UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

ADVANCES IN MODELING AND SIGNAL PROCESSING FOR BIT-PATTERNED MAGNETIC RECORDING CHANNELS WITH WRITTEN-IN ERRORS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

RODNEY C. KEELE
Norman, Oklahoma
2012

ADVANCES IN MODELING AND SIGNAL PROCESSING FOR BIT-PATTERNED MAGNETIC RECORDING CHANNELS WITH WRITTEN-IN ERRORS

A DISSERTATION APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

_____
Dr. J. R. Cruz, Chair

_____
Dr. Robert Palmer

_____
Dr. Joseph Havlicek

_____
Dr. Mark Yeary

_____
Dr. Tian-You Yu

Acknowledgements

Obtaining one's doctorate degree is far from a trivial task. It takes considerable hard work and dedication spread across a timeframe of several years. The commitment required to succeed often encompasses many individuals in addition to the doctor-to-be. In my own case I have found this to be very true and I would be regretfully remiss if I neglected to acknowledge the contributions made by a number of individuals.

To begin I would like to extend both sincere gratitude and a well-deserved thank you to my major professor, Dr. J. R. Cruz. In my many years at the University of Oklahoma, he has always acted in dual capacity as a remarkable resource and as a strong positive influence. I will always appreciate his many contributions and sacrifices that have helped me in numerous ways.

I would also like to express my appreciation to Dr. Robert Palmer, Dr. Joseph Havlicek, Dr. Mark Yeary, and Dr. Tian-You Yu for their time, consideration and support while serving on my doctoral committee.

I also need to thank several fellow students in the Communications Signal Processing Laboratory, most of whom have already graduated, for their helpful discussions and assistance throughout my many years of work and study. For the time and efforts of Dr. Chang Wu, Dr. Richard M. Todd, Dr. Weijun Tan, Dr. Hongxin Song and Jeffrey Enfeng, I am very grateful.

Finally, few things can prematurely end a long journey faster than a lack of motivation. Fortunately, I have never had to experience this during my journey. And for this I must offer my greatest gratitude to my spouse, Susan, my three sons, and my parents. Without their constant support this journey would have never ended because it would have never begun.

Table of Contents

List of Tables

List of Figures

Abstract

In the past perpendicular magnetic recording on continuous media has served as the storage mechanism for the hard-disk drive (HDD) industry, allowing for growth in areal densities approaching 0.5 Tb/in^2. Under the current system design, further increases are limited by the superparamagnetic effect where the medium's thermal energy destabilizes the individual bit domains used for storage. In order to provide for future growth in the area of magnetic recording for disk drives, a number of various technology shifts have been proposed and are currently undergoing considerable research. One promising option involves switching to a discrete medium in the form of individual bit islands, termed bit-patterned magnetic recording (BPMR).

When switching from a continuous to a discrete media, the problems encountered become substantial for every aspect of the hard-disk drive design. In this dissertation the complications in modeling and signal processing for bit-patterned magnetic recording are investigated where the write and read processes along with the channel characteristics present considerable challenges. For a target areal density of 4 Tb/in^2, the storage process is hindered by media noise, two-dimensional (2D) intersymbol interference (ISI), electronics noise and written-in errors introduced during the write process. Thus there is a strong possibility that BPMR may prove intractable as a future HDD technology at high areal densities because the combined negative effects of the many error sources produces an environment where current signal processing techniques cannot accurately recover

the stored data. The purpose here is to exploit advanced methods of detection and error correction to show that data can be effectively recovered from a BPMR channel in the presence of multiple error sources at high areal densities.

First a practical model for the readback response of an individual island is established that is capable of representing its 2D nature with a Gaussian pulse. Various characteristics of the readback pulse are shown to emerge as it is subjected to the degradation of 2D media noise. The writing of the bits within a track is also investigated with an emphasis on the write process's ability to inject written-in errors in the data stream resulting from both a loss of synchronization of the write clock and the interaction of the local-scale magnetic fields under the influence of the applied write field.

To facilitate data recovery in the presence of BPMR's major degradations, various detection and error-correction methods are utilized. For single-track equalization of the channel output, noise prediction is incorporated to assist detection with increased levels of media noise. With large detrimental amounts of 2D ISI and media noise present in the channel at high areal densities, a 2D approach known as multi-track detection is investigated where multiple tracks are sensed by the read heads and then used to extract information on the target track. For BPMR the output of the detector still possesses the uncorrected written-in errors. Powerful error-correction codes based on finite geometries are employed to help recover the original data stream. Increased error-correction is sought by utilizing two-fold EG

codes in combination with a form of automorphism decoding known as auto-diversity. Modifications to the parity-check matrices of the error-correction codes are also investigated for the purpose of attempting more practical applications of the decoding algorithms based on belief propagation. Under the proposed techniques it is shown that effective data recovery is possible at an areal density of 4 Tb/in^2 in the presence of all significant error sources except for insertions and deletions. Data recovery from the BPMR channel with insertions and deletions remains an open problem.

Chapter 1 – Advances in Magnetic Storage Channels

As technology advances so do the demands placed on current embodiments of that technology. With the onset and aggressive growth of the information age, the need for greater and more reliable storage of the multitude of digital versions that embody human knowledge and interest continues to increase. One of the most important devices responsible for meeting such needs is the hard-disk drive (HDD). The HDD industry is currently undergoing a transition from today's standard storage technologies to one that will handle the growing need for compact data storage in the approaching years. Four major competing technologies have taken shape that are capable of overcoming the physical limitations imposed upon the perpendicular magnetic recording systems currently utilized by HDD manufacturers. These four innovative technologies include shingled write recording (SWR) in conjunction with two-dimensional magnetic recording (TDMR); heat-assisted magnetic recording (HAMR); microwave-assisted magnetic recording (MAMR); and bit-patterned magnetic recording (BPMR). While all of these possibilities provide an avenue for growth, they also introduce their own unique problems embedded within the technology, making it difficult to determine which one or ones will spawn the next generation of magnetic storage for HDDs. Because the path leading to greater achievable storage densities is still unclear, many companies, organizations and universities are involved in extensive research initiatives that are attempting to push each possible technology out into the forefront ahead of other competing

technologies. Regardless of the final outcome of this partially collaborative effort, the development of the next generation of HDDs will take place as a migration from the current drive technology. This is not too surprising as only BPMR represents a truly radical jump from current perpendicular magnetic recording practice. While this dissertation will mainly explore cutting-edge advancements in bit-patterned magnetic recording, it is still instructive to highlight each technology in turn as well as to revisit some of the crucial elements of standard perpendicular recording in order to better understand the subsequent investigation into specifics of BPMR.

1.1 Current State of HDD Magnetic Recording

Perpendicular magnetic recording (PMR) on continuous media is the present cornerstone of the HDD industry and various elements of its functionality are embedded within each of its four possible successors. In PMR the write and read heads are always positioned in close proximity to the magnetic medium fabricated as a set of platters within a HDD. Each platter consists of a continuous field of perpendicularly magnetized ferromagnetic material that is polarized by the write field imposed by the main pole of the write head at the surface interface of the ferromagnetic layer [1]. The write field effectively organizes the disk into small groupings of magnetic grains where the magnetization is perpendicular to the direction of travel for the write head. Various groupings of these magnetic grains tend to sustain a common magnetic field pointing in either an upward or downward direction with respect to the surface of the platter and are referred to as magnetic domains (see Figure 1.1).

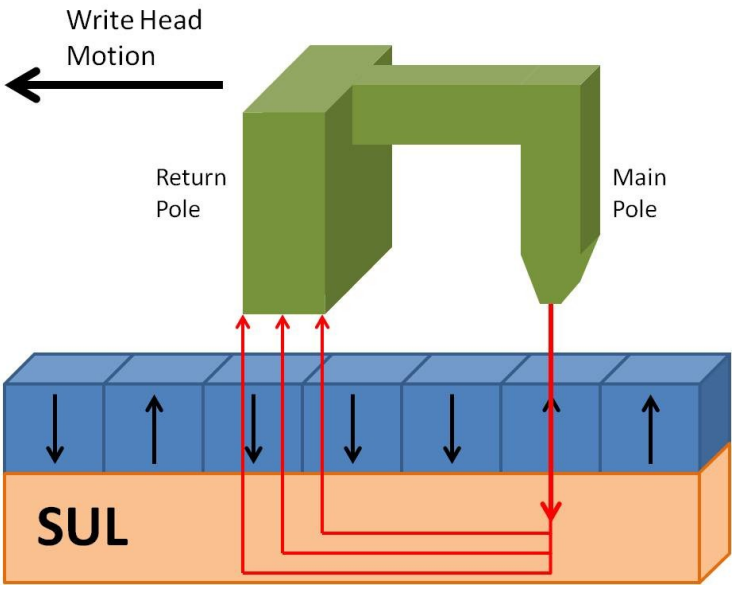

Figure 1.1 – A standard write head performing the write operation in perpendicular magnetic recording with an SUL present.

These small magnetic domains comprised of individual magnetic grains are what represent individual bits from the input bit stream and are commonly referred to as bit cells. Within the media zeros and ones are differentiated by domain magnetizations of opposite direction. To switch a bit a write field is initiated at the main pole which travels down through the magnetic layer and into a magnetically soft under layer (SUL). The SUL's purpose is to sufficiently diffuse the write field strength before the write field flux reenters the magnetic layer on its path back to the return pole of the write head. This prevents the bits underneath the return pole from being rewritten. In order to recover the data during the reading process, the read head picks up the magnetic field within the medium and outputs a continuous signal tuned to the "change in flux" from the magnetized bit cells. This means the readback signal is sensitized to the transitions where the direction of the

magnetization flips. Thus changes in the readback signal coincide with either (0 → 1) or (1 → 0) transitions between bit cells in the recorded bit stream. Normally within magnetic recording channels a recorded bit stream actually represents an input bipolar sequence $x_n \in \{-1, +1\}$ of length n, which is just a mapping of the original 0's and 1's. When transitions occur underneath the read head, a hyperbolic tangent function can be used to model the single isolated signal transition response $s(t)$ of the channel [2],

$$s(t) = V_p tanh\left(\frac{ln3}{T_{50}}t\right), \tag{1.1}$$

where the parameter V_p is related to the amplitude of the waveform. The other parameter, T_{50}, is constant for a given media and is defined as the time required for the isolated transition response $s(t)$ to go from $\left(-{}^{V_p}/_2\right)$ to $\left(+{}^{V_p}/_2\right)$. Since the parameter T_{50} is defined by the media, increasing the density of the recorded bits and thus the overall storage capacity of the drive comes at a cost. The density can be expressed as the ratio $\left(\frac{T_{50}}{T_b}\right)$ where T_b is the bit period, or equivalently, the time allocated for a single bit cell. Placing more bits within a given area by decreasing the size of the bit cells (decreasing T_b) effectively spreads the readback waveform across more bits as the isolated response requires a set time (T_{50}) to switch [3]. The sensitivity of the read head is adequate enough to sense multiple transitions when bit cells of opposite magnetizations are adjacent. This results in an increased level of unwanted signal power in the readback waveform known as intersymbol

interference (ISI). Its presence impacts the information available in the readback signal and is a significant problem for perpendicular magnetic recording. In addition to the noise introduced by the ISI, transitions in the magnetization between bit cells also degrades the readback signal with a substantial amount of media noise, which is typically referred to as jitter position noise.

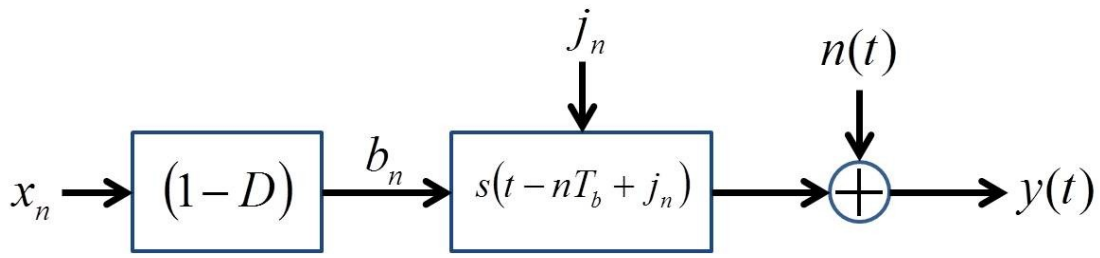

Figure 1.2 – Diagram of a perpendicular magnetic recording channel with jitter noise and AWGN.

The ever-present thermal noise within the electronic system is modeled as additive white Gaussian noise (AWGN) and will degrade the output of the storage channel as well. Other degrading factors are present within a HDD utilizing perpendicular recording on a continuous media, but the ISI, position jitter and AWGN are the main impediments considered here. Since the read head is sensitive to transitions in the magnetic states stored within the medium, the bipolar channel inputs x_n are mapped to a transition sequence using a $(1 - D)$ delay filter. The $(1 - D)$ outputs, denoted b_n, are then representative of both the transitions in x_n and the direction of transitions, $(-1 \rightarrow +1)$ and vice versa, with the definition $b_n = x_n - x_{n-1}$. Incorporating these considerations with the isolated transition response (see Figure 1.2) allows the storage channel output to be modeled as

$$y(t) = \sum_{n=-\infty}^{\infty} b_n s(t - nT_b + j_n) + n(t), \tag{1.2}$$

where $y(t)$ is the channel output, b_n is the transition sequence resulting from the $(1 - D)$ versions of the original bipolar input sequence, s is the isolated transition response, T_b is the previously mentioned bit period, j_n is the jitter noise samples which are assumed to be independent and Gaussian with variance σ_j^2, and $n(t)$ is the AWGN with single-sided power spectral density N_0. Since the jitter noise term j_n is buried within the isolated response $s(t - nT_b + j_n)$, its overall contribution is masked, so it is actually more advantageous to rewrite (1.2) with the jitter noise term expressed outside of the channel response term by means of a Taylor series expansion in j_n [4] where

$$s(t - nT_b + j_n) = \\ s(t - nT_b) + j_n s'(t - nT_b) + \frac{{j_n}^2}{2} s''(t - nT_b) + \cdots \tag{1.3}$$

Assuming the jitter noise contribution to the overall channel output is small compared to s$(t - nT_b)$, a first-order approximation is adequate and the channel output becomes

$$y(t) \approx \left\{ \sum_{n=-\infty}^{\infty} b_n s(t - nT_b) \right\} \\ + \left\{ \sum_{n=-\infty}^{\infty} b_n j_n s'(t - nT_b) \right\} + n(t). \tag{1.4}$$

This overall channel response provides the output as the sum of separate contributions from the noiseless transition response, jitter noise and AWGN

$$y(t) \approx y_{\text{signal}}(t) + y_{\text{jitter}}(t) + y_{\text{AWGN}}(t)\,. \tag{1.5}$$

This method is very useful and will be employed in subsequent chapters in the context of jitter noise contributions within BPMR. Robust systems for signal processing necessarily follow the output of the channel in order to recover from the effects imposed by the channel and degrading noises. A development of signal processing techniques for BPMR will be covered later. At this point it is necessary to expand on the reasons why PMR on continuous media cannot provide the storage densities needed in the future.

1.2 Limiting Factors for Magnetic Storage Channels

While PMR can effectively handle the combined impairments of ISI, position jitter and AWGN, its current implementation cannot exceed certain physical limits. On continuous media the most important among PMR's detractions is the technology limiting encumbrance imposed by the *superparamagnetic effect*. When the physical size of ferromagnetic particles is drastically reduced, a point is reached where the ambient temperature of the medium alone can provide enough thermal excitation to flip the direction of the inherent magnetic field. Thus the medium has succumbed to the superparamagnetic effect [5] and is rendered unusable for reliable data storage. For magnetic storage channels, any technology must strike a balance between three physical limitations: the ability to write to the physical media, the

signal-to-noise ratio (SNR) supported by the medium, and the thermal stability of the media. Such a balance is often called the *media trilemma* [6] and its implications impose challenging restrictions on the design of HDDs. In order to achieve greater storage capacity within the same physical space using a given medium, the area associated with a given bit representation must be reduced, eventually reaching the limits of the medium. As such miniaturization occurs one or more aspects of the trilemma adversely restrains the process. In terms of the characteristics of a given media, both the magnetic anistrophy and its coercivity determine its part in balancing the media trilemma. The magnetic anistrophy represents the medium's ability for maintaining a given directional magnetic field in the absence of any external field. So a high anistrophic medium tends to hold its magnetization better than a medium with low anistrophy and would seem to be more desirable in magnetic recording. However, a high anistrophic medium also possesses a high coercivity, H_c, which is the resistance of the medium to being demagnetized. A high H_c implies that the material will hold its magnetization well and exhibit greater thermal stability against changes to its magnetic field. Unfortunately, H_c is equivalent to the write head field strength required to switch the direction of magnetization necessary when storing bits. A high head field requires a powerful and larger write head which then spreads its imposed magnetic flux across multiple bit cells in the medium. This give-and-take relationship is what establishes the need for a balance of the trilemma.

Overcoming the superparamagnetic effect is not possible with current PMR technology. The predicted storage limit for perpendicular magnetic recording on continuous media is estimated to be an areal density of 1 Tb/in^2 [7]. This density estimate is far short of what future systems will require. Current industry regimes are hoping to extend the areal density to around 10 Tb/in^2 [8] within the decade using new technologies. The four distinct possibilities of SWR/TDMR, HAMR, MAMR and BPMR have all been established to meet such an impressive goal by attacking the balancing act set forth by the media trilemma in their own unique way.

1.3 Shingled Write Recording with TDMR

Of the four leading technologies SWR is considered by many to be the most straight forward migration from the current methods and hardware. Its marriage with TDMR allows for a possible increase in areal density up to 10 Tb/in^2 [8]. For this technology option the writing process consists strictly of shingled write recording where the local size of the write head is greatly expanded to allow multiple tracks to be written simultaneously. As illustrated in Figure 1.3 these tracks overlap much like shingles on a roof. By overwriting adjacent tracks in subsequent writes, a very narrow write margin can be achieved. This is possible because the expanded size of the write head allows for the generation of a much stronger write field. Thus the trilemma is handled by coupling a larger and more powerful head field with a higher anistrophic medium. The sharp corner edge field produced by the write head also results in a much narrower erase band between tracks, allowing individual tracks to lay closer together [9]. The size of an individual track of data can be squeezed down by

reducing the size of the individual magnetic grains or the total number of grains required to effectively store a bit of information.

