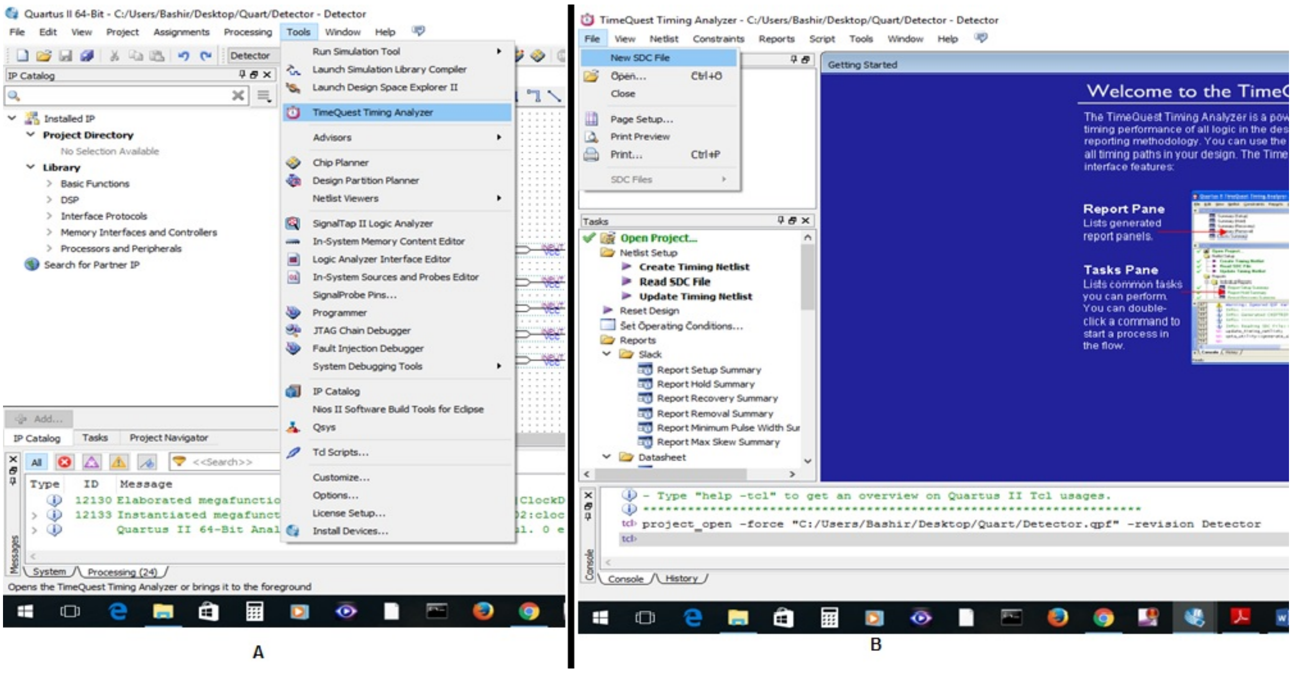
```
12130 Elaborated megafunction instantiation "Cloc
12133 Instantiated megafunction "ClockDiv:inst5|c
      Quartus II 64-Bit Analysis & Elaboration wa
```

Edits pin assignments

Named: * Edit: PIN_AF14

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Sle
CLOCK_50	Input	PIN_AF14	3B	B3B_N0	2.5 V (default)		12mA (default)	
LED[3]	Output	PIN_AD7	3A	B3A_N0	2.5 V (default)		12mA (default)	1 (defa
LED[2]	Output	PIN_AE11	3A	B3A_N0	2.5 V (default)		12mA (default)	1 (defa
LED[1]	Output	PIN_AD10	3A	B3A_N0	2.5 V (default)		12mA (default)	1 (defa
LED[0]	Output	PIN_AF10	3A	B3A_N0	2.5 V (default)		12mA (default)	1 (defa
PROCEED	Input	PIN_AD9	3A	B3A_N0	2.5 V (default)		12mA (default)	
REPEAT	Input	PIN_AD11	3A	B3A_N0	2.5 V (default)		12mA (default)	
RESET	Input	PIN_AE9	3A	B3A_N0	2.5 V (default)		12mA (default)	
START	Input	PIN_AE12	3A	B3A_N0	2.5 V (default)		12mA (default)	
<<new node>>								

From the Menu bar, select “Assignments”, than click on “Pin Planner”. A Pin planner window as in B above shows up.
Details of the pins of the board can be found in the Sockit User manual [80]. For this project, we assigned CLOCK_50 input to PIN_AF14, LED(1) to PIN_AD7 and so on as shown in figure B above.