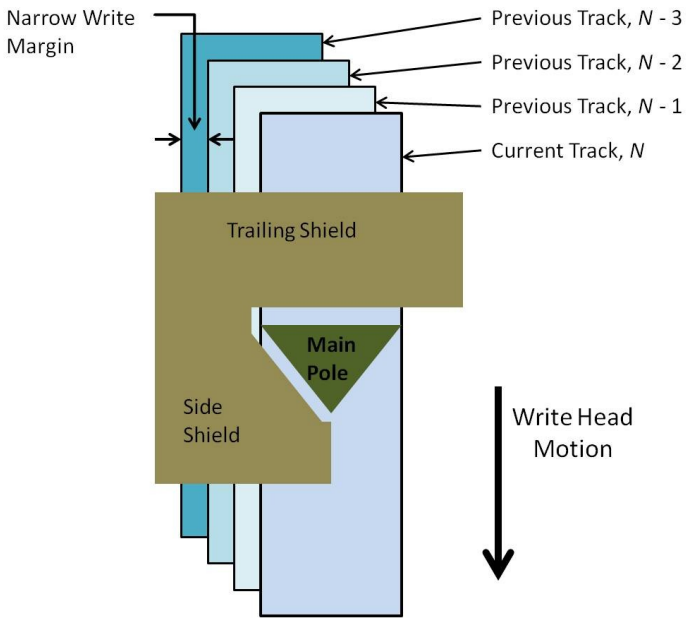

Figure 1.3 – Write head arrangement for shingled write recording where adjacent tracks are largely overwritten in subsequent passes.

However, SWR coupled with conventional reading techniques is estimated at being able to achieve only up to 2-3 Tb/in^2 [10]. This results because higher densities can only be obtained by shrinking the write margin, which places adjacent tracks close enough together for multiple tracks to influence the readback signal produced by conventional read heads. Noise from adjacent tracks results in intertrack interference (ITI) which will be an important consideration in the future of many HDD designs. Thus advances on the read side of the process must also be incorporated to reach an areal density goal of around 10 Tb/in^2. TDMR makes this feasible by incorporating innovative 2D reading. A 2D architecture typically senses multiple tracks and then processes the two-dimensional information using advanced

signal processing algorithms to reconstruct the 2D waveform and then recover the recorded bit stream. A two-dimensional approach attempts to turn the ITI present in the readback signal into usable information for detection and decoding algorithms [11]. Thus the trilemma is approached from the read side as well by focusing on maintaining appropriate SNR in a high noise environment. For continuous media performing TDMR is a nontrivial task that requires knowledge of the bit cell boundaries which are commonly modeled as Voronoi regions [12] (see Figure 1.4). A grid coincident with individual bit cell sizes is then overlapped on the Voronoi regions to map the various magnetic centers to an array of individual bit cells. This results in sets of tracks where the bit cells either contain a perpendicular magnetization coincident with a "0" or "1" or they are empty. The empty bit cells are considered erasures so that the resulting channel output can be characterized as a binary erasure channel (BEC) with three transition probabilities p, e and $1 - p - e$.

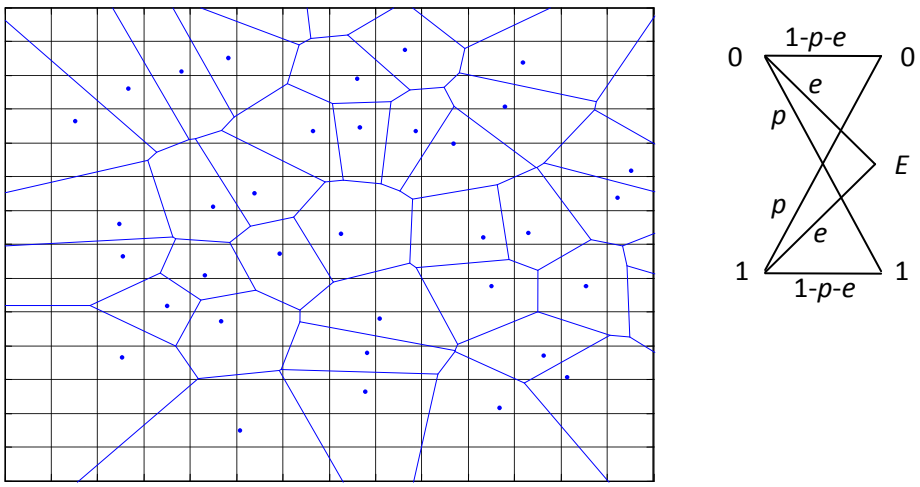

Figure 1.4 – Geometric layout of the bit cells on a continuous medium used to model the readback signal in TDMR along with its BEC trellis representation.

Conceivable architectures for TDMR include designs with multiple heads or a single head capable of progressive scans of adjacent tracks. Regardless of the final designs, 2D signal processing inherently includes greater complications, but it will almost certainly find its way eventually into all future HDD designs. The major disadvantages of SWR/TDMR are the loss of ability to update a single track during the write process [13] and the additional burden associated with the sophistication of 2D hardware and signal processing algorithms [14], [15]. Some advanced two-dimensional techniques will be explored later in this dissertation in light of bit-patterned magnetic recording.

1.4 Heat-Assisted Magnetic Recording

HAMR is an energy-assisted technology where heat generated by a focused laser influences the write head's ability to affect changes in the medium's inherent magnetization. When heat is engineered as an element of the write process, it is typically necessary in order to facilitate switching in high anisoptrophic magnetic fields. The technique, illustrated in Figure 1.5, works by heating a small region of the medium where the write head needs to impose a switch of the local magnetization held by the grains. This heating acts to reduce the coercive force, H_c, of the grains allowing them to be magnetized in the desired direction. During the remagnetization the heated region is rapidly cooled during the application of the write field, which still determines the direction of domain magnetization used to distinguish the individual bits.

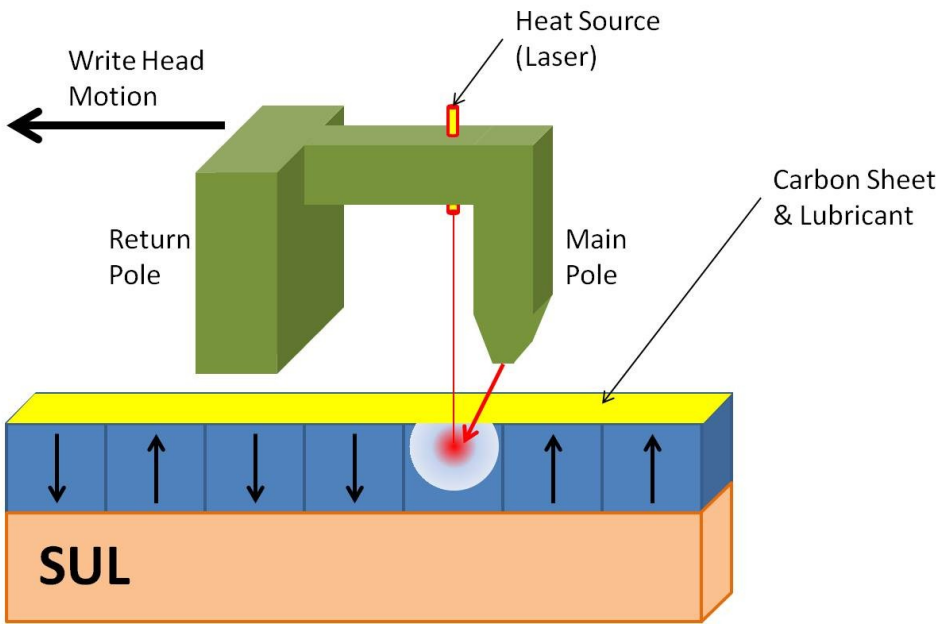

Figure 1.5 – A write head modified for HAMR by adding a laser heat source to assist in writing a single bit within a high anistrophic medium.

For this method to be effective the modeling of a HAMR drive must incorporate various influences of heat on the interaction of the magnetic fields. To better understand how the governing processes work, consider the following expression which describes the magnetization $M(x)$ of a single magnetic transition coincident with the writing of a bit [16],

$$M(x) = \frac{2M_r(T(x))}{\pi} tan^{-1}\left(\frac{x - x_0}{a}\right), \tag{1.6}$$

where $M_r(T(x))$ is the temperature dependent remnant magnetization retained by the medium, x_0 is the transition location, and a is the transition parameter. The thermal Williams-Comstock model, which was modified from the original Williams-Comstock model to specifically include temperature dependencies within conventional magnetic recording systems, is used to determine the crucial values of

x_0 and a for a given HAMR setup [17]. This is accomplished by iteratively solving the following set of equations which relates the unknown parameters (the parameters are actually buried inside expressions for the individual magnetic fields) to the changes within the interacting magnetic fields:

$$H_c(x) = H_h(x) + H_d(x), \tag{1.7}$$

$$\frac{dM(x)}{dx} = \frac{dM(x)}{dH_c(x)}\left[\frac{dH_h(x)}{dx} + \frac{dH_d(x)}{dx} - \frac{dH_c(T)}{dT}\frac{dT}{dx}\right]. \tag{1.8}$$

$H_c(x)$ is the medium's coercivity, $H_h(x)$ is the head field, and $H_d(x)$ is the demagnetizing field. Furthermore, all of these fields are also subject to various medium specifications and write head design parameters [18]. Normally the magnetization gradient $\frac{dM(x)}{dx}$ of (1.8) is not highly dependent on the thermal gradient of the medium's coercivity, given by $\frac{dH_c(T)}{dT}$, but with a HAMR design the local operating temperature is well above ambient temperature. The overall goal of this process is to locate where along the magnetization gradient curve the addition of heat energy allows the write head design in question to switch the direction of the medium's magnetic field. Equation (1.7) identifies the transition location x_0 as the point where the medium's coercivity is matched by the combination of head field and demagnetizing field. Initially, a value for the transition parameter a is used within (1.7) to obtain a value for the transition center x_0. This value is then substituted into (1.8), the expression is set equal to the derivative of (1.6), and the

system solved for a new value of the transition parameter a. The process repeats until the solutions converge. This identifies the true operating transition center for the HAMR drive and thus the design's overall usefulness.

Using heat in the magnetization process of a HDD introduces some notable challenges [19], [20]. The thermal properties of the medium in addition to its magnetization characteristics must be designed to be adequately synergistic. Given the number of average writes that take place throughout the lifespan of a HDD, the use of heat degrades both the lubricant and the carbon overcoat of the platters, so higher quality replacements for conventional materials must be found. Perhaps the greatest challenge for HAMR may prove to be the mounting of a laser diode (the heating source) to a HDD write head. Regardless of the difficulties involved, HAMR may have both current and future applicability to the HDD industry. One possible future for a HAMR design would be an HDD that utilizes heat assistance to write on a high anisoptrophy BPMR island to push the bit density beyond what HAMR alone is capable. On continuous media HAMR has currently achieved an areal capacity of only 1 Tb/in^2 [21], but when combined with BPMR, estimates indicate plausible increases up to 9 Tb/in^2 [22].

1.5 Microwave-Assisted Magnetic Recording

MAMR is another form of energy-assisted writing technology applicable to HDDs. While HAMR delivers energy directly to the lattice structure of the medium, MAMR uses microwave energy to assist in directly placing the correct magnetization within

the grains comprising an individual bit cell [23]. The underlying physics of the write process are best explained in light of the Landau-Lifshitz-Gilbert (LLG) equation, which is often used when modeling the precessional motion of magnetic fields within ferromagnetic solids [24],

$$\frac{\partial m}{\partial t} = -\gamma m \times \left(H_{eff} + H_{ac}\right) + \alpha m \times \frac{\partial m}{\partial t}. \tag{1.9}$$

The LLG equation basically expresses the magnetic change as the combination of two terms, a torque [$-\gamma m \times (H_{eff} + H_{ac})$] and a damping term [$\alpha m \times \frac{\partial m}{\partial t}$]. In this equation γ is the electron gyromagnetic ratio, H_{eff} is the effective magnetic field which is equal to the difference between the media's inherent magnetic field (H_K) and the field imposed by the write head (H_{write}), H_{ac} is the assistive magnetic field produced by the harmonized microwaves, and α is the Gilbert damping constant. With the help of Figure 1.6 and the rewritten LLG equation,

$$\frac{\partial m}{\partial t} = -\gamma m \times (H_K - H_{write} + H_{ac}) + \alpha m \times \frac{\partial m}{\partial t}, \tag{1.10}$$

the role of microwaves in assisting the write process is more easily visualized. Initially the magnitude of the write field is less than what is required to switch the embedded magnetization state of the grains in a bit cell of the high anistrophic media. So a targeted burst of microwaves is supplied which produces its own assistive magnetic field, H_{ac}, orthogonal to the plane of the switching field. The result is a precessional field capable of switching the direction of the magnetic field

in the bit cell [25]. The precessional field can be either clockwise (CW) or counterclockwise (CCW) depending on the direction of the assistive field.

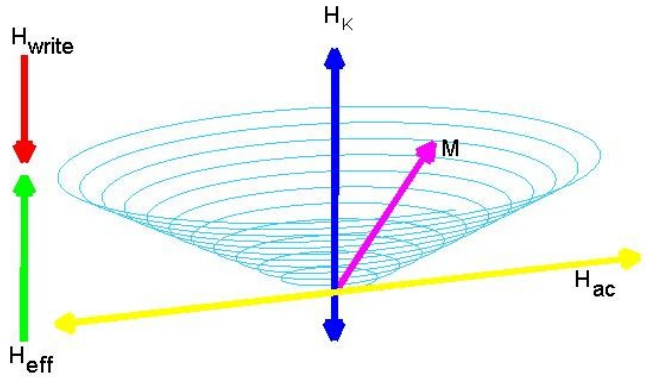

Figure 1.6 – Vector representation of the magnetic fields interacting under the influence of a MAMR write head in order to switch the direction of the current magnetic domain.

A clockwise procession results in a downward magnetization and a CCW precession produces the opposite upward magnetization. The combination of a high anistropic medium coupled with a reasonable switching field makes it possible to shrink the size of the bit cells to achieve higher densities and thus manages to balance the media trilemma. One possible representation of a MAMR write head is illustrated in Figure 1.7 where the field generating layer (FGL) is responsible for producing the assistive microwaves. However, the problems that come with MAMR make it an unviable technology for targeting 10 Tb/in^2 areal density without coupling it with BPMR [26]. One of the main challenges lies in maintaining a workable balance between the flying height of the write head and the strength of the assistive field.

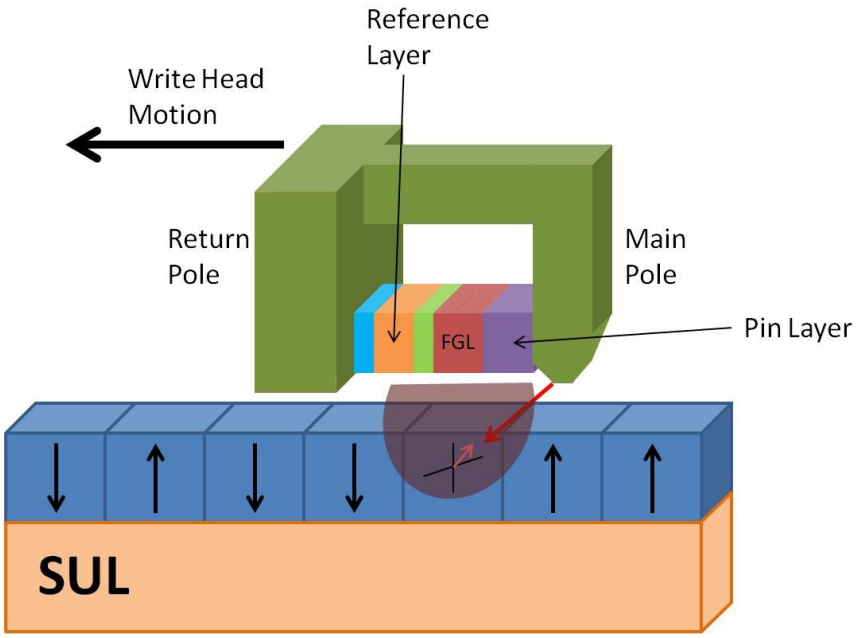

Figure 1.7 – One possible rendition of a MAMR write head where the microwave field assists in modifying the magnetization of a bit cell.

Since the strength of the assistive field decays rapidly with distance from the recording location, the height at which the write head passes above the medium must be kept as small as possible. Also, an innovative microwave oscillator coupled with a magnetic writing head which meets all design specifications still needs to be engineered before MAMR can move to the forefront ahead of its competitors or be realized in a high density combination with BPMR.

1.6 Bit-Patterned Magnetic Recording

The particulars of bit-patterned magnetic recording break from the traditional mold of PMR specifications significantly. No longer is the medium continuous, but instead is a manufactured plane of individual islands standing in a sea of non-magnetic material as illustrated in Figure 1-8. As an HDD technology, BPMR makes use of

conventional write and read heads but requires a radically different media. In BPMR each island represents a single bit recorded on yet a single magnetic grain. This is possible because each island is lithographically manufactured as a singular magnetic domain residing within a single grain [27].

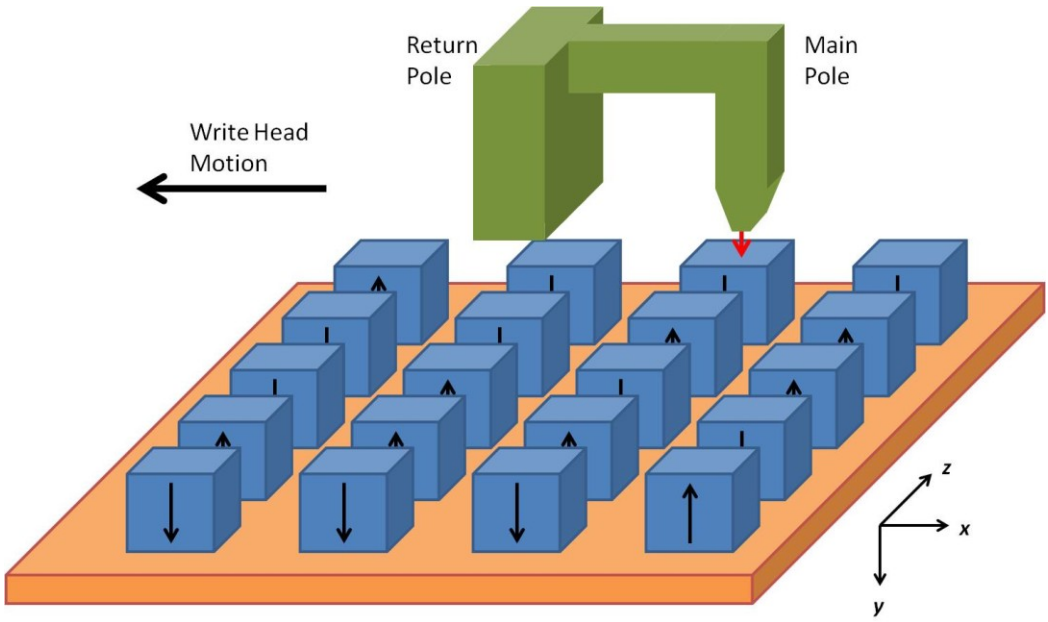

Figure 1.8 – An array of individual islands as they would appear at a local scale on the surface of a platter within a HDD.

Because lithography is utilized, it is also possible to manufacture islands comprised of individual grains of varying shapes and sizes. Square, rectangular, circular and elliptical island geometries have all been proposed [28], [29]. The islands can also be generated so that adjacent tracks follow either a rectangular or hexagonal pattern. With such variations at hand, a target areal density is reached by manipulating the island shape and size in conjunction with the distance between adjacent islands. However, these manipulations are not unconstrained because the size and shape of the islands affect the design space of the read and write heads as well as the various

interactions of the magnetic fields during both the writing and reading processes. Islands spaced closer together will naturally experience greater interference from its neighbors.

1.6.1 Bit Aspect Ratio and Grain Volume

A typical BPMR arrangement of islands in two dimensions is shown in Figure 1.9 where the islands are arranged in a rectangular array. The size of each island is denoted L_x and L_z where x and z coincide with the along-track and cross-track directions. The center of each island is a set distance apart in each direction. T_x is the bit period and represents the distance between individual island centers in the along-track direction. T_z is called the track pitch and represents the same measure as T_x but in the cross-track direction. The islands occupy a given area of $L_x \times L_z$ nm^2 and in a perfect world would exert a magnetic influence only within the boundary defined by $T_x \times T_z$ where the island is located at the center of that locale.

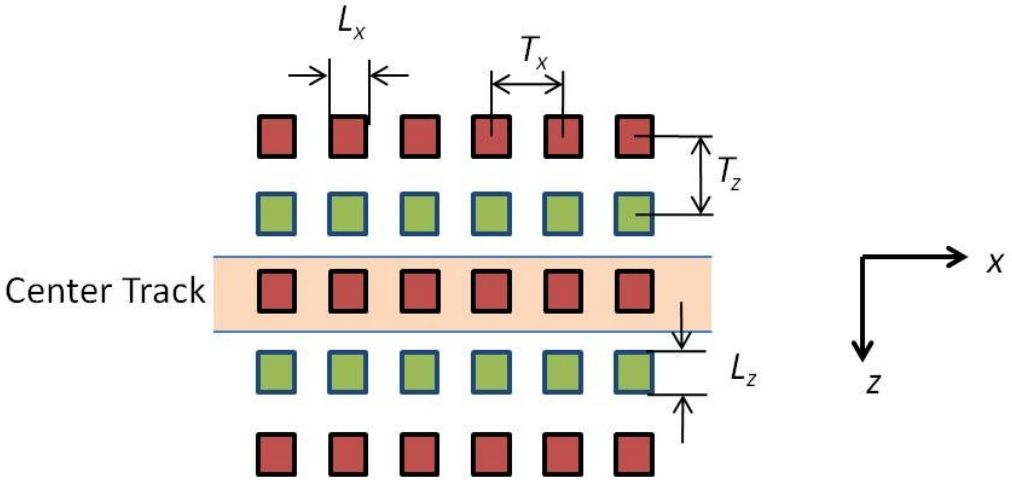

Figure 1.9 – Two-dimensional island layout of five adjacent tracks in BPMR with square island geometry.

It should be noted that when the islands are centered within their respective tracks, the gap distance between neighboring islands in the along-track and cross-track directions is equal to L_x and L_z respectively, and the distance between adjacent island centers is equal to T_x and T_z respectively. For square islands the quantities L_x and L_z are equal and so are T_x and T_z. In BPMR the bit aspect ratio (BAR) is a measure of the relative size of the island and is defined as the ratio of the cross-track island size to the along-track island size,

$$\mathrm{BAR} = \frac{L_z}{L_x}. \tag{1.11}$$

Thus, for circular and square islands the BAR is equal to 1. For rectangular or elliptical islands the BAR is greater than 1, and the longest side of an individual island is typically placed in the cross-track direction. This definition of BAR is not appropriate for continuous media since its magnetic domains are comprised of multiple grains. The bit aspect ratio in BPMR plays an important role in both the fabrication of the medium and the design of the write and read heads. In conventional PMR the BAR is usually determined by the design specifications of the write and read head since multiple grains are magnetized within a single bit domain. In contrast BPMR island size is determined by the fabrication process, which strongly favors a BAR of 1 [30], [31]. This would not at first appear to be a problem except that the design of conventional recording heads, which are used with BPMR, favors larger BARs. This is because larger BARs allow for a larger write and read head which in turn allows for stronger write head fields with steeper magnetic gradients and

better SNR levels for sensing by the read head. Larger heads are also easier to design and fabricate. The tradeoffs associated with the bit aspect ratio become even more complicated when fabrication tolerances are considered. Less stringent fabrication results in larger variations in island size (L_x and L_z), island position (T_x and T_z) and island amplitude which all degrade system performance [32], [33].

In light of such competing design issues, one might wonder why proceed with BPMR at all. One of the compelling reasons BPMR was proposed is because it offers thermal stability for the media coupled with a scalable path to higher areal densities [34], [35]. Thus BPMR circumvents the media trilemma by utilizing increased media thermal stability, which is accomplished by increasing the grain volume beyond what conventional continuous media can support. While this might at first seem counterintuitive since increasing areal density requires the scaling down of the island size, the volume associated with a given island and thus a given grain is actually larger for BPMR than for the grain sizes used in continuous media [36]. For a given magnetic grain, the quantity $K_A V$ represents the energy that can be stored in a single grain of volume V with magnetic anistrophy energy density K_A. For a grain to be useful for magnetic storage, it must successfully compete with the thermal energy $k_B T$ of the medium, where k_B is Boltzmann's constant and T is the operating temperature in Kelvin. In BPMR the size of the island dictates the grain volume. When this is reduced, requirements for thermal stability are normally still satisfied, so K_A does not have to be increased unless the current anistrophy results

in $K_A V < k_B T$. Thus, BPMR can take advantage of media with grain materials that possess conventional levels of anistrophy.

1.6.2 Writing to BPMR Islands

As areal densities are pushed forward, the scaling down of island sizes places ever increasing constraints on the design. In both continuous and bit-patterned media the write function can be thought of as a two-step process which involves (1) positioning the write head at the correct location and then (2) effectively manipulating the embedded field within the targeted bit domain to flip its magnetic state. BPMR greatly differs from conventional recording in its constraints for step (1). In order to successfully switch magnetizations of individual islands, the write head must maintain synchronous positioning over the islands within its targeted track or successful writing cannot occur. This stringent requirement is not upheld for continuous media HDDs as the write process is asynchronous. While an asynchronous write process is possible for BPMR, it has been shown to induce too high of a penalty on the channel capacity for BPMR to even be considered for future HDDs [37]. Fortunately for the synchronous case the technology is still viable for future HDD design in terms of limits on the channel capacity [38], [39].

But this still brings to bear the problem of advantageously placing the write head over a given island. Current servo tolerances, mechanical vibrations, spindle speed variations and various other physical factors limit the ability to place the write head exactly where it needs to be during the write process. In a simplified sense the

collective adversities to write head placement can be represented as a synchronizing jitter, σ_s, that acts to degrade the writing process. With regard to the previously mentioned second step of the writing process involving write field, once the write head is in place, the interactions of magnetic fields determine whether or not a bit is actually written. In BPMR no assistive magnetic fields are present so the imposed head field must overcome the island's inherent resistive field as well as any resistive fields induced by the nearest neighbors of the targeted island. The opposing forces acting against the head field are referred to as the switching field distribution (SFD) and its contribution can be denoted with an associated standard deviation of σ_{SFD}. The write process is also affected by the fabrication jitter, σ_j, which causes variations in the interactions of the magnetic fields. Using an approach analogous to [40], the design space for BPMR from the write perspective can be explored. In terms of an upper bound, the write ability for BPMR islands can be expressed as a bit error rate, BER_W, subject to the following constraint

$$T_x \geq 2u \, . \tag{1.12}$$

Here the quantity T_x refers to the bit period and $2u$ is the one-dimensional size of the region required by a centered island in order to achieve the upper bound on BER_W. The quantity u, referred to as the uncertainty zone in [40] and [41], can be defined as

$$u = \beta_w \sqrt{\left(\frac{\sigma_{SFD}}{\frac{\partial H_{eff}}{\partial x}} \right)^2 + \sigma_s^2 + \sigma_j^2}\,, \tag{1.13}$$

where σ_{SFD}^2 is the variance of the switching field distribution, σ_j^2 is the variance of the fabrication jitter, σ_s^2 is the variance of the placement of the write head, and $\frac{\partial H_{eff}}{\partial x}$ is the gradient of the effective write field. All variances are considered to be independent and Gaussian distributed. The parameter β_w simply links the uncertainty zone to the targeted upper bound on bit error rate of write ability:

$$\mathrm{BER}_W = \frac{1}{2} erfc\left(\frac{\beta_w}{\sqrt{2}}\right). \tag{1.14}$$

Equations (1.12 – 1.14) are only approximate as they ignore a number of justifiable interactions and degradations as well as require the effective write field to be closely linear within the operable range of the applied write field. However, the linear slope assumption is not a bad model, and the total approximations for the current scope are quite reasonable. If T_x is chosen as the upper bound for equation (1.12) then the achievable areal density, A_2, will be given by

$$A_2 = \frac{A_0}{T_x^2 \times \mathrm{BAR}}, \tag{1.15}$$

where A_0 is a constant and BAR refers to the bit aspect ratio. With T_x in nanometers the constant A_0 = 645.16 for an areal density in Tb/in^2, and equation (1.15) can be used for some insightful calculations:

$$\frac{2\text{Tb}}{\text{in}^2} = \frac{A_0}{(17.96)^2(\text{BAR}=1)} \qquad \frac{4\text{Tb}}{\text{in}^2} = \frac{A_0}{(12.7)^2(\text{BAR}=1)}$$
$$\frac{2\text{Tb}}{\text{in}^2} = \frac{A_0}{(12.7)^2(\text{BAR}=2)} \qquad \frac{4\text{Tb}}{\text{in}^2} = \frac{A_0}{(8.98)^2(\text{BAR}=2)} \quad (1.16)$$

As the areal density is doubled from 2 to 4 Tb/in^2 at a BAR of 1, the size of the bit period must be reduced from 17.96 nm to 12.7 nm. But when the BAR is 2, the increase in density from 2 to 4 Tb/in^2 still requires a substantial decrease in bit period from 12.7 nm to 8.98 nm. Thus, ISI will remain a problem for all scaled island sizes in the along-track direction regardless of the bit aspect ratio. Since the islands are uniformly distributed, the ITI is also expected to remain if islands are scaled in the cross-track direction.

Unfortunately the analysis from equations (1.12 – 1.16) does not reveal the most troublesome part of write errors. The synchronizing jitter, σ_s, can actually lead to a compounding problem. As the write head proceeds from the start of a sector, it will procure some form of synchronization information so that it begins the sector more or less on target with the first island. From that point the write head will continue its write function until reaching the end of the sector, which could be either around 4K bits later or 32K bits later if long sectors are employed [42]. Given that the write clock phase offset can accumulate with every single positioning of the head over a target island, the total write offset can exceed half a bit period. This places the write head over the wrong island in what is known as an insertion/deletion condition. If this error is not corrected by resynchronizing the

write clock, the write head will continue to wrongly address islands until the end of the sector is reached. The result can be a sector with a multitude of errors. Since these errors occur at the writing stage, they are not seen by the channel detection as anything other than correctly written bits. This leaves the cleanup of such a mess solely to the decoding, which means the sector information may not be recoverable unless some form of synchronizing correction occurs within the sector [43]. If the written-in errors can be limited to only single or short runs of substitution errors without incurring insertion/deletion errors, then sophisticated detection algorithms can assist in correcting the written-in errors [44].

1.6.3 Reading from BPMR Islands

Because individual islands are separated by physical free space where theoretically the read head would register significantly less signal power, the problematic transition noise between adjacent bits of opposite magnetization within a continuous media does not exist. So the transitional jitter noise present in all of the other competing technologies does not plague BPMR. However, BPMR is not without its own significant engineering challenges for the read head including contributions to media noise specific only to bit-patterning technology. These noise contributions are the result of patterning tolerances from the governing fabrication process. Thus, bit-patterning does not result in a media with perfectly shaped islands, orthogonal island distributions or homogeneous magnetic properties. In BPMR the media noise affecting the performance of the write head will also negatively impact the performance of the read head. The read back process will pick

up multiple variations in the media which include island height and size variations, island positions that vary in both along-track and cross-track position, and variations in the inherent magnetizations of individual islands. Figure 1.10 provides an example of what some of these individual media jitter effects would look like. Under the influence of a position jitter (ΔT_x or ΔT_z) the island shape remains unchanged but the island itself is displaced from the center of the bit cell. When the jitter affects the size of an island (ΔL_x or ΔL_z) the island becomes a rectangle that is either elongated or shortened in the direction of the jitter. In a true channel, various combinations of these jitters are possible and can result in significant degradations on the write and read processes.

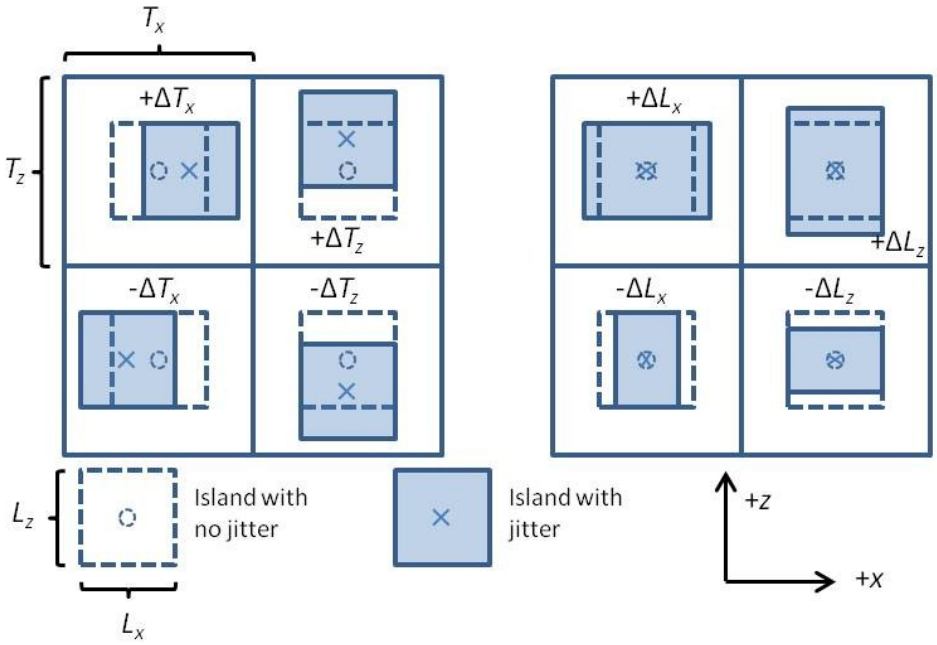

Figure 1.10 – BPMR array of islands with various types of fabrication media noise. The dashed boxes show the size and positions of the islands without media noise. The solid boxes illustrate what the position or shape would be under the appropriate jitter influence.

For example a larger than average island ($L_x + \Delta L_x$; $L_z + \Delta L_z$; or $L_x + \Delta L_x$ and $L_z + \Delta L_z$) displaced slightly along the positive *z*-axis ($T_z + \Delta T_z$) would produce an amplified readback signal for itself and exacerbate the ITI for its neighbor in the positive *z*-direction. In [45] variations in media jitter were shown to have a substantial negative effect on the read channel performance in terms of BER for BPMR while island shape had none. Thus, the choice of square or circular islands does not noticeably affect channel performance so either geometry can be used to model the individual bit cells.

1.7 Dissertation Overview

The preceding discussions help reveal the extent of many of the design challenges facing BPMR as one of four future candidates for HDDs. It faces significant obstacles on both the write and read side of the design. And since the media itself is not continuous, the write head will be required to maintain a synchronous position close to the center of each and every island, which is a daunting challenge for any areal density goal above 1 Tb/in^2. Solving this problem alone is paramount to the success of BPMR; for without it, the number of errors introduced into the readback signal by a loss of write synchronization can easily exceed the recovery threshold of any known combinations of detection and error correction. Despite these daunting challenges, many in the HDD industry are striving to develop the technology. This is understandable because even if one of the aforementioned promising contenders establishes itself as the successor to perpendicular magnetic recording, bit-patterned magnetic recording is still likely to be embedded within the future

magnetic recording technology achieving an areal density of 10 Tb/in^2 or more. While many resources have come to light over the past decade regarding BPMR, few have honestly attempted to extend the areal density beyond 1 Tb/in^2 in any practical sense. Most notable is the work of Chang and Cruz [46] and Cai *et al.* [47] who have addressed some of the difficult signal processing problems facing BPMR with varying degrees of success. Thus the main topic of this dissertation will focus on BPMR, and more specifically, on the signal processing aspects and associated models coincident with a future transition to this form of HDD. These efforts are both necessary and timely as the costs associated with producing a new media are high, so confidence must be established in signal processing techniques that can manage their share of the problems before investing in expensive retooling facilities.

In the second chapter of this dissertation, models of both the read and write process will be developed that include the most relevant aspects of the strengths and weaknesses of a BPMR design. The write model will encompass the three major types of written-in errors: substitution, insertion and deletion errors. Chapter 3 will then address a system model for the purposes of detection based on the channel output. Since the channel is characterized by significant levels of both ISI and ITI, TDMR detection techniques will be explored to determine their usefulness in dealing with elevated levels of this kind of interference. At the detection stage the written-in errors are decoupled from the system model since they occur outside of the detection process. In Chapter 4 error-correction coding will be added to the system model to help determine its possible role in dealing with the written-in errors in

combination with the ISI, ITI and media noise. Finally, in Chapter 5 the dissertation will be concluded and some relevant suggestions for future research will be expounded.

Chapter 2 - Modeling the BPMR Channel

As seen earlier with traditional perpendicular magnetic recording on continuous media, the key component to addressing the many problems faced by a BPMR hard disk drive resides in the establishment of a simulation model capable of capturing the essence of the proposed technology. This is never a simple task especially with a fledgling technology where many unknowns are certain to be encountered. While BPMR represents a substantial jump from conventional PMR systems, it will still utilize many of the standard components typical of PMR drives. This allows many of the modeling constructs and algorithms used in PMR to be extended for use within a BPMR context.

2.1 Elements of a Magnetic Recording System

In all of the magnetic recording systems considered here, the goal is to utilize the magnetic properties of a given medium to safely and reliably store as much user data as possible so as to meet the demands of the user. Both storage and retrieval must be relatively quick and the integrity of the data must be maintained across the lifetime of the drive. To accomplish this feat a HDD typically accepts the user data in the form of 1's and 0's, encodes this information in a variety of forms and then stores it on the physical platters. The data recovery is simply the reverse procedure subject to additional constraints because the physical write and read actions are not inverse processes. From both the write and read perspectives a number of

embedded processes are necessary. These processes are depicted within the simplified block diagram in Figure 2.1.