Type ID Message

```
12130 Elaborated megafunctio
12133 Instantiated megafunct
      Quartus II 64-Bit Anal
```

Opens the TimeQuest Timing Analyzer or brings it to the foreground

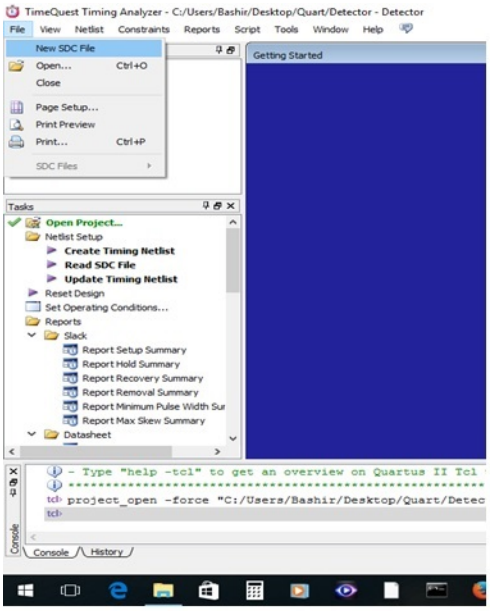
The TimeQuest Timing Analyzer is a pow timing performance of all logic in the des reporting methodology. You can use the all timing paths in your design. The Time interface features:

Lists generated report panels.

Lists common tasks you can perform. You can double-click a command to start a process in the flow.

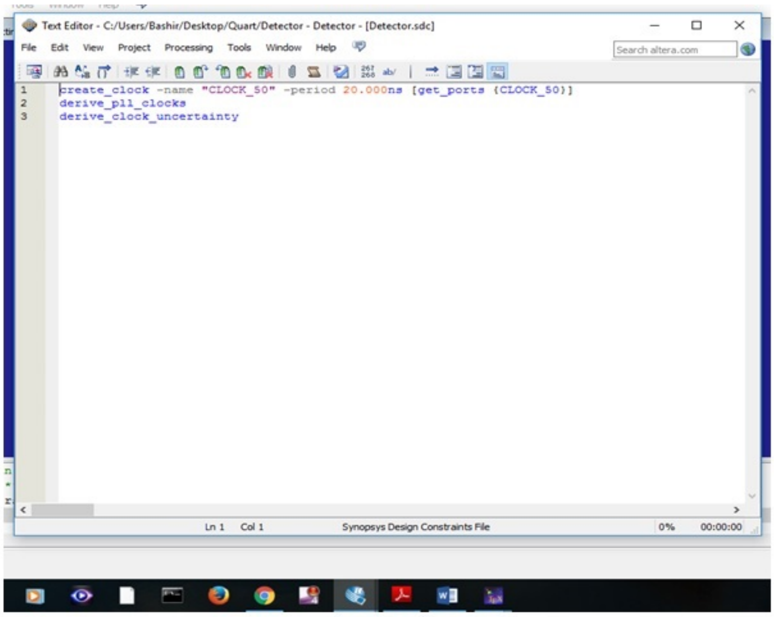
```
- Type "help -tcl" to get an overview on Quartus II Tcl usages.
- ****************************************************************
tcl> project_open -force "C:/Users/Bashir/Desktop/Quart/Detector.qpf" -revision Detector
tcl>
```

Timing analysis is to be carried out in order to verify the speed at which the program can run on the device. On the Menu bar, select "Tools", then "TimeQuest Timing Analyser". A TimeQuest window, as shown in B above, shows up.
On its Menu bar, select "File" menu, then click on "New SDC File". A new empty file opens up as shown in B below. Type the code shown below in B specifying the Clock PIN name, the period of the Clock, the pin in the project connected to it. We use the same name as the one in the project. Save the file with the same name as the project schematic, for the program to automatically assign it to the project. Close the TimeQuest Analyser and return to the main project area.



```
- Type "help -tcl" to get an overview on Quartus II Tcl
***************************************************************
tcl> project_open -force "C:/Users/Bashir/Desktop/Quart/Detec
tcl>
```