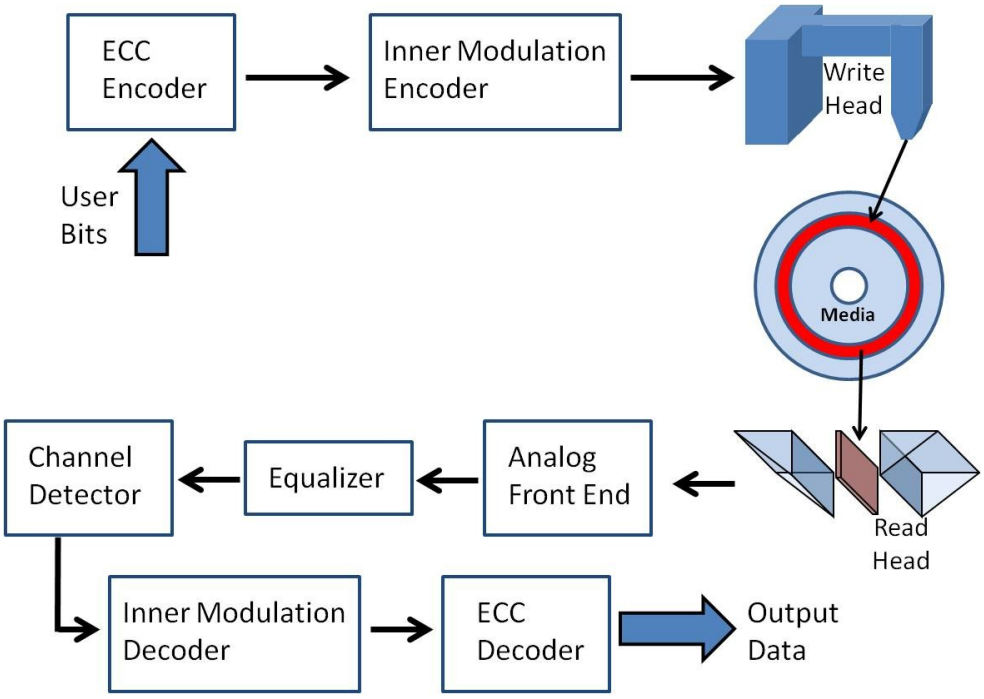

Figure 2.1 – Typical block diagram for a magnetic recording system.

User data {1's and 0's} is first processed by an error-correcting encoder where the data is mapped to an allowable codeword within the error-correcting code's set. Based on the redundancy of the particular ECC, the total number of bits to be written is expanded to accommodate the parity-check bits. The bits are then passed to a concatenated inner modulation encoder where the bits are once more remapped. While the purpose of the ECC is to assist in recovering message bits from a corrupted message, the modulation encoder mainly seeks to assist in timing recovery during the read process. Depending on the particulars of a chosen modulation code, the inner encoder may also attempt to enhance the distance properties of the message bits. If the message bits at the output of the modulation

encoder are in a non-return-to-zero (NRZ) form, then the message bits are passed to the write head and subsequently impressed on the HDD media. If not, then a precoder can be used to perform the appropriate mapping on the modulation output to achieve an NRZ format. In this work only polar binary inputs are assumed, which is an NRZ format with the basic mapping (0's and 1's) → (-1's and +1's). A true modulation encoder will not be included in the subsequent chapters as its presence does not enhance the discussions and only adds to the processing time for the channel simulations.

On the read side the recovery process begins with a sector scan by the read head to recover the stored information. The output of the read head is a continuous time signal that is passed to an analog front end where the continuous signal is prefiltered and then sampled at bit rate. This produces a finite power noisy sample of the readback signal which can be utilized by the equalizer. The equalizer attempts to recast the samples as a desired target response to within some predetermined limits on the equalizer error. One of the most effective equalizer designs seeks to minimize the mean-square-error of the equalizer output with respect to the desired target response. This type of equalizer will be developed in later sections with regard to two-dimensional ISI. Once equalization is completed the output is sent to the channel detector which is usually based on either the Viterbi algorithm (VA) [48], [49] or on the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [50]. When the detection is based on the VA, the output represents a hard decision on the bit sequence that coincides with the most-likely sequence given the detector inputs. Such hard

decisions would then dictate the usage of a hard-decision ECC as well. When soft decisions on each bit are desired, the BCJR algorithm represents the optimum *a posteriori* bit detector. Its outputs can be fashioned as log likelihood ratios (LLRs) for individual bits representing the probabilities that each bit is either a one or a zero. Even though its complexity is quite high, the BCJR algorithm is usually the detector of choice as its soft output can be handed off to a message-passing decoder. When a reduced complexity alternative is sought, the soft-output Viterbi algorithm (SOVA) can be used. It is an extended version of the VA where the output results in soft information on the individual bits [51]. When a modulation encoder is present in the write process, its bit manipulations must be undone on the reading side before proceeding to the decoder. The decoder makes an attempt to match the received sequence to a valid codeword which most likely matches the codeword produced by the encoder. In a hard decision decoder the input accepts the bit decisions of the detector. For a soft-decision decoder, LLRs from the detector are utilized as initial guesses on the individual bits before iterating on a decoding solution for the received codeword. If the decoder manages a successful decision on the bits, then it returns the output sequence that matches the sequence input at the encoder.

All of the components shown in Figure 2.1 are basically required in some form or fashion within a HDD. Thus, a viable simulation model for a magnetic channel must incorporate reasonable representations of an encoder/decoder, plausible filtering and sampling, an equalizer and a channel detector. In addition the synergistic compilation of all model components must be tailored to match the

characteristics of a read head pulse response, a channel response, and a realistic noise model. For BPMR a fairly representative system model must be constructed in order to substantiate its role as a prominent technology for future HDDs.

2.2 The Readback Signal

Determining the readback pulse response in magnetic recording systems is crucial for establishing any reasonable simulations. Unfortunately, the exact readback signal is only known by individual HDD manufacturers, so it becomes necessary to develop a model for the readback pulse response. As with most models, it is best if a pulse response can be formulated that captures the essence of what a true readback signal will most likely entail. For BPMR this amounts to capturing the characteristics of an individual island subject to the sensing process of a typical read head. To obtain a model of the readback waveform, it is assumed that the read head is located directly over the center of an island and is a specified distance away that is equal to the set fly height for the read head. In terms of all required specifications, the combination of a read head in the process of sensing an individual island would look like the image shown in Figure 2.2 where the island's image in the magnetically soft under layer is also shown. The presence of an SUL is somewhat questionable as it serves conflicting roles within PMR. An SUL has been shown to actually improve the writing performance in perpendicular recording by allowing for a larger write gradient through the magnetic layer [52], [53]. On the read side, however, the SUL introduces a negative impact by increasing the amplitude of the readback pulse, and with greater amplitude comes an undesirable broadening of the pulse [54]. The

development of BPMR is still too preliminary to ascertain whether or not the presence of an SUL is advantageous or not, so for now it is assumed as part of the medium.

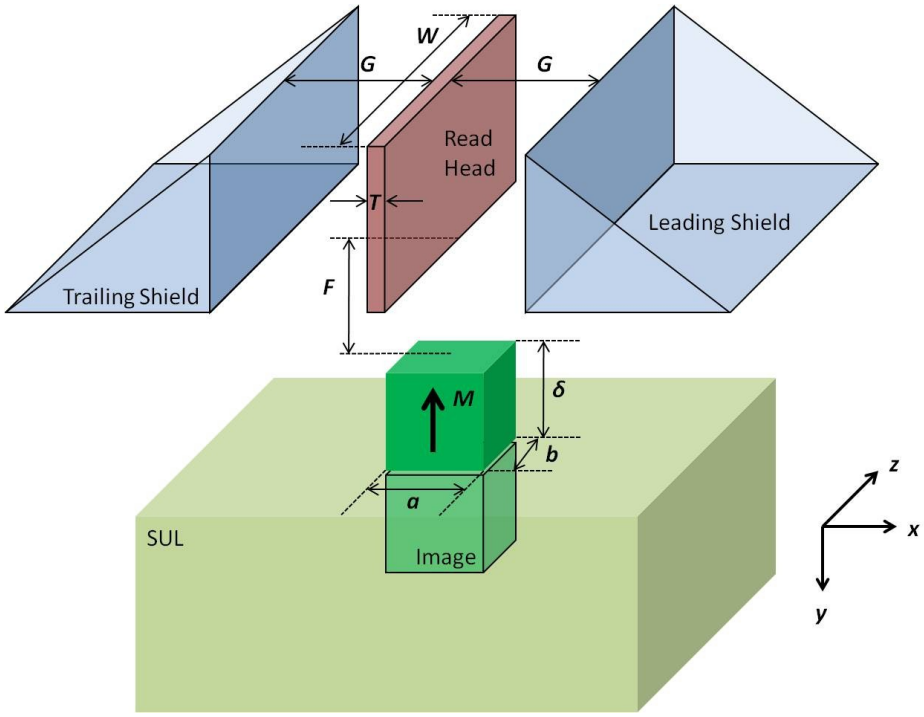

Figure 2.2 – Orientation of the read head in BPMR with the sensing element centered directly above a single magnetized island.

No intermediate layers are assumed between the SUL and the magnetic island. The island is assumed to have a rectangular shape with side lengths *a* and *b*. The height of the island is designated as delta (δ). It is also assumed that the island possesses perfect along-track and cross-track orientation. The island's perpendicular magnetization is considered uniform with magnitude *M*. The read head is comprised of a sensing element and side shields. The sensing element has thickness *T*, width *W*, and semi-infinite height. The shields are nonmagnetic and considered to have semi-infinite dimensions so as to perfectly shield the read head element from stray

magnetic flux. The read element is centered between the side shields and is a set distance *G*, called the gap length, from each shield. The read element's flying height above the island is denoted *F*. The read head functions by sensing the magnetization of an individual island and then responding with an output voltage. This output voltage is proportional to the magnetic flux injected into the read element at its air bearing surface (ABS),

$$V(x,z) = C\varphi(x,z). \tag{2.1}$$

By normalizing the amplitude of the readback waveform, the constant C can be ignored and the problem of arriving at a model for the readback signal reduces to modeling the output voltage as a two-dimensional magnetic flux. The flux is modeled in two dimensions because the individual islands will be close enough to experience interference in both the along-track and cross-track directions, which complicates the readback waveform considerably. Fortunately, much work has already been done with modeling two-dimensional magnetic flux. One practical method involves using the three-dimensional reciprocity integral [55] to analytically calculate two-dimensional flux,

$$\varphi(x,z) = \frac{\mu_0}{i}\int_{-\infty}^{\infty} d\tilde{x}\int_{F}^{F+\delta} d\tilde{y}\int_{-\infty}^{\infty} d\tilde{z}\,\big[H_y(\tilde{x},\tilde{y},\tilde{z}) \times M_y(\tilde{x}-x,\tilde{y},\tilde{z}-z)\big], \tag{2.2}$$

where μ_0 is the permeability of free space, i is the current residing within the imaginary coil of the read head, H_y is the magnetic field produced by the read head within its imaginary coil, and M_y is the perpendicular magnetization of the island.

The reciprocity integral can be expanded if a substitution is made for both magnetic fields H_y and M_y. Following [55], the magnetic fields can be represented by their respective gradients in magnetic potential,

$$\varphi(x,z) = \frac{\mu_0}{i}\int_{-\infty}^{\infty} d\tilde{x}\int_{F}^{F+\delta} d\tilde{y}\int_{-\infty}^{\infty} d\tilde{z}\left[\Psi(\tilde{x},\tilde{y},\tilde{z}) \times \frac{\partial M_y(\tilde{x}-x,\tilde{y},\tilde{z}-z)}{\partial \tilde{y}}\right]. \tag{2.3}$$

To continue deriving the readback response based on the specifications from Figure 2.2 and using (2.1), the three-dimensional gradient of magnetic potential of the read head and the magnetization of the medium must be known. Given the island is square (*a* = *b*) with uniform magnetization, then M_y is the only magnetization component present and it has a constant value of M confined within the boundaries of the island:

$$M_y = \begin{cases} M & \begin{cases} -\frac{a}{2} \le x,z \le \frac{a}{2} \\ \delta \le y \le \delta + F \end{cases} \\ 0 & \text{otherwise} \end{cases}. \tag{2.4}$$

Since the magnetization is simply a single square pulse, taking the derivative with respect to y results in two impulse functions at the boundaries of the island. Thus the magnetization of the island within the reciprocity integral is known. The magnetic potential, however, still needs to be obtained.

Since the magnetic flux of the read head system must flow into the SUL beneath the island (somewhat like the flux for the write head in perpendicular

recording illustrated in Figure 1.1), the SUL affects the magnetic flux and thus the corresponding magnetic potential. To incorporate such effects, the SUL is assumed to have semi-infinite dimensions coupled with infinite magnetic permeability. This allows the SUL's influence to be modeled as a perfect mirror image of the true read head across the SUL boundary. Thus the total magnetic potential is equal to the potential of the true read head minus the magnetic potential of the image read head, which is at a distance of $y = F + 2\delta$ from the top of the island. Based on the assumptions for the read head, SUL and island magnetization, the reciprocity integral can be simplified,

$$\varphi(x,z) = \frac{\mu_0}{i}\int_{-\infty}^{\infty} d\tilde{x}\int_{F}^{F+\delta} d\tilde{y}\int_{-\infty}^{\infty} d\tilde{z}\left[\Psi_{\text{Total}}(\tilde{x},\tilde{y},\tilde{z}) \times \frac{\partial M_y(\tilde{x}-x,\tilde{y},\tilde{z}-z)}{\partial \tilde{y}}\right], \tag{2.5}$$

$$\varphi(x,z) = \frac{\mu_0}{i}\int_{-\infty}^{\infty} d\tilde{x}\int_{F}^{F+\delta} d\tilde{y}\int_{-\infty}^{\infty} d\tilde{z}\left\{\left[\Psi(\tilde{x},\tilde{y},\tilde{z}) + \Psi_{\text{image}}(\tilde{x},\tilde{y},\tilde{z})\right]\frac{\partial M_y(\tilde{x}-x,\tilde{y},\tilde{z}-z)}{\partial \tilde{y}}\right\}, \tag{2.6}$$

$$\varphi(x,z) = C\int_{-\infty}^{\infty} d\tilde{x}\int_{-\infty}^{\infty} d\tilde{z}\left\{\left[\Psi(\tilde{x},\tilde{y}=F,\tilde{z}) - \Psi(\tilde{x},\tilde{y}=F+2\delta,\tilde{z})\right]M_y(\tilde{x}-x,\tilde{z}-z)\right\}. \tag{2.7}$$

Again, C is just a constant that goes away with normalization and M_y is defined by (2.4). Even with this simplification, the values of magnetic potential are still undetermined. The strategy for managing these unknowns starts by defining the

magnetic potential on the ABS of the read element. This function is at best an approximation and represents the magnetic potential where

$$\Psi_S(x,z) = \Psi(x, y=0, z)\,. \tag{2.8}$$

Now it is necessary to utilize the normal derivative at the read element surface of the three-dimensional Green's function assuming the perfect image in the SUL,

$$\frac{\partial G}{\partial n} = \frac{\partial G}{\partial \tilde{y}}(\tilde{y}=0) = \frac{-2y}{[(x-\tilde{x})^2 + y^2 + (z-\tilde{z})^2]^{3/2}}\,. \tag{2.9}$$

With (2.9) the magnetic potential at any point under the read head can be determined by the following well-defined Dirichlet boundary value solution to Laplace's equation, $\nabla^2\Psi = 0$, [56],

$$\begin{aligned}
&\Psi(x,y,z)\\
&= -\frac{1}{4\pi}\iint \Psi(\tilde{x},0,\tilde{z})\frac{\partial G}{\partial n}\,d\tilde{x}d\tilde{z}\\
&= \frac{y}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{\Psi_S(\tilde{x},\tilde{z})d\tilde{x}d\tilde{z}}{[(x-\tilde{x})^2 + y^2 + (z-\tilde{z})^2]^{3/2}}\,.
\end{aligned} \tag{2.10}$$

In order for this somewhat convoluted process to be useful, $\Psi_S(\tilde{x},\tilde{z})$ must be provided. Depending on the exact form for $\Psi_S(\tilde{x},\tilde{z})$, (2.10) can be solved either analytically or numerically for the two functions $\Psi(\tilde{x},\tilde{y}=F,\tilde{z})$ and $\Psi(\tilde{x},\tilde{y}=F+2\delta,\tilde{z})$. Finally, these two evaluations of the magnetic potential can be used within the reciprocity integral of (2.7) which can then be solved numerically to arrive at the two-dimensional magnetic flux. All unspecified constants are simply incorporated

into a final coefficient for the magnetic flux and then dropped under normalization. This effectively results in a two-dimensional readback waveform.

Since these derivations are based on a single magnetic island, the specified parameters for the read head and island size tend to scale the readback pulse response. To demonstrate this, consider the following sets of parameters shown in Table 2.1 that represent two different recording densities, one at 2 Tb/in^2 and another at 4 Tb/in^2.

Table 2.1 – Various design parameters describing the dimensions of an individual island and read head at recording densities of 2 Tb/in^2 and 4 Tb/in^2.

		Recording Density (Tb/in^2)	
Parameter	**Symbol**	**2**	**4**
Read element thickness	T	3	3
Read element width	W	15	15
Shield gap	G	6	6
Flying height	F	10	3
Island height	δ	10	8
Island size (along track)	a	9	6.35
Island size (cross track)	b	9	6.35
Bit period	T_x	18	12.7
Track pitch	T_z	18	12.7

In conjunction with the parameters for a recording density of 2 Tb/in^2, the magnetic potential on the ABS can be defined by the following [57]

$$\Psi_S(x,z) = \begin{cases} \left(G + \frac{T}{2} + x\right)/G & -G - \frac{T}{2} \le x < -\frac{T}{2}; -\frac{W}{2} \le z \le \frac{W}{2} \\ 1 & -\frac{T}{2} \le x < \frac{T}{2}; -\frac{W}{2} \le z \le \frac{W}{2} \\ \left(G + \frac{T}{2} - x\right)/G & \frac{T}{2} \le x < \frac{T}{2} + \mathrm{G}; -\frac{W}{2} \le z \le \frac{W}{2} \\ 0 & \text{otherwise} \end{cases} . \quad (2.11)$$

This represents a potential on the ABS that is linearly attenuated between the read element and the two side shields. A graph of this potential is shown in Figure 2.3.

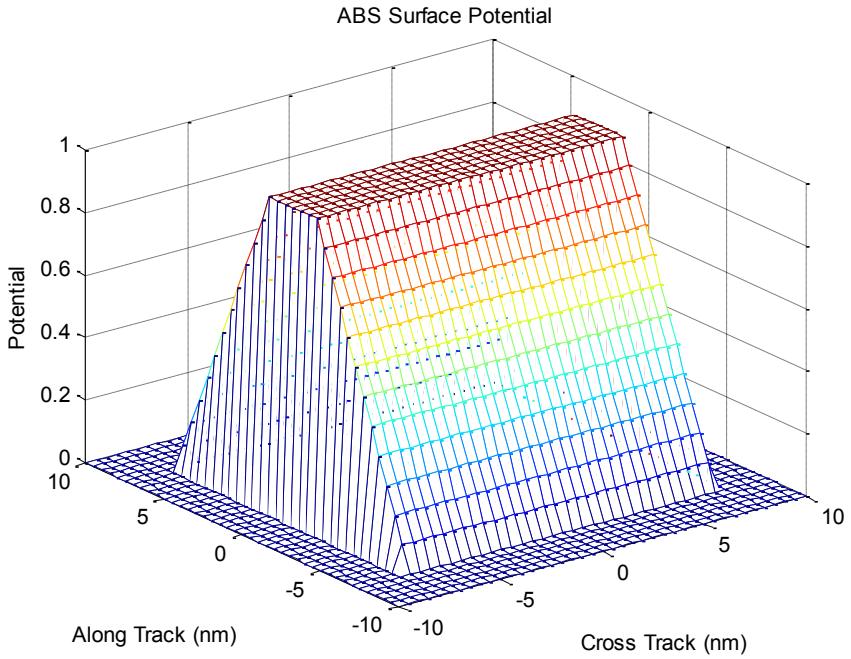

Figure 2.3 – The magnetic potential along the air bearing surface of a read head with linear attenuation between sensing element and side shields.

As can be seen in the figure, the magnetic potential on the unshielded sides is assumed to fall off immediately to zero. When this simple magnetic potential is substituted into (2.10), the following analytical solution exists [58]:

$$\Psi(x,y,z) = \frac{y}{4\pi G} log\left[\frac{R+(z-\tilde{z})}{R-(z-\tilde{z})}\right] + \frac{G+\frac{T}{2}+x}{2\pi G} arctan\left[\frac{(z-\tilde{z})(x-\tilde{x})}{yR}\right]\Bigg|_{\tilde{x}=-G-\frac{T}{2}}^{-\frac{T}{2}}$$

$$+\frac{1}{2\pi} arctan\left[\frac{(z-\tilde{z})(x-\tilde{x})}{yR}\right]\Bigg|_{\tilde{x}=-\frac{T}{2}}^{\frac{T}{2}} \tag{2.12}$$

$$-\frac{y}{4\pi G} log\left[\frac{R+(z-\tilde{z})}{R-(z-\tilde{z})}\right] + \frac{G+\frac{T}{2}-x}{2\pi G} arctan\left[\frac{(z-\tilde{z})(x-\tilde{x})}{yR}\right]\Bigg|_{\tilde{x}=\frac{T}{2}}^{G+\frac{T}{2}}.$$

The entire expression is evaluated across $-\frac{W}{2} \leq \tilde{z} \leq \frac{W}{2}$ and the *R*-term is defined as $R = \sqrt{(x-\tilde{x})^2 + y^2 + (z-\tilde{z})^2}$. This result allows for the evaluation of the magnetic potential at any point beneath the read head. Fortunately, only two locations are of interest: $\Psi(\tilde{x}, \tilde{y} = F, \tilde{z})$ and $\Psi(\tilde{x}, \tilde{y} = F + 2\delta, \tilde{z})$. By evaluating the analytical expression at the top of the island ($\tilde{y} = F$) and then again at the top of the island's image in the SUL ($\tilde{y} = F + 2\delta$), the magnetic potentials at the locations of interest are obtained and shown in Figures 2.4 and 2.5. These evaluations are the exact expressions required in the evaluation of the reciprocity integral of (2.7). To determine the overall flux and thus the readback signal pulse response per (2.1), the reciprocity integral should be evaluated numerically using the two previous evaluations of the magnetic potential and the island magnetization defined by (2.4). The resulting normalized pulse response is shown in Figure 2.6.

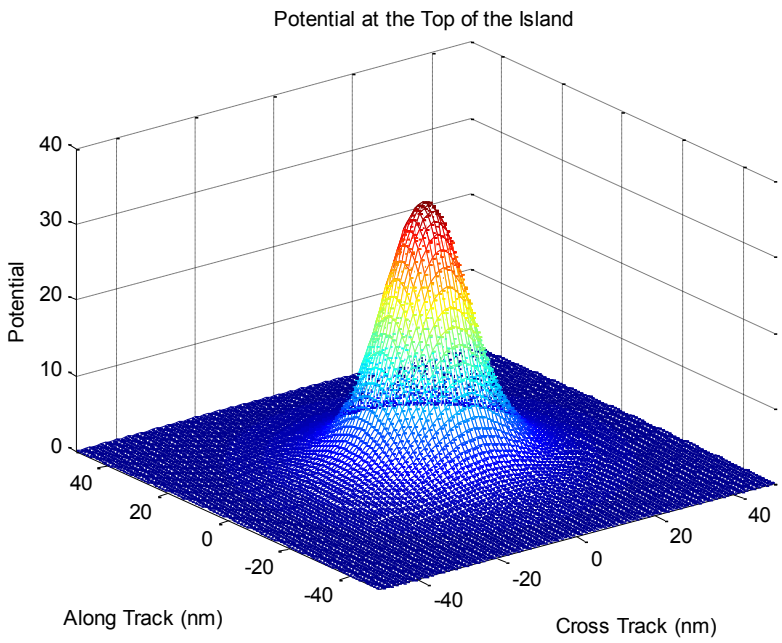

Figure 2.4 – Magnetic potential measured by the read head coincident with the top of a BPMR island for an island height of 10 nm.

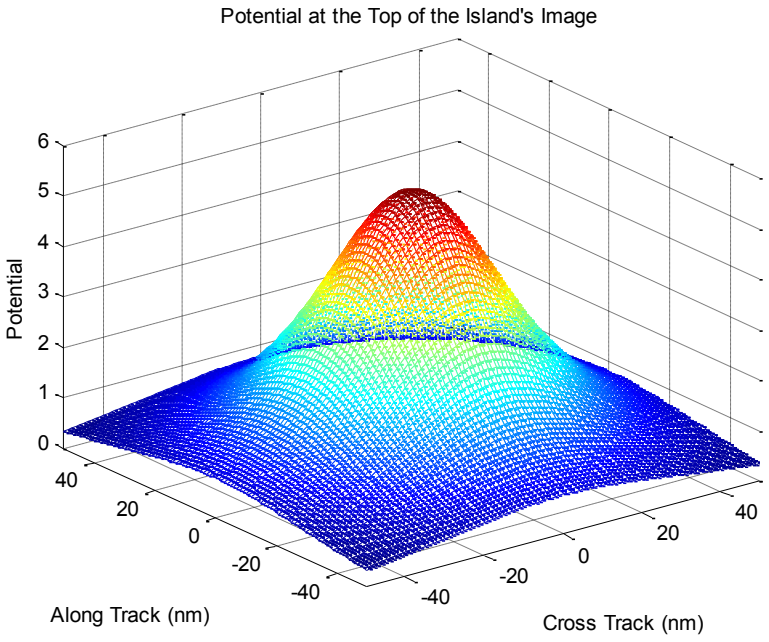

Figure 2.5 – Magnetic potential measured by the read head coincident with the top of a BPMR island's image in the SUL at a distance of 30 nm.

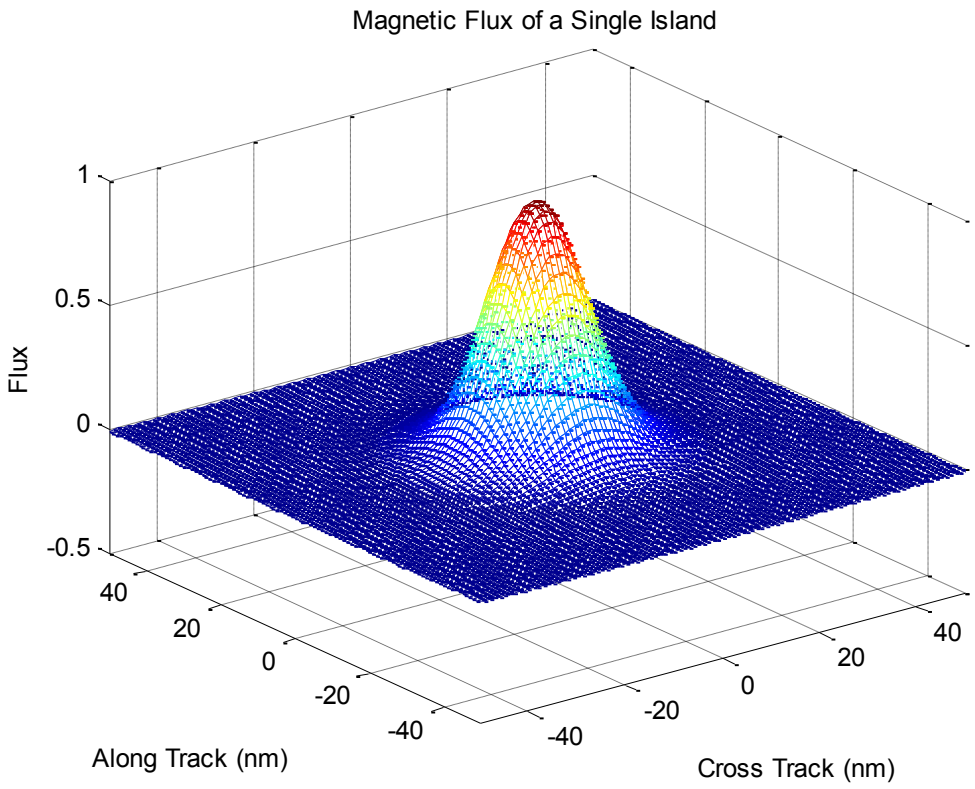

Figure 2.6 – Normalized magnetic flux of a single BPMR island at a recording density of 2 Tb/in^2.

The normalized pulse response is basically the readback waveform for an individual island sensed by the read head. Its shape is closely Gaussian and very nearly symmetrical in both along-track and cross-track directions as can be seen in the contour profile of Figure 2.7 and two-dimensional signal profiles of Figure 2.8. Closer inspection reveals that the cross-track profile is slightly broader compared to the along-track profile. This phenomenon has been observed by others utilizing similar analytical methods [59] and even more sophisticated techniques involving numerical micromagnetic simulations in combination with various analytic aspects [60].

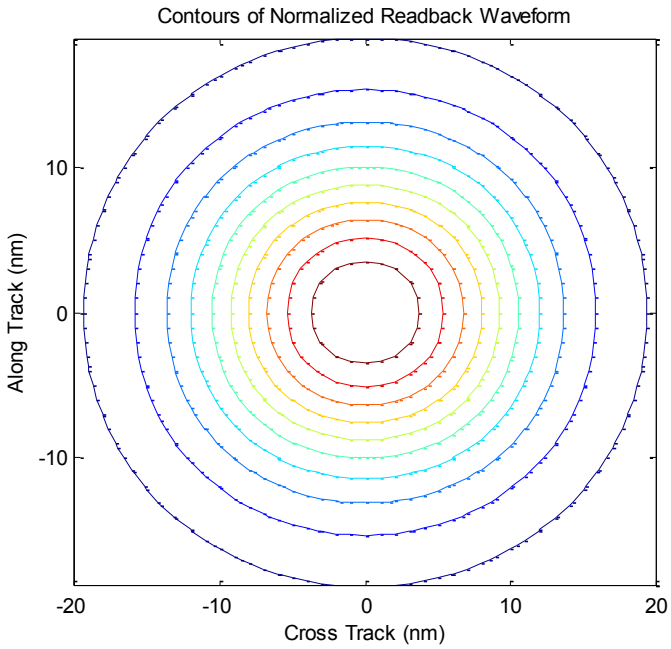

Figure 2.7 – Contours for the normalized pulse response of a single BPMR island showing slight asymmetry in the cross-track direction compared to the along-track direction.

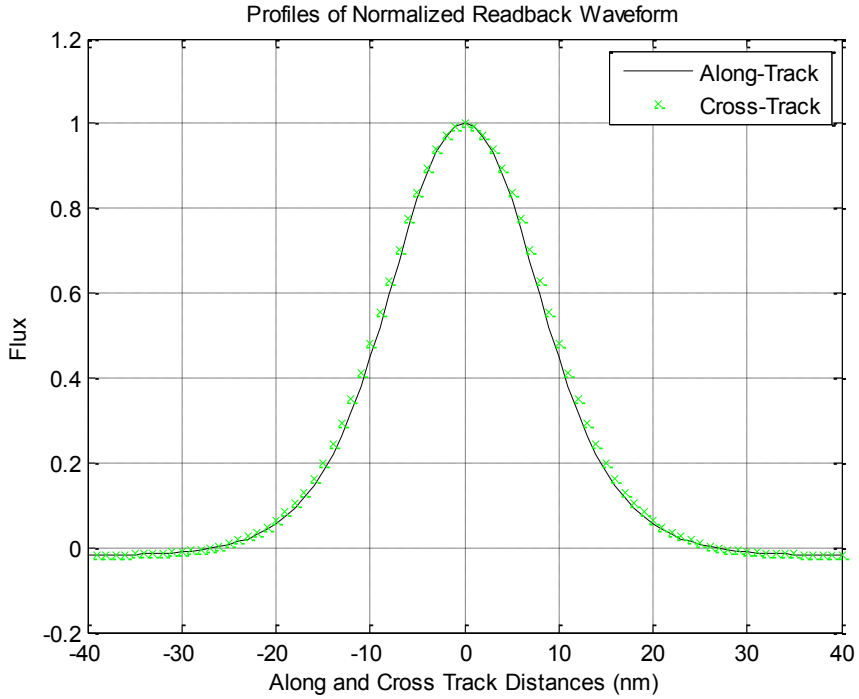

Figure 2.8 – Along-track and cross-track profiles of the individual pulse response of a single BPMR island.

To better see how the individual pulse response scales with increased recording density, the entire derivation process was repeated with parameters suitable for a density of 4 Tb/in^2. The final computed magnetic flux and corresponding profiles are shown in Figures 2.9 and 2.10.

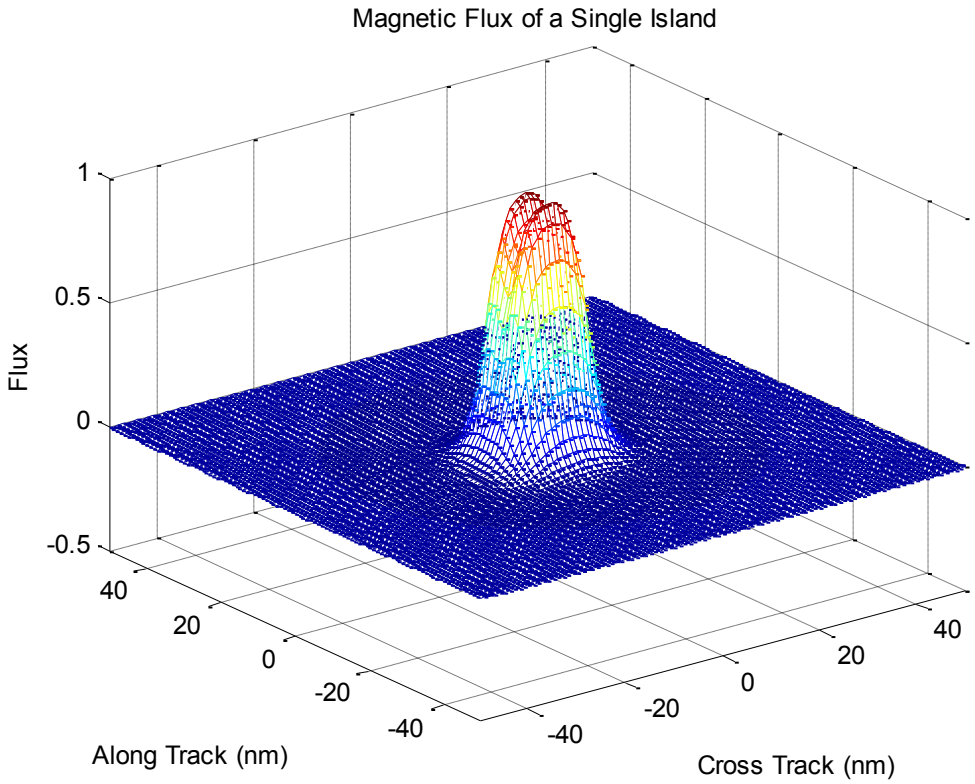

Figure 2.9 – Normalized magnetic flux of a single BPMR island at a recording density of 4 Tb/in^2.

At first glance the pulse response seems to be a scaled down version of the previous response at a recording density of 2 Tb/in^2 but with a noticeable depression at the center directly over the island. This is most likely due to the extreme proximity of the read head to the island. The original assumption of linear attenuation in the magnetic potential along the ABS was also highly simplified and is likely contributing

to the misshaped peak. The pulse profiles provide a clearer picture of the depression at the center of the pulse.

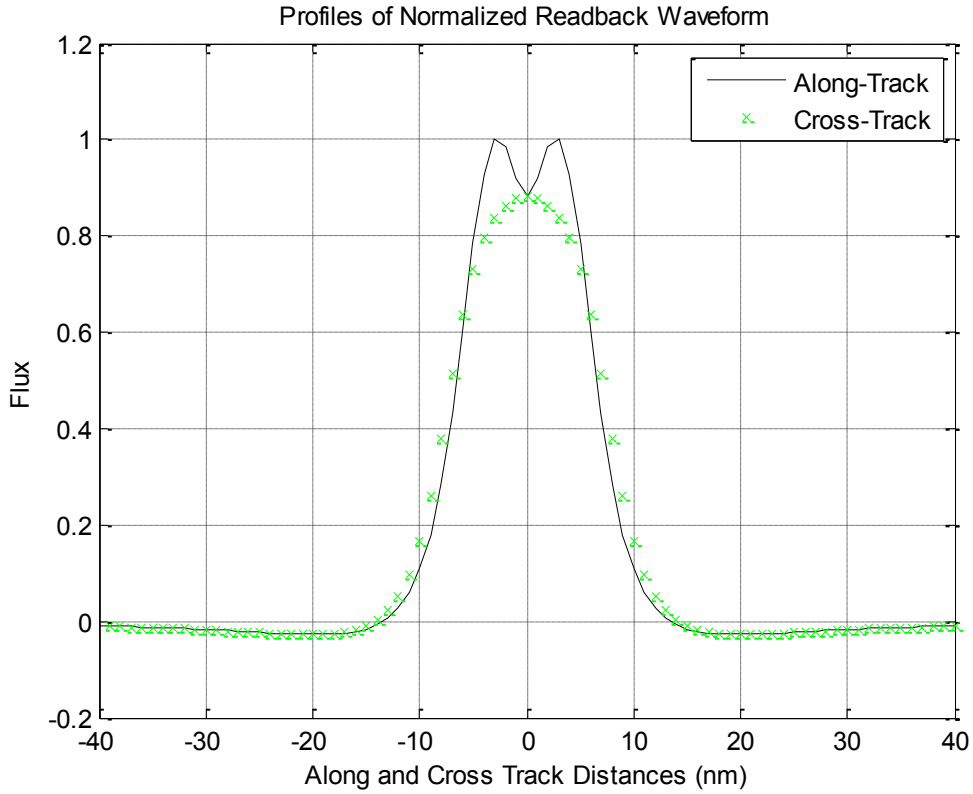

Figure 2.10 – Along-track and cross-track profiles of the individual pulse response of a single BPMR island at a recording density of 4 Tb/in^2.