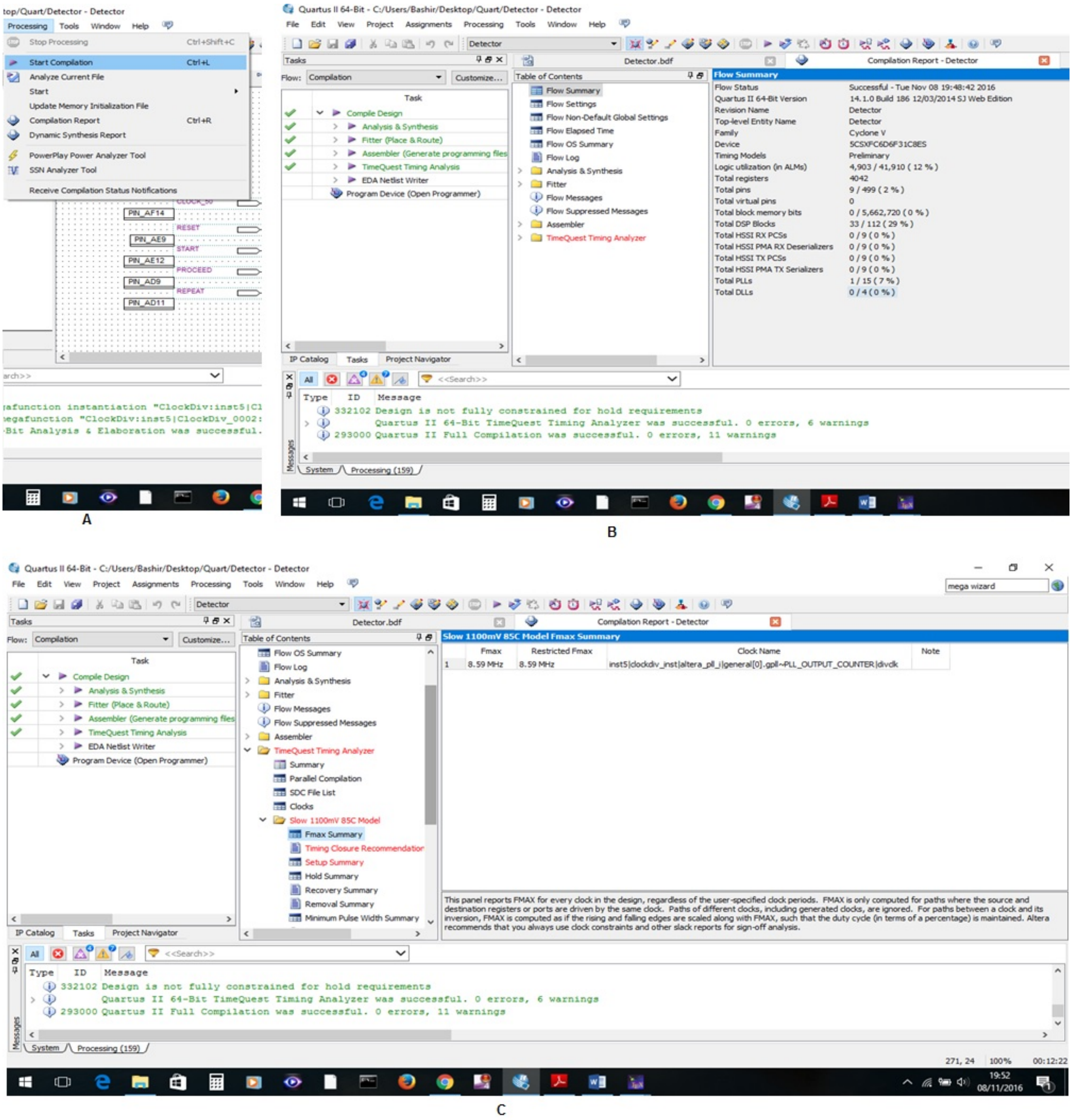
A


Text Editor - C:/Users/Bashir/Desktop/Quart/Detector - Detector - [Detector.sdc]
File Edit View Project Processing Tools Window Help
Search altera.com
Ln 1 Col 1
Synopsys Design Constraints File
0%
00:00:00

```
create_clock -name "CLOCK_50" -period 20.000ns [get_ports {CLOCK_50}]
derive_pll_clocks
derive_clock_uncertainty
```

B

On the Menu bar, select "Processing" then "Start Compilation". A successful compilation will show green tick mark by the side of each process in the "Tasks" pane by the side of the project area and a summary of hardware requirement will be displayed in the project area. To verify if the frequency chosen for the PLL is ok, check the TimeQuest Timing Analyser folder under "Table of contents as shown in C below.


top/Quart/Detector - Detector
Processing
Tools
Window
Help
Stop Processing Ctrl+Shift+C
Start Compilation Ctrl+L
Analyze Current File
Start
Update Memory Initialization File
Compilation Report Ctrl+R
Dynamic Synthesis Report
PowerPlay Power Analyzer Tool
SSN Analyzer Tool
Receive Compilation Status Notifications
PIN_AF14
RESET
PIN_AE9
START
PIN_AE12
PROCEED
PIN_AD9
REPEAT
PIN_AD11
A

```
egafunction instantiation "ClockDiv:inst5|Cl
egafunction "ClockDiv:inst5|ClockDiv_0002:
Bit Analysis & Elaboration was successful.
```

Flow Summary

Flow Status	Successful - Tue Nov 08 19:48:42 2016
Quartus II 64-Bit Version	14.1.0 Build 186 12/03/2014 SJ Web Edition
Revision Name	Detector
Top-level Entity Name	Detector
Family	Cyclone V
Device	5CSXFC6D6F31C8ES
Timing Models	Preliminary
Logic utilization (in ALMs)	4,903 / 41,910 (12 %)
Total registers	4042
Total pins	9 / 499 (2 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	33 / 112 (29 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	1 / 15 (7 %)
Total DLLs	0 / 4 (0 %)

```
Type   ID     Message
       332102 Design is not fully constrained for hold requirements
>             Quartus II 64-Bit TimeQuest Timing Analyzer was successful. 0 errors, 6 warnings
       293000 Quartus II Full Compilation was successful. 0 errors, 11 warnings
```

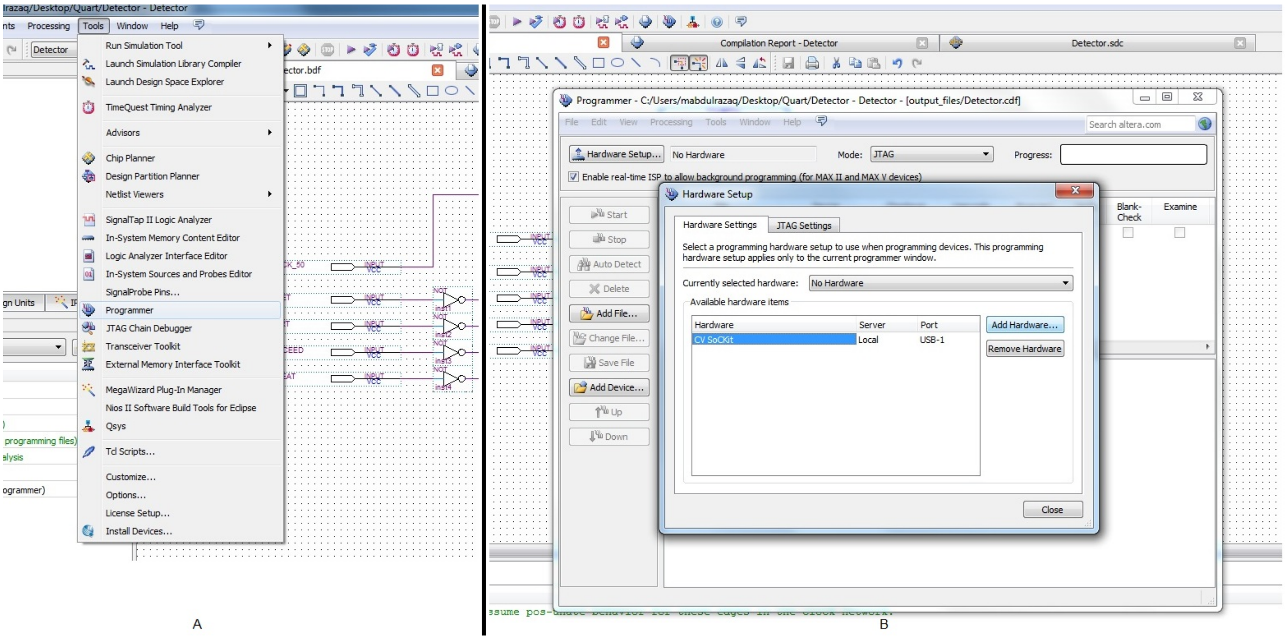
Slow 1100mV 85C Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	8.59 MHz	8.59 MHz	inst5|clockdiv_inst|altera_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER|divclk	