In a comparison between Figures 2.8 and 2.10, it can be seen that the pulses at the two different recording densities are somewhat scaled with the broader cross-track profile showing up in each. However the feature at the peak for 4 Tb/in^2 is cause for some concern. The model for linear attenuation along the ABS does not work as well at high recording densities when the islands are significantly close together and the flying height is necessarily reduced. A better model for the potential would have

been the one proposed by Ruigrok [61] where a curvature term is added to the linear variation between the shields and sensing element.

$$\Psi_S(x,z) = \frac{G + \frac{T}{2} - |x|}{2G} + \frac{1}{\pi} arcsin\left(\frac{G + \frac{T}{2} - |x|}{G}\right). \quad (2.13)$$

While this results in steeper falloff of the potential along the ABS, the increased curvature will still be insufficient to adequately display the true nature of the potential on the ABS in a high density BPMR design. If a cleaner representation of the readback pulse response is to be obtained, then better approximations or modeling techniques must be employed. Fortunately, a number of researchers have already traveled this road and met with various levels of success. The approaches aimed at determining better signal response models typically stem from established methods and models used previously for perpendicular recording on continuous media.

To arrive at a pulse response either an analytical approach or a strictly numerical approach can be taken. Either one must incorporate approximations regarding the magnetic potential on the ABS and in the region directly beneath the ABS with how the potentials interact with the specified medium. No approach is error free and all suffer some form of degradation based on extent of the approximations used. In this work an analytic approach was taken that suffices to model the pulse response until the scaling down of the island and read head dimensions to achieve an areal density of 4 Tb/in^2 brings the head too close to the

island. Nabavi [59] used the same method but made different assumptions on the ABS potential. She chose the potential recommended by Wiesen and Cross [62] in combination with the mesh splicing techniques of Lindholm [63] to reproduce a more realistic ABS potential. Unfortunately, Nabavi never scaled her resulting pulse response beyond 2Tb/in^2, so it is unknown if it would suffer the same degradation from a drastically reduced flying height. Along similar lines Nutter in combination with various other authors [45], [56], [64] investigate readback pulse responses constructed using 3D reciprocity in the Fourier transform domain:

$$\varphi(x,z) = \mu_0\{\mathrm{IFT}[\widehat{M}_y^*(k_x,k_z)\cdot\hat{\varphi}(k_x,d,k_z)] - \mathrm{IFT}[\widehat{M}_y^*(k_x,k_z)\cdot\hat{\varphi}(k_x,d+\delta,k_z)]\}, \tag{2.14}$$

where $M_y \leftrightarrow \widehat{M}_y$ and $\varphi \leftrightarrow \hat{\varphi}$ are Fourier transform pairs, $\widehat{M}_y^*$ is the complex conjugate of $\widehat{M}_y^*$, IFT designates the inverse Fourier transform, and k_x and k_z are the Fourier transform wave numbers in their respective directions. When formulating a pulse response with Fourier transforms the magnetization distribution along the plane of the ABS and the projected potential distribution on any plane below the ABS are required. Nutter *et al*. wisely used an excellent method by Wilton *et al*. [65] that approximates such potential distributions with results that closely match numerical simulations. Nutter *et al.*'s resulting pulse responses closely match Nabavi's and the one shown in Figure 2.6, but again, the simulations are not attempted at an areal density of 4 Tb/in^2. Another possible approach would be to perform micromagnetic simulations with the parameters appropriate for 4 Tb/in^2 to

determine a readback response. This requires a finite-element method based on various physics-constrained approximations and is capable of handling differing magnetic properties of the materials in combination with complex write/read head and medium geometries. It is strictly numerical, computationally heavy and sensitive to localized boundary conditions. However, given that the methods of [65] nearly match the results from a finite-element method, the approach is unlikely to produce a pulse response much different than those found by Nutter *et al*.

2.2.1 Modeling with a 2D Gaussian Pulse

While much of this work is beneficial, performing sophisticated microsimulations or intensive numerical calculations to determine the nature of every island does not lend itself to further extension where the inner simulations are wrapped within an error-correction code. Fortunately, a good deal of the cumulative results are consistent and point to the alternative model using a two-dimensional (2D) Gaussian distribution function [59]. A 2D Gaussian pulse closely matches the results for the pulse response derived earlier as well as the results of others [45], [56], [64]. It has also been used successfully to model the channel response in conjunction with both two-dimensional detection techniques [46], [55], [66], [67] and error-correcting codes [47], [68]. There are various characteristics of this pulse that make it versatile for use within simulations. In terms of a readback response, a 1D Gaussian pulse can be expressed

$$h(x) = A\exp\left\{-\frac{1}{2}\left(\frac{cx}{PW_{50}}\right)^2\right\}, \tag{2.15}$$

where A is the amplitude of the pulse, PW_{50} is the pulse width at half maximum in the *x*-direction and c is a constant ($c^2 = 8\ln 2$) used to associate the PW_{50} to the standard deviation of a Gaussian. When expanded as a 2D Gaussian pulse with $h(x,z) = h(x)h(z)$,

$$h(x,z) = A\exp\left\{-\frac{1}{2}\left(\frac{cx}{PW_{50x}}\right)^2\right\}\exp\left\{-\frac{1}{2}\left(\frac{cz}{PW_{50z}}\right)^2\right\}. \tag{2.16}$$

This expression allows for separable treatment of the readback response in both the along-track and cross-track directions. This function can also be expressed as

$$h(x,z) = A\exp\left\{-b\left(\frac{x}{T_{50x}}\right)^2\right\}\exp\left\{-d\left(\frac{z}{T_{50z}}\right)^2\right\}, \tag{2.17}$$

where b and d are parameters whose values depend on various read head and media parameters which vary for a given areal density, and T_{50} is simply the rise time for the transition response to go from negative 50% amplitude to positive 50% amplitude. A 2D Gaussian pulse for both 2 and 4 Tb/in^2 areal densities along with their corresponding profiles can be seen in Figures 2.11 – 2.13 where the broader cross-track profile compared to the along-track profile is easily seen. What is not so apparent is the correspondence between the along-track profiles at the two different densities. For the parameters from Table 2.1, the readback pulse at 2

Tb/in^2 is slightly broader in the along-track direction than at 4 Tb/in^2. This is not completely expected and will be explained in a later section.

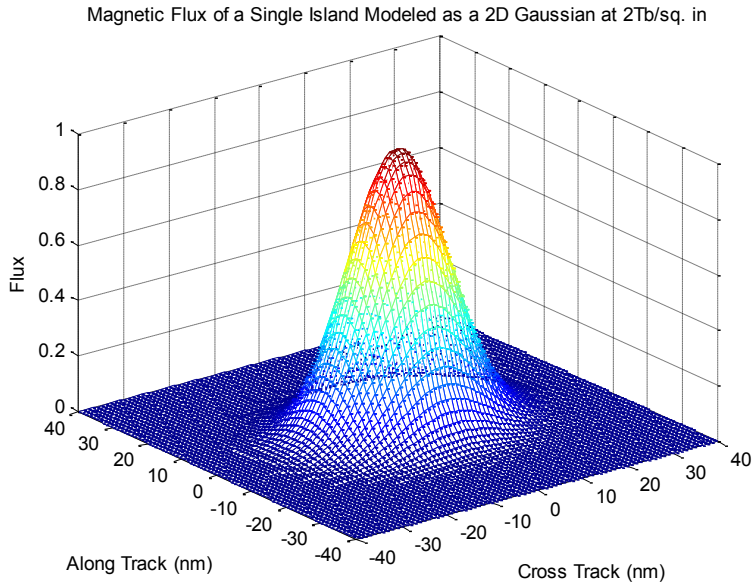

Figure 2.11 – Normalized 2D Gaussian readback pulse response at an areal density of 2 Tb/in^2.

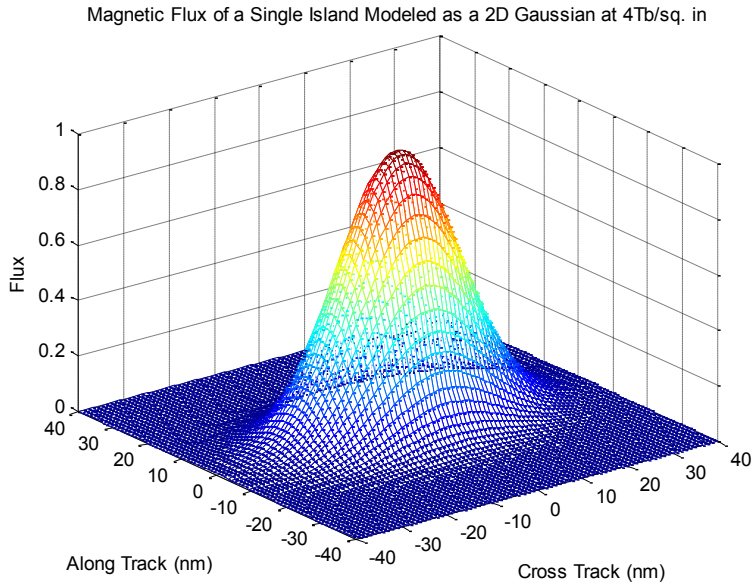

Figure 2.12 – Normalized 2D Gaussian readback pulse response at an areal density of 4 Tb/in^2.

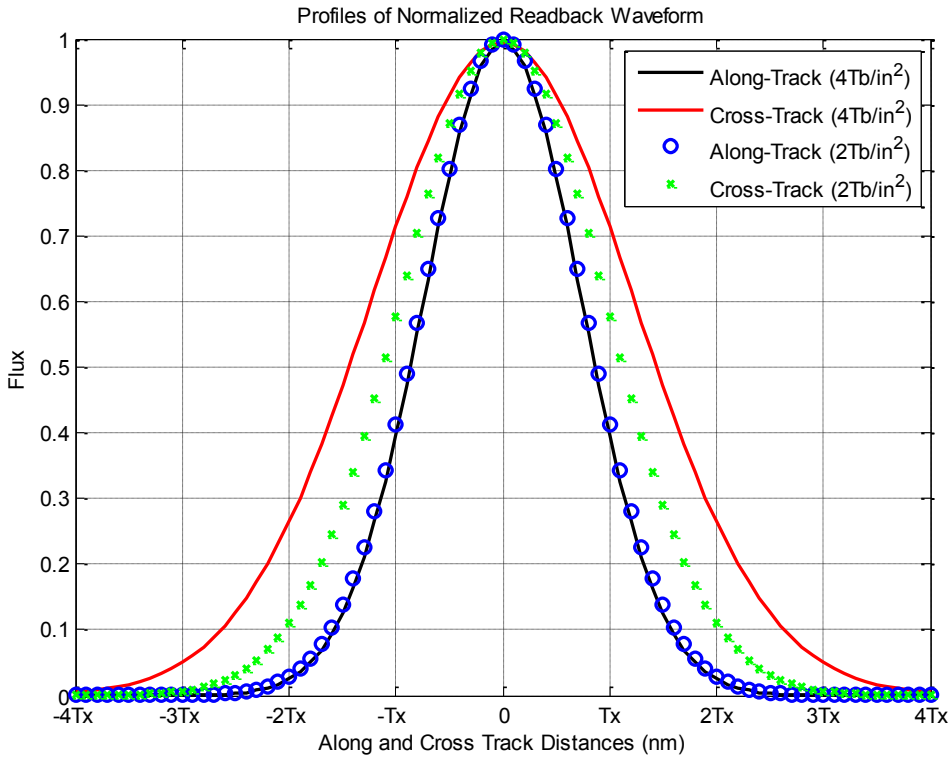

Figure 2.13 – Along-track and cross-track profiles for both 2 and 4 Tb/in^2 areal densities.

At the higher areal density the pulse broadens significantly in the cross-track direction and this will contribute to increased ITI. Thus, at 4 Tb/in^2 significantly more ITI will be present in the channel while the ISI levels will remain relatively steady when compared to the ITI and ISI at half that areal density. The full extent at which ITI exists at 4Tb/in^2 is displayed in the contour plot of the pulse response in Figure 2.14. Obviously the island will exert a localized influence in both along-track and cross-track directions. If the 1D readback response profiles are expressed as a train of five pulses as in Figures 2.15 and 2.16, then the extent of the two-dimensional interference can be seen.

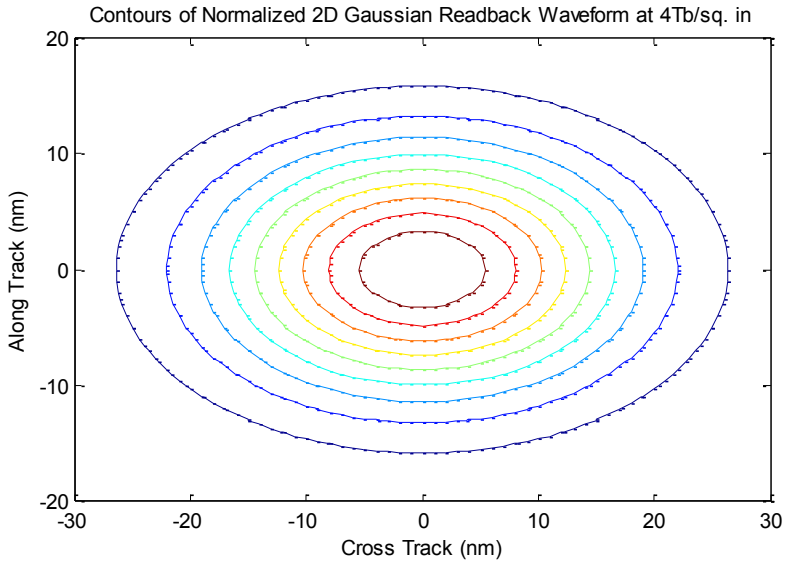

Figure 2.14 – Contour plot of the normalized 2D Gaussian pulse response for an areal density of 4 Tb/in^2.

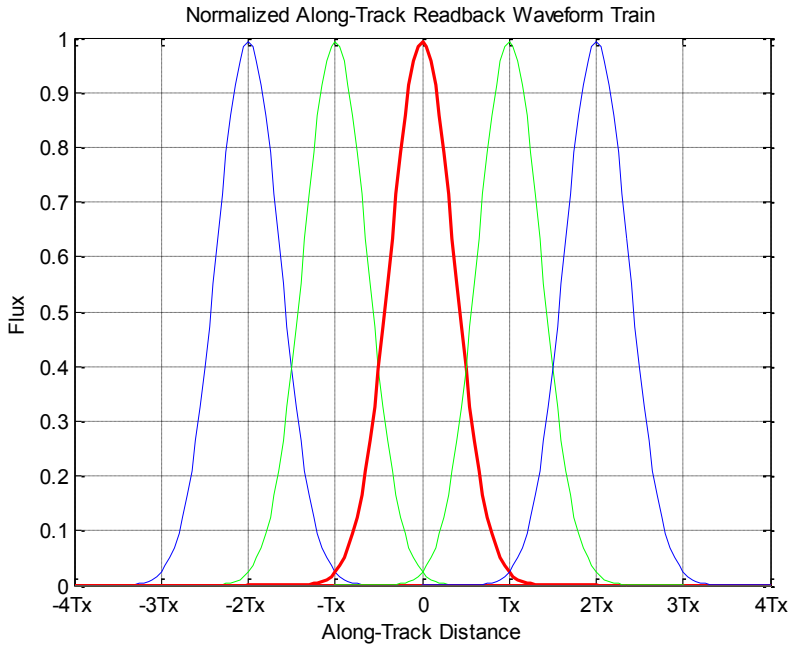

Figure 2.15 – Along-track profiles of the normalized 1D Gaussian pulse as a train of five adjacent pulses at an areal density of 4 Tb/in^2. The centers of the various islands are a distance T_x away and coincide with the pulse peaks. The center island pulse is in red, the adjacent island pulses are in green, and the next two adjacent island pulses are blue.

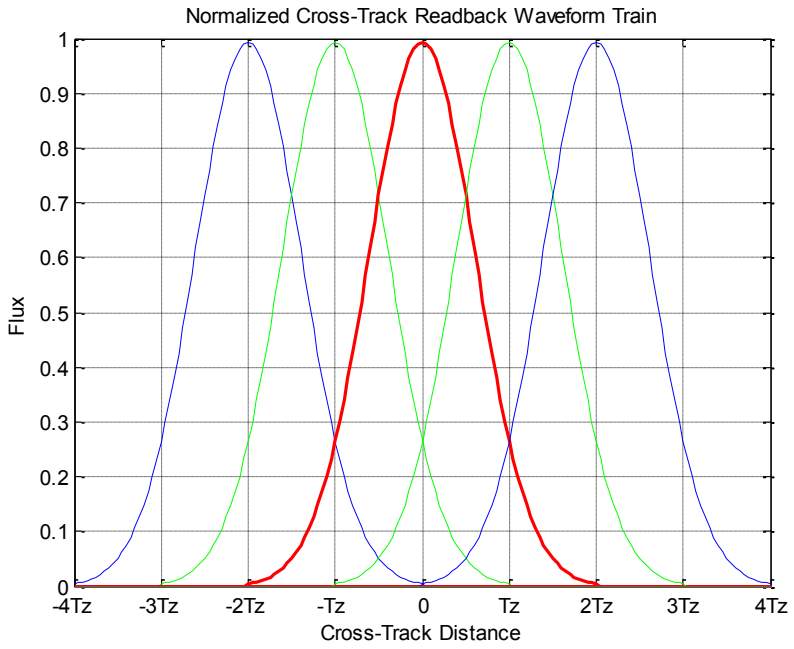

Figure 2.16 – Cross-track profiles of the normalized 1D Gaussian pulse as a train of five adjacent pulses at an areal density of 4 Tb/in^2. The centers of the various islands are a distance T_z away and coincide with the pulse peaks. The center island pulse is in red, the adjacent island pulses are in green, and the next two adjacent island pulses are blue.

When analyzing the pulse overlaps in Figures 2.15 and 2.16, it should be noted that the ISI is only significant from the nearest set of islands at positions $\pm T_x$. The islands at $\pm 2T_x$ do not directly increase the ISI on the center pulse. But as expected, the ITI is a more difficult issue. The influence on the center pulse from adjacent pulses is quite high. So much so that the outer set of pulses exert a non-zero interference on the center pulse. When the islands are perfectly arranged within their respective tracks, the ITI from distances of $\pm 2T_z$ is not very significant. But when media noise is present and the islands are located somewhere around their intended center location, the ITI from islands that are $\pm 2T_z + \Delta z$ away could become very important. One method of combating the high levels of ITI in the channel is to

arrange the islands differently. Figure 2.17 shows what a small array of pulses look like from the read head perspective.