This panel reports FMAX for every clock in the design, regardless of the user-specified clock periods. FMAX is only computed for paths where the source and destination registers or ports are driven by the same clock. Paths of different clocks, including generated clocks, are ignored. For paths between a clock and its inversion, FMAX is computed as if the rising and falling edges are scaled along with FMAX, such that the duty cycle (in terms of a percentage) is maintained. Altera recommends that you always use clock constraints and other slack reports for sign-off analysis.

```
Type   ID     Message
       332102 Design is not fully constrained for hold requirements
>             Quartus II 64-Bit TimeQuest Timing Analyzer was successful. 0 errors, 6 warnings
       293000 Quartus II Full Compilation was successful. 0 errors, 11 warnings
```

If the “FMax” is less than the chosen PLL output frequency, the PLL has to be modified by right-clicking on it and going to its properties. If FMax is very much higher than the selected frequency, increasing the PLL frequency may be considered to maximally utilise the device capabilities. In these cases, the program has to be compiled again and we are now ready to program the device.



Programmer - C:/Users/mabdulrazaq/Desktop/Quart/Detector - Detector - [output_files/Detector.cdf]

Hardware Setup... No Hardware Mode: JTAG Progress:

☑ Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

Hardware Setup

Hardware Settings | JTAG Settings

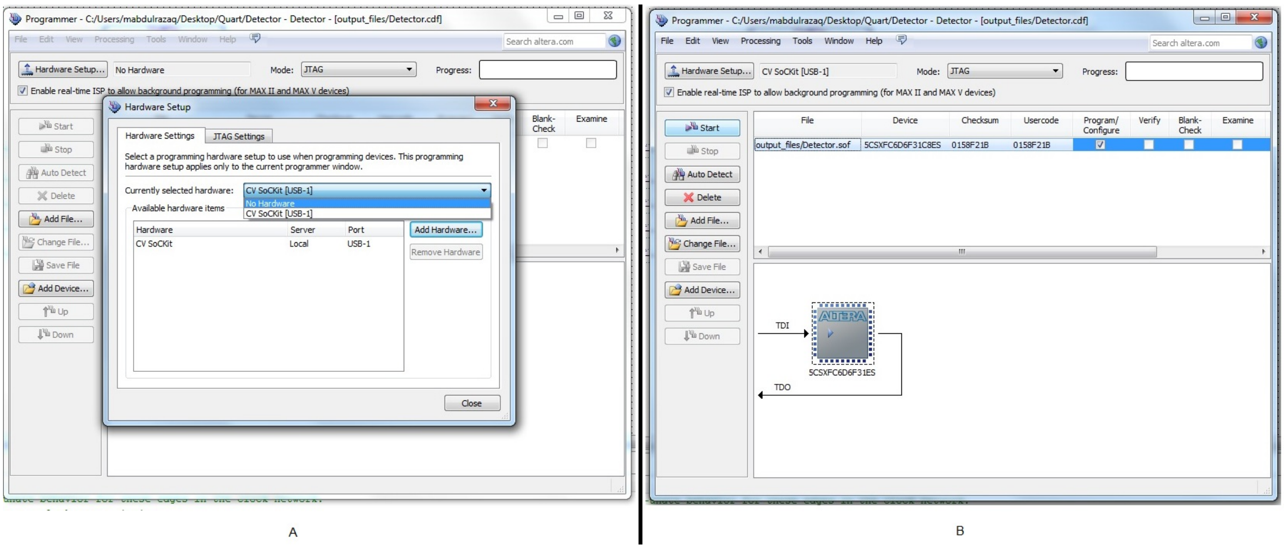
Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

Currently selected hardware: No Hardware

Available hardware items

Hardware	Server	Port
CV SoCKit	Local	USB-1

To program the device, on the Menu bar, select “Tools”, then “Programmer”. The Programmer window opens up. Click the “Hardware Setup” button and another window shows as shown in B above. Make sure the board is connected to the system using the USB blaster cable and is switched on. In the drop box that displays “No Hardware” Click and select “CV SoCKit [USB-1]” and close the hardware setup window. If the “.sof” programming file has not appeared you can manually use “Add File” button to select it.



Programmer - C:/Users/mabdulrazaq/Desktop/Quart/Detector - Detector - [output_files/Detector.cdf]

Hardware Setup... No Hardware Mode: JTAG Progress:

Currently selected hardware: CV SoCKit [USB-1]

No Hardware
CV SoCKit [USB-1]

Hardware	Server	Port
CV SoCKit	Local	USB-1

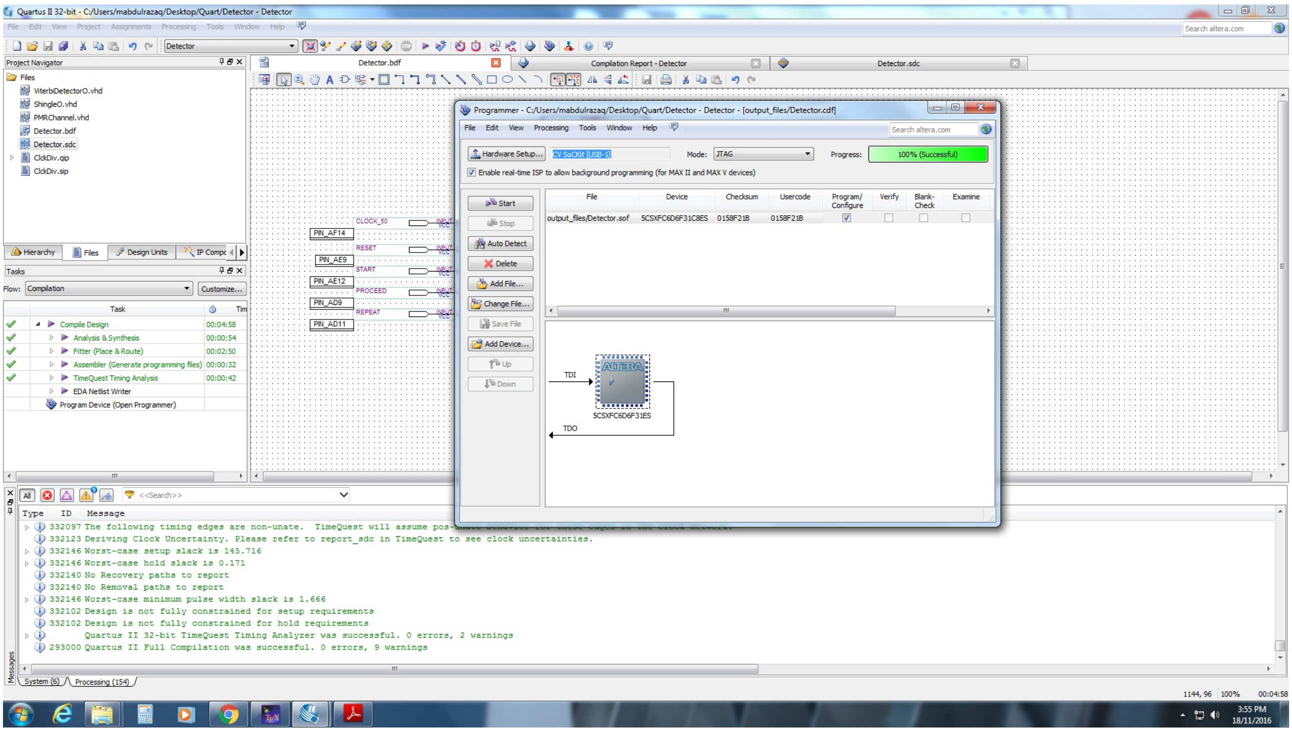
Programmer - C:/Users/mabdulrazaq/Desktop/Quart/Detector - Detector - [output_files/Detector.cdf]

Hardware Setup... CV SoCKit [USB-1] Mode: JTAG Progress:

☑ Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine
output_files/Detector.sof	5CSXFC6D6F31C8ES	0158F21B	0158F21B	☑	□	□	□

Click on the “Start button on the Programmer window and the programming will start. If it is successful, the progress bar at the top of the programmer window will show 100% Successful in green. The board is, therefore, ready to use.