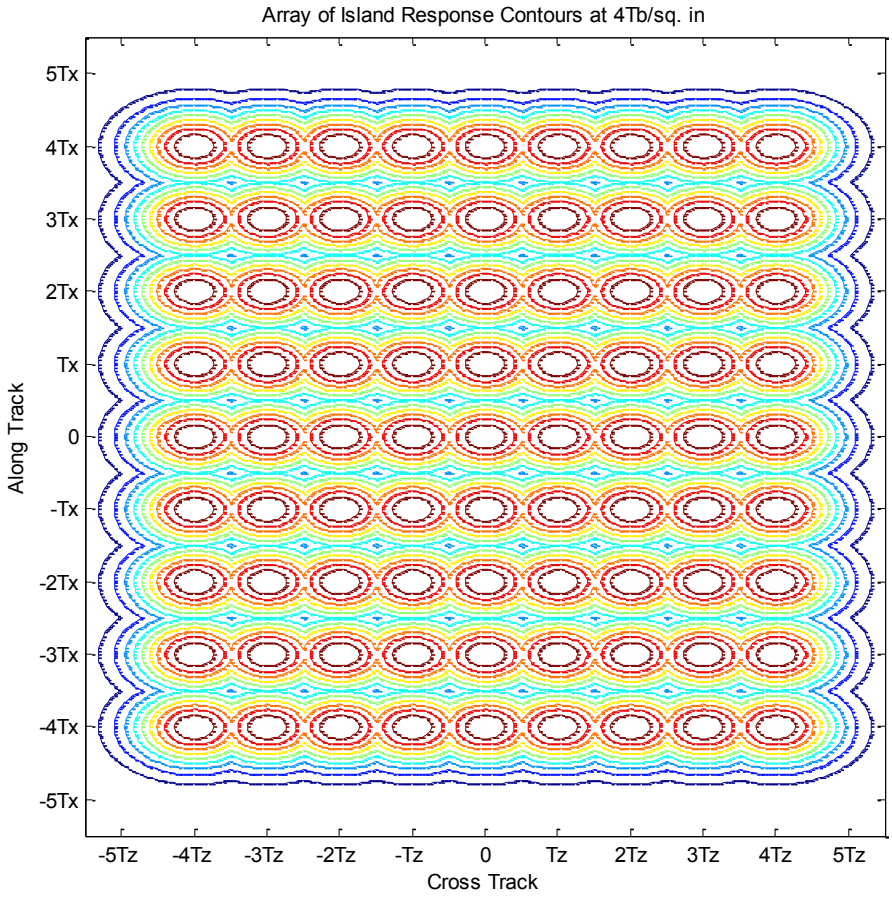

Figure 2.17 – Array of 81 normalized 2D Gaussian pulses shown as contours for an areal density of 4 Tb/in^2.

From the contours in Figure 2.17, one can observe a low spot in the pulse response field located at the geometric intersections of island center cross diagonals. This indicates that a staggered pattern of adjacent tracks (hexagonal packing of the individual islands) might allow for decreased levels of ITI. However, moving adjacent tracks to the indicated intersection $\left(\frac{T_x}{2}, \frac{T_z}{2}\right)$ causes the corners of the islands in adjacent tracks to physically touch which would result in more problems than the

hexagonal packing of the islands could solve. This type of hexagonal packing is best implemented where the condition of (1.12) is not at its minimum and the islands are physically separated by a distance greater than half of the track pitch, $\frac{T_z}{2}$. Another similar strategy though, would be to shift adjacent tracks forward by half the bit period, $\frac{T_x}{2}$, while maintaining the same track spacing, T_z. This still places adjacent islands at a geographic low in a rectangular equivalent arrangement and maintains the BAR and track spacing at the same time. In a hexagonal packing such as this, ITI is basically traded off as ISI. Given the previous investigation of the track pulse profiles, this could prove advantageous. Figure 2.18 gives a clearer picture of the extent of the ISI at the 2 and 4 Tb/in^2 areal densities.

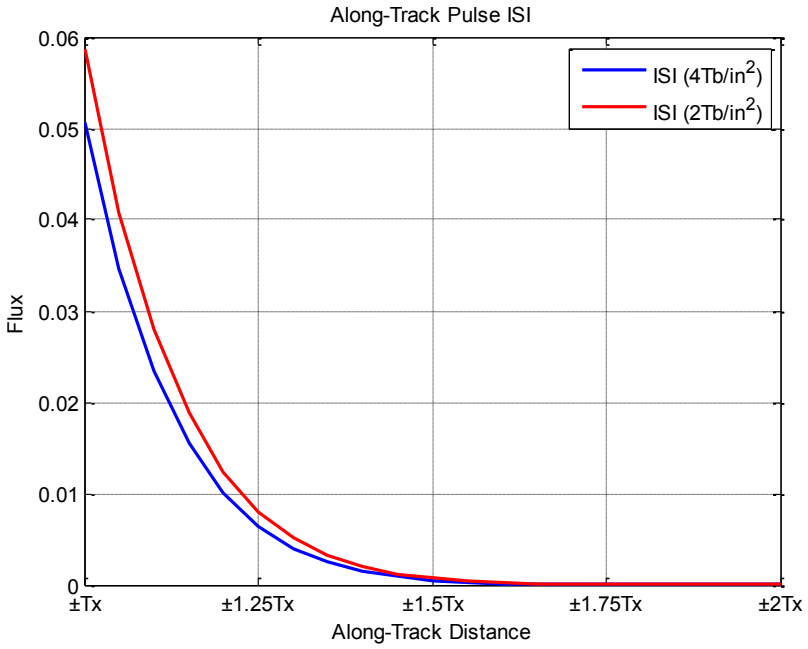

Figure 2.18 – ISI present in the normalized 2D Gaussian readback pulse at areal densities of 2 and 4 Tb/in^2.

As noted earlier, the ISI is marginally more of a concern for the lower areal density channel, but in either case, the level of total ISI in the channel is below 10% of the magnitude of the pulse. In terms of the ITI, moving to a hexagonal pattern shows some improve (see Figure 2.19) but not anything drastic.

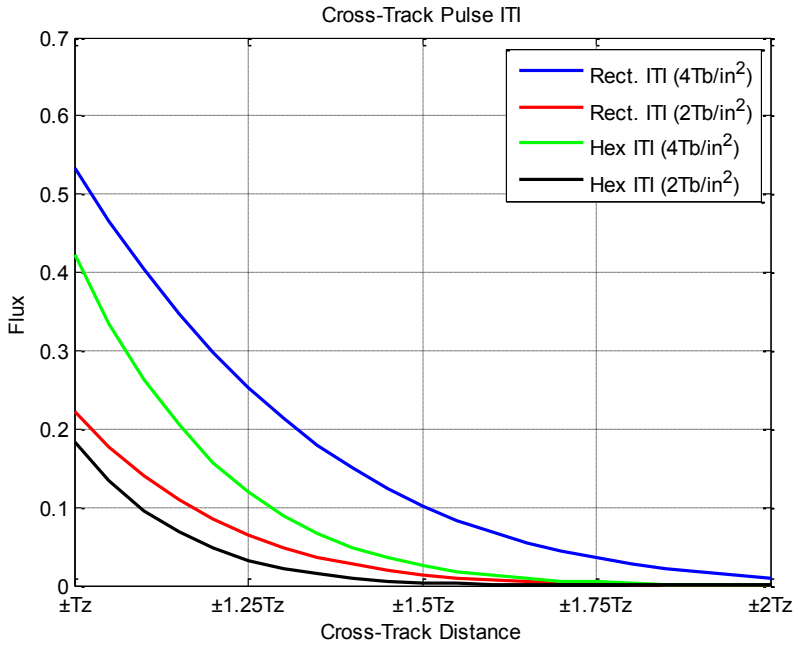

Figure 2.19 – ITI present in the normalized 2D Gaussian readback pulse at areal densities of 2 and 4 Tb/in^2 for both rectangular and hexagonal island patterns.

At the 4 Tb/in^2 areal density staggered islands saves about 10% on the total ITI compared to the orthogonal islands. For 2 Tb/in^2 the advantage drops to less than 5% improvement. While this may not seem like much to get excited about, the additional hardship of media noise may make even a 5% decrease in ITI well worth it. In fact hexagonal patterning of the islands with BAR = 1 is one of the easiest island distributions to manufacture, and it does not significantly affect costs in terms of signal processing complexity.

2.2.2 Media Noise and its Effects on the Readback Response

During fabrication the individual islands are always constructed with less than perfect dimensions, position and intrinsic magnetic properties. The sum total of all the adversities which impact the readback response cannot be determined exactly, but some level of influence from the major contributing degradations can be incorporated into the pulse response. This is unique for BPMR compared to PMR, SWR/TDMR, HAMR and MAMR as the localization of the magnetic domains to individual islands eliminates the transition noise present in conventional continuous media magnetic recording schemes. While transitional noise is gone, the expansion of the pulse response into two dimensions increases the noise sources of the readback response. In all recording channels there are actually more noise sources than can be practically monitored. So the focus is always on those sources which cause a particular media the most trouble. For BPMR these have been found to be variations in island position and island size [64]. It was discovered in [69] that high BAR islands suffer more from island size jitter $(\Delta L_x, \Delta L_z)$ than lower BAR islands or islands with unit BAR. It was also concluded that these jitters were most detrimental to read head performance when the target island was small compared to the size of its neighbors. In [55] and [67] results showed that after jitter noise was incorporated into the design of the detector, position jitter $(\Delta x, \Delta z)$ became the most detrimental jitter noise contributor.

To incorporate jitter noise and its adverse effects to system performance, each kind of fabrication jitter must be included in the pulse response $h(x, z)$. This

means either (2.16) or (2.17) must be modified. In reference to the two system designs listed in Table 2.1, the parameters for an areal density of 2 Tb/in^2 match (2.16) while the parameters for an areal density of 4 Tb/in^2 match (2.17). Thus both equations must actually be modified. Starting with (2.16), the incorporation of jitter terms results in

$$h(x+\Delta x, z+\Delta z, \Delta L_x, \Delta L_z, \Delta A) = (A+\Delta A)$$
$$\times \exp\left[-\frac{c^2}{2}\left(\frac{x+\Delta x}{PW_{50x}+\Delta L_x}\right)^2\right] \cdot \exp\left[-\frac{c^2}{2}\left(\frac{z+\Delta z}{PW_{50z}+\Delta L_z}\right)^2\right], \qquad (2.18)$$

where ΔA is the island height variation, Δx is the along-track position jitter, Δz is the cross-track position jitter, ΔL_x is the along-track island size variation, and ΔL_z is the cross-track island size variation. All of these jitters are modeled as white Gaussian random processes with zero means, and none of the jitters are considered to be correlated. When an equivalent jitter treatment is applied to (2.17), the result is

$$h(x+\Delta x, z+\Delta z, \Delta L_x, \Delta L_z, \Delta A) = (A+\Delta A)$$
$$\times \exp\left[-b\left(\frac{x+\Delta x}{T_{50x}+\Delta L_x}\right)^2\right] \exp\left[-d\left(\frac{z+\Delta z}{T_{50z}+\Delta L_z}\right)^2\right], \qquad (2.19)$$

where the jitters are the same as before. For (2.18) and (2.19) one simplification can be made if a high-resolution numerical pulse response is known. Based on the results of such a numerical simulation of the pulse response at a specific density, the parameters of the analytic 2D Gaussian pulse can be fitted to closely match. This procedure was implemented in [67] where it was revealed that the island size jitter affected both the amplitude and width of the numerical pulse. This characteristic of

the readback response is distinguished by a pulse amplitude that is nearly linear in the island size variation at the target island's geometric center in the along-track and cross-track directions. Thus a best-fitting line can be constructed in each direction based on a minimum mean-squared error (MMSE) criteria to get $y_x = m_x \Delta L_x + b_x$ and $y_z = m_z \Delta L_z + b_z$ where m_x, b_x and m_z, b_z are the respective slopes and y-intercepts in each direction of the best-fitting MMSE lines. Incorporating this characteristic of the readback response into (2.19) results in

$$h(x + \Delta x, z + \Delta z, \Delta L_x, \Delta L_z) = \left\{\exp\left[-b\left(\frac{x + \Delta x}{T_{50x}}\right)^2\right] \times y_x\right\}\left\{\exp\left[-d\left(\frac{z + \Delta z}{T_{50z}}\right)^2\right] \times y_z\right\}, \tag{2.20}$$

where y_x and y_z are the jitter lines in each direction. If the constants are combined with the parameter values, (2.20) can be written in more compact form as

$$h(x + \Delta x, z + \Delta z, \Delta L_x, \Delta L_z) = a \times \exp[-b(x + \Delta x)^2](c + \Delta L_x) \times \exp[-d(z + \Delta z)^2](e + \Delta L_z), \tag{2.21}$$

with $a = m_x m_z$, $b = \tilde{b}/{T_{50x}}^2$, $c = b_x/m_x$, $d = \tilde{d}/{T_{50z}}^2$, and $e = b_z/m_z$ where $\tilde{b}, \tilde{d}$ simply refer to their previous parameter values unnormalized by T_{50x} and T_{50z}. To understand how these jitters adversely affect the readback pulse response, contributions from each jitter term are examined individually in Figures 2.20 – 2.23.

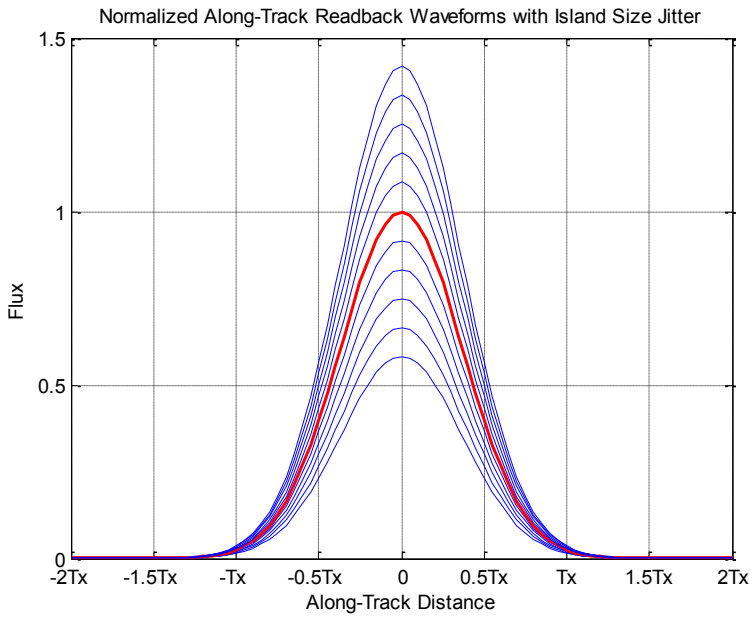

Figure 2.20 – Normalized along-track 2D Gaussian pulse readback waveform at an areal density of 4 Tb/in^2 under the influence of along-track island size variation. The red curve indicates a zero jitter value. The blue curves indicate jitter between $\pm$50% of the island's along-track size at intervals of 10%.

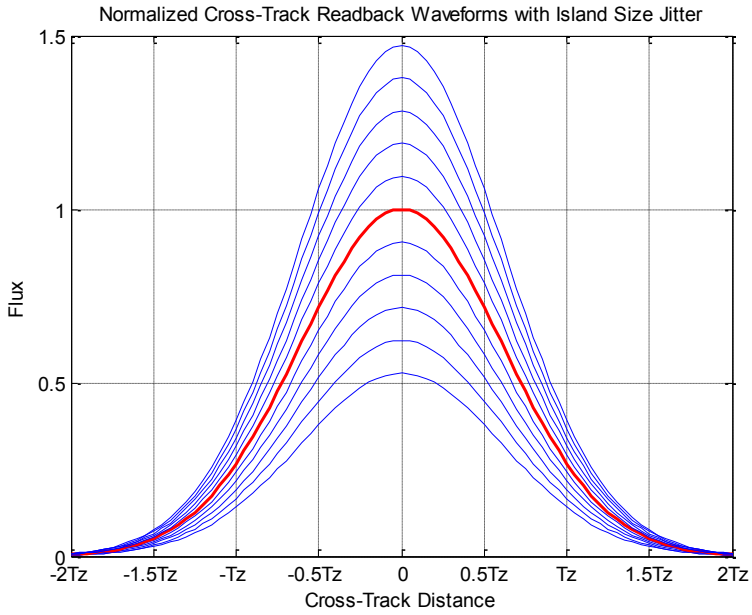

Figure 2.21 – Normalized cross-track 2D Gaussian pulse readback waveform at an areal density of 4 Tb/in^2 under the influence of cross-track island size variation. The red curve indicates a zero jitter value. The blue curves indicate jitter between $\pm$50% of the island's cross-track size at intervals of 10%.

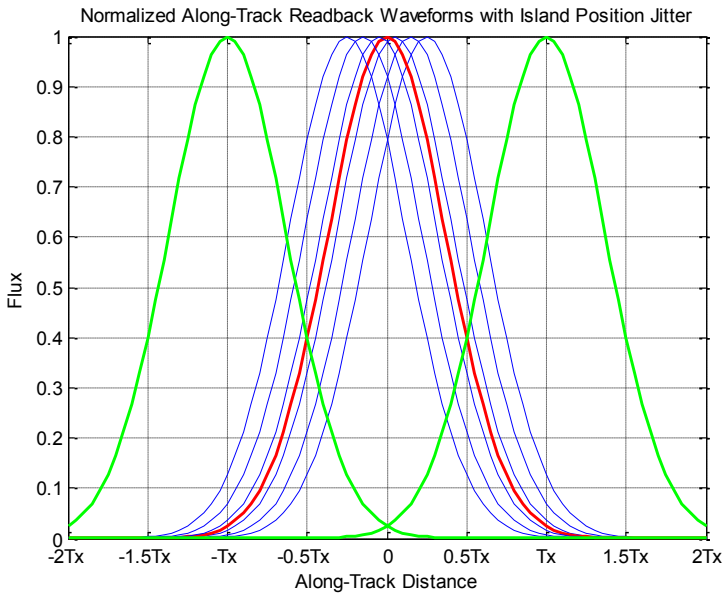

Figure 2.22 – Normalized along-track 2D Gaussian pulse readback waveform at an areal density of 4 Tb/in^2 under the influence of along-track island position jitter. The red curve indicates a zero jitter value. The blue curves indicate jitter between $\pm$25%, $\pm$15% and $\pm$5% of the bit period. The green curves indicate the position of the nearest along-track neighbors.