Programmer - C:/Users/mabdulrazaq/Desktop/Quart/Detector - Detector - [output_files/Detector.cdf]

Hardware Setup... CV SoCKit [USB-1] Mode: JTAG Progress: 100% (Successful)

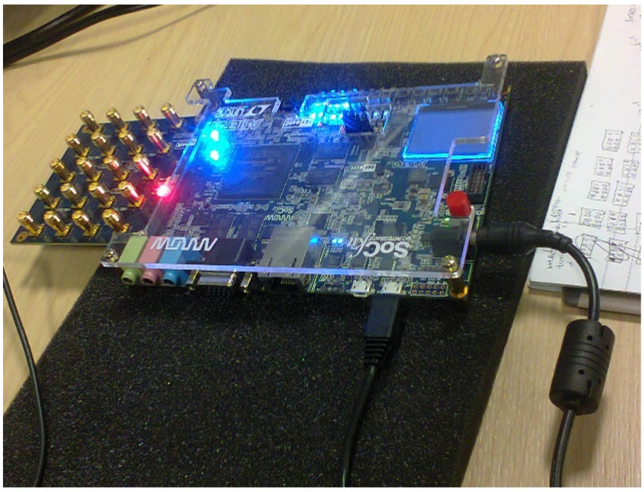
☑ Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine
output_files/Detector.sof	5CSXFC6D6F31C8ES	0158F21B	0158F21B	☑	□	□	□

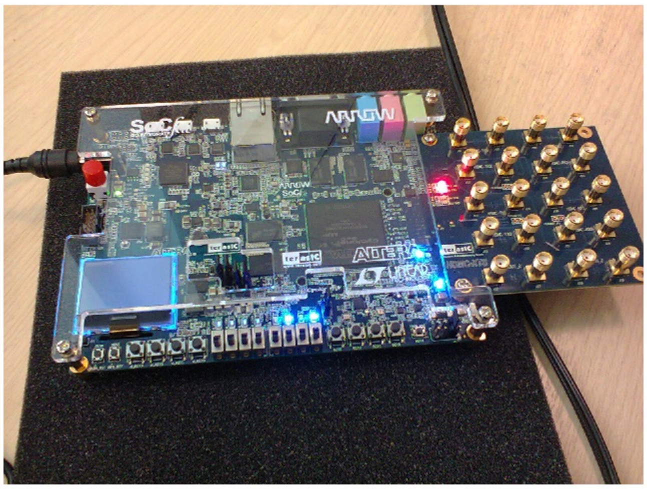
	Task	Time
✓	Compile Design	00:04:58
✓	Analysis & Synthesis	00:00:54
✓	Fitter (Place & Route)	00:02:50
✓	Assembler (Generate programming files)	00:00:32
✓	TimeQuest Timing Analysis	00:00:42
	EDA Netlist Writer	
	Program Device (Open Programmer)	

```
Type  ID     Message
      332097 The following timing edges are non-unate.  TimeQuest will assume pos-unate behavior for these edges in the clock network.
      332123 Deriving Clock Uncertainty. Please refer to report_sdc in TimeQuest to see clock uncertainties.
      332146 Worst-case setup slack is 145.716
      332146 Worst-case hold slack is 0.171
      332140 No Recovery paths to report
      332140 No Removal paths to report
      332146 Worst-case minimum pulse width slack is 1.666
      332102 Design is not fully constrained for setup requirements
      332102 Design is not fully constrained for hold requirements
             Quartus II 32-bit TimeQuest Timing Analyzer was successful. 0 errors, 2 warnings
      293000 Quartus II Full Compilation was successful. 0 errors, 9 warnings
```

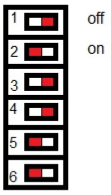
The figure below shows the board in action. Figure C below shows how SW6 (MSEL and CODEC_SEL) switches are arranged for the FPGA programming. Details of the options is found in [80].



A



B



C