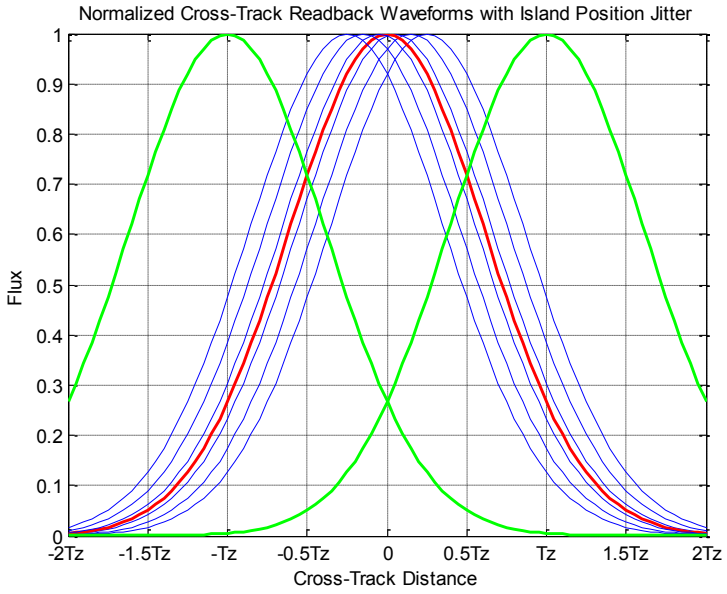

Figure 2.23 – Normalized cross-track 2D Gaussian pulse readback waveform at an areal density of 4 Tb/in^2 under the influence of cross-track island position jitter. The red curve indicates a zero jitter value. The blue curves indicate jitter between $\pm$25%, $\pm$15% and $\pm$5% of the track pitch. The green curves indicate the position of the nearest cross-track neighbors.

Based on the figures for island size variations, islands that are either bigger or smaller than their nominal size will produce only slight changes in the broadness of the pulse response but large changes in the amplitude. This is actually good news as the falloff from a higher amplitude island does not increase greatly with increasing island size. Thus only islands experiencing very large size jitter will cause problems in terms of ISI and ITI on its neighbors. Small islands will be a problem though as this directly decreases the desired signal amplitude and allows the high ITI present to cause single-bit written–in errors as also seen in [69]. As expected, the cross-track size jitter results in slightly greater amplitude variation compared to its along-track jitter counterpart. In terms of the position jitters, one can easily see why these fabrication defects result in damaging the readback response especially in the along-track direction. If an island experiences the maximum along-track jitter, the amplitude of the along-track pulse response drops by 20%. Given that its neighbors will also experience some level of position jitter, the resulting pulse response could very easily lead to a detector error. For the cross-track jitter, the broader pulse prevents steep drop-off in the pulse amplitude. But because of the high ITI environment, the influence of the neighboring islands results in greater relative decay of the pulse response compared to the same position jitter value in the along-track direction.

In order to actually implement the channel response with jitter noise, the 2D Gaussian response needs to be expanded to isolate the different contributions to the channel response. Referencing the method used previously in Section 1.2, the

individual jitter terms can be expressed as first-order Taylor series expansions with reasonable accuracy. This results in the following expression for the pulse response

$$\tilde{h}(x,z) \approx h(x,z) + \left[\Delta x \frac{\partial h(\cdot)}{\partial x} + \Delta z \frac{\partial h(\cdot)}{\partial z} + \Delta L_x \frac{\partial h(\cdot)}{\partial T_{50x}} + \Delta L_z \frac{\partial h(\cdot)}{\partial T_{50z}}\right], \quad (2.22)$$

where $\partial h(\cdot)$ designates the partial derivative of the expression for the full pulse response including jitters with respect to a given jitter. When modeling the channel output, the jitters will be considered Gaussian random variables normalized by T_x, T_z, L_x and L_z respectively with variances of $\sigma^2_{\Delta x}$, $\sigma^2_{\Delta z}$, $\sigma^2_{\Delta L_x}$ and $\sigma^2_{\Delta L_z}$. In the absence of a numerical model, the pulse response would fall back to (2.18) and (2.19). Under a first-order jitter approximation, (2.18) would become

$$\tilde{h}(x,z) \approx h(x,z) + \left[\Delta A \frac{\partial h(\cdot)}{\partial A} + \Delta x \frac{\partial h(\cdot)}{\partial x} + \Delta z \frac{\partial h(\cdot)}{\partial z} + \Delta L_x \frac{\partial h(\cdot)}{\partial PW_{50x}} + \Delta L_z \frac{\partial h(\cdot)}{\partial PW_{50z}}\right], \quad (2.23)$$

where the variation in island height is included.

2.3 Defining the SNR

For various PMR channels there are a number of ways to specify the signal-to-noise ratio (SNR), which is a common metric utilized to compare performance levels for both detection schemes and error-correcting codes. In continuous media PMR defining a functional SNR is a sensitive issue as the average bit energy associated with a single isolated signal response works out to be infinity. There is also the

additional problem caused by the presence of the transition jitter noise between bit cells of opposite magnetization which complicates the SNR definition. This resulted in a typical SNR definition of

$$\mathrm{SNR} = \frac{E_i}{N_0 + M_0} \ , \tag{2.24}$$

where E_i is the energy of the isolated pulse response, N_0 is the AWGN single-sided power spectral density and M_0 is a power spectral density version of the average transition jitter noise energy associated with a single transition. While this equation seems simple enough, arriving at this expression and determining the individual values for the signal and noise contributions are far from trivial [71].

As mentioned in the previous section, BPMR suffers from multiple jitter noise sources since its two-dimensional signal representation allows for increased sources of noise. This jump to two dimensions substantiates a revisited treatment of the typical SNR definition. Unfortunately, such an investigation of SNR and its many facets is lacking for BPMR. Thus an SNR definition which includes all contributions to an isolated bit response has yet to be formulated. If the contributions from the jitter are simply assumed to be masked within the signal power of an island's individual pulse response, then the standard 1D definition of SNR can be used:

$$\mathrm{SNR} = \frac{E_i}{N_0} \ . \tag{2.25}$$

For a binary-input channel subject to an error-correcting code, this definition is slightly modified to better suit the channel simulations which come later:

$$\mathrm{SNR} = \frac{E_b}{N_0} \Longrightarrow \mathrm{SNR} = \frac{RA^2}{2\sigma^2}, \tag{2.26}$$

where E_b is the average energy per information bit, $N_0/2$ is the two-sided power spectral density of the AWGN (equal to σ^2) corresponding to the channel output of a bit with energy E_b, R is the code rate of the ECC, and A is the amplitude of the isolated pulse response [72]. This definition requires a little explanation. The code rate R will reduce the channel's average energy per bit as it adds an overhead of parity bits with a ratio equal to the code rate. Also, for the remainder of this dissertation, the pulse amplitudes will be normalized so $A^2 = 1$ throughout. In the log domain the SNR definition used will be

$$\mathrm{SNR(dB)} = 10\log_{10}\left(\frac{RA^2}{2\sigma^2}\right). \tag{2.27}$$

2.4 Errors in the Writing Process

As mentioned previously, one of the greatest challenges facing BPMR lies in cleaning up the write process as it can introduce written-in errors into the channel that the system's signal processing design must handle. This is unfortunate because the errors created while writing the bits to the disk are unknown to the detection algorithms since detection is based on the readback response acquired while scanning islands in a track. If one of the islands was mistakenly written as a $(+1)$

instead of as a (-1) or vice versa, then the detection has no choice but to treat any such bit errors as correct writes. Thus, only the system's ECC has an opportunity to correct the written-in errors, and it must do so in conjunction with correcting mistakes made by the detector as well. Since continuous media PMR systems do not suffer significantly from miswritten bits, the error-correction codes used in BPMR must perform a dual function in guarding the information stored in the medium.

In BPMR written-in errors can basically be classified into two types: errors created by problems in write field when an individual island is addressed, and errors that occur because of miscues in write timing. The errors due to problems in the magnetic field fluctuations during writing are single-bit errors and are often referred to as either substitution errors or as reversal errors. Substitution errors only happen when the target island ends up magnetized in the wrong direction. There are a number of factors which determine if a given bit is miswritten or not. These include the intrinsic switching field distribution of an individual island (σ_{SFD}), the interfering magnetic fields from neighboring islands (σ_{H}), and the deviation in position (jitter) of the target island (σ_p). When these factors are taken to be independent, the total field fluctuation (σ) can be expressed [73]

$$\sigma^2 = \sigma_{\mathrm{SFD}}^2 + \sigma_{\mathrm{H}}^2 + \left(\frac{dH}{dx}\sigma_p\right)^2 . \tag{2.28}$$

The island position jitter deviation can be expressed as the product of the write field gradient with the position deviation. Often the write head is in shingled write orientation so it can have a large dimension in the along-track direction and a much

smaller size in the cross-track position. This means that the trailing edge of the write head determines the write condition as the next few future bits are overwritten at the same time. When the trailing edge of the write head is over its target island, it may still be close enough to the preceding island to overwrite that island's current magnetic condition. This actually depends on a number of interacting factors related to (2.28). The overwrite condition for the previous adjacent island can be stated

$$H_{max} - \frac{dH}{dx}(\Delta T_x + T_x) \geq H_c + \Delta H_c^{-1} + H_d^{-2} + H_d^0 \,, \tag{2.29}$$

where H_{max} is the applied write head field; $\frac{dH}{dx}$ is the trailing edge head field gradient; $(\Delta T_x + T_x)$ represents the island's distance from the write head's trailing edge; H_c is the medium's coercivity; ΔH_c^{-1} is the trailing island's deviation from the medium's magnetic field and represents the switching field distribution (SFD); $H_d^{-2} + H_d^0$ represents the interference from the nearest neighbors as a set of demagnetizing fields. Values for H_{max}, $\frac{dH}{dx}$, T_x and H_c are determined by a given head-media design. All of the other terms in (2.29) are considered to be normalized random variables following a Gaussian distribution with some appropriate standard deviation. The interfering fields from the nearest neighbors are determined by their previous writing condition. This means if the islands are magnetized in the same direction as the bit trailing the write head, then there fields are in unison and the demagnetizing field is absent. If the neighbors are magnetized in the opposite direction as the trailing island, then they facilitate overwriting the trailing island by

contributing an amount H_d^{-2} and/or H_d^0 to the right-side of (2.29). To help establish some bounds on the design parameters, (1.12 – 1.14) can be used as they allow for a simplified examination of the design space for BPMR from only the perspective of the write head. Considering the areal densities of interest to range from 1 to 5 Tb/in^2, Figures 2.24 and 2.25 show the relationship between the switching field distribution ΔH_c^{-1} and the effective head field gradient $\frac{dH}{dx}$ for BAR = 1 in both cases.

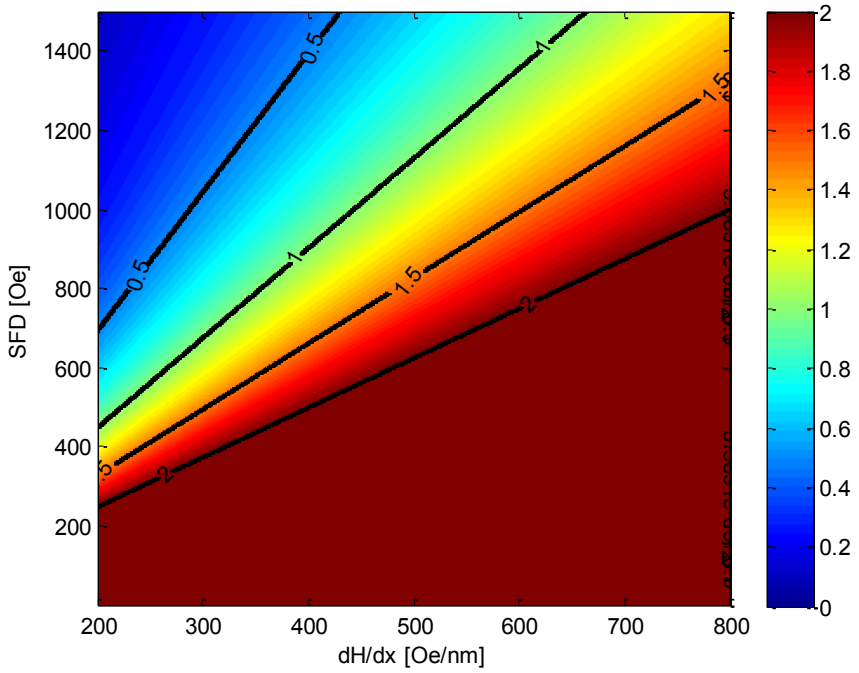

Figure 2.24 – Design chart showing the design space for areal densities ranging from 0.5 – 2 Tb/in^2. Parameter values are set at BAR = 1; $\sigma_s = 1$ nm; $\sigma_j = 1$ nm; $\mathrm{BER}_W = 10^{-6}$.

For the design chart at higher densities, the values for σ_s and σ_j have been scaled appropriately for a target areal density of 4 Tb/in^2. Both charts reflect the design tradeoff between the need for high trailing field gradients and the respective magnitude of the SFD. Sharper gradients in the head field unfortunately dictate high

corresponding levels in the switching field distribution. For example, if a trailing edge gradient of 500 Oe/nm is desired at 4 Tb/in^2, then the resulting magnitude of the SFD will approach 600 Oe.

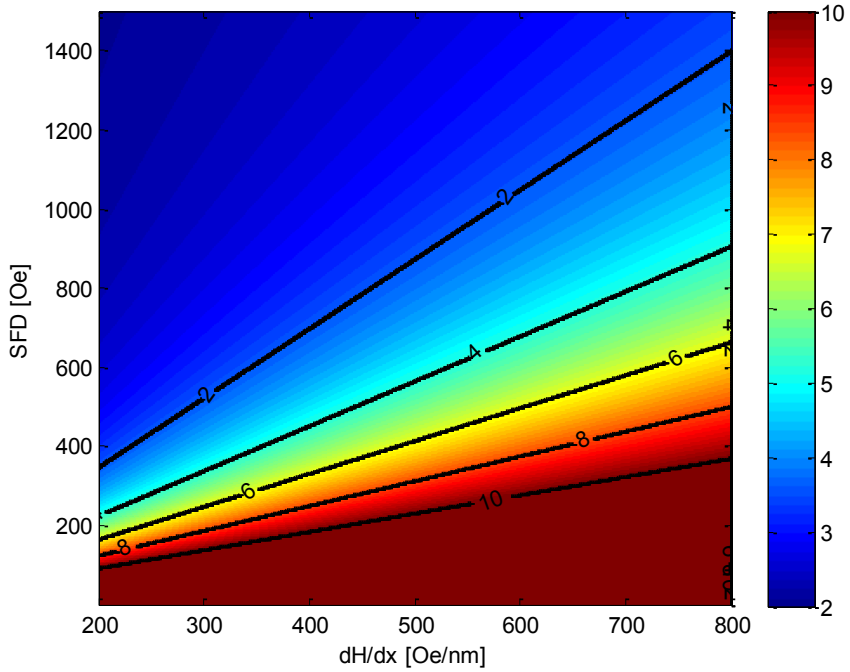

Figure 2.25 – Design chart showing the design space for areal densities ranging from 2 – 10 Tb/in^2. Parameter values are set at BAR = 1; $\sigma_s = 0.5$ nm; $\sigma_j = 0.5$ nm; $\mathrm{BER}_W = 10^{-6}$.

Based on the above design chart for an areal density of 4 Tb/in^2, the following list of plausible values can be taken: $\frac{dH}{dx} = 550$ Oe/nm; SFD = $\Delta H_c^{-1} = 600$ Oe; with SFD set to 10% of H_c, then $H_c = 6000$ Oe; $H_{max} = 8500$ Oe. With these values and the constraint imposed by (2.29), an assessment of the write error rate for substitution errors as a function of the write head's offset from the center of the target island is shown in Figure 2.26. From the graph it can be seen that the chosen parameter

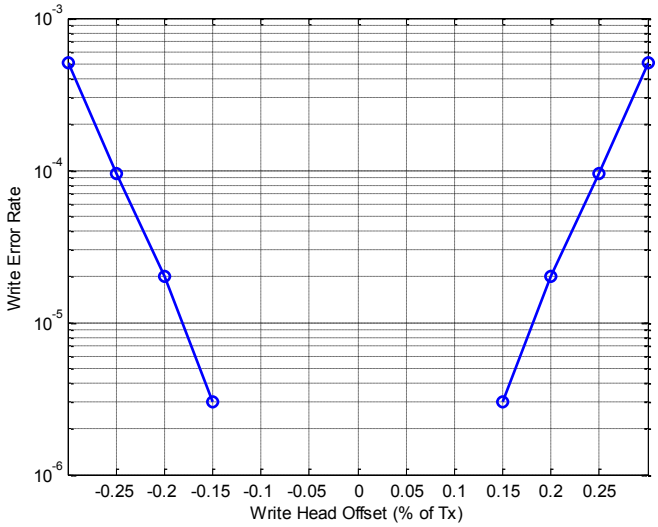

Figure 2.26 – Write error rates obtained by counting erroneously written bits as a function of the write head offset from the center of the target island. Simulations were performed with the following parameters: $\frac{dH}{dx} = 550$ Oe/nm; $H_c = 6000$ Oe; $H_{max} = 8500$ Oe; $\sigma_{Hc} = 0.07$; $\sigma_{\Delta Tx} = 0.08$; $\sigma_{Hd} = 0.04$; $T_x = 12.7$ nm.

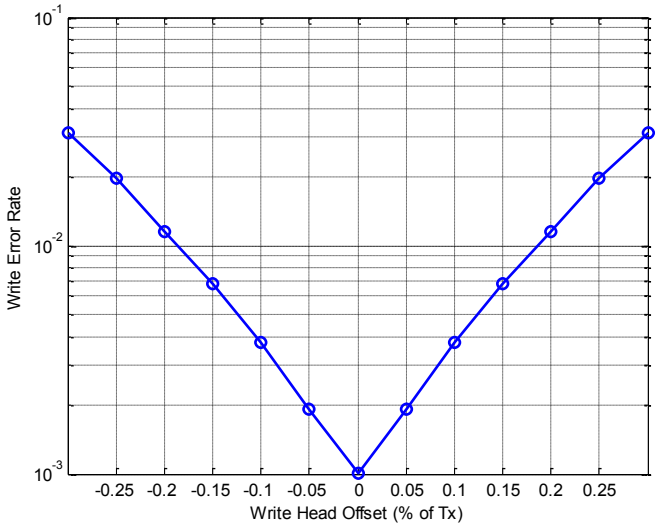

Figure 2.27 – Write error rates obtained by counting erroneously written bits as a function of the write head offset from the center of the target island. Simulations were performed with the following parameters: $\frac{dH}{dx} = 550$ Oe/nm; $H_c = 6000$ Oe; $H_{max} = 8500$ Oe; $\sigma_{Hc} = 0.14$; $\sigma_{\Delta Tx} = 0.16$; $\sigma_{Hd} = 0.08$; $T_x = 12.7$ nm.

values actually perform quite well for the BPMR channel as all write error rates for write head offsets between $\pm 0.15T_x$ have write error rates below 10^{-6}. Unfortunately, slight changes in the parameter values can result in substantially different write error rates. In Figure 2.27 very different write error rates are obtained when the standard deviations are doubled for the variations in the island position jitter, the variation in the island's coercivity, and the variation in the neighboring islands' interference. When these variances are not changed in favor of selecting different values for the magnetic fields $\frac{dH}{dx}$, H_c, and H_{max}, the write error rates show an improvement from the previous graph. This implies that the parameters selected for a given write head from the design chart in Figure 2.25 are highly sensitive in terms of the corresponding write error rate performance.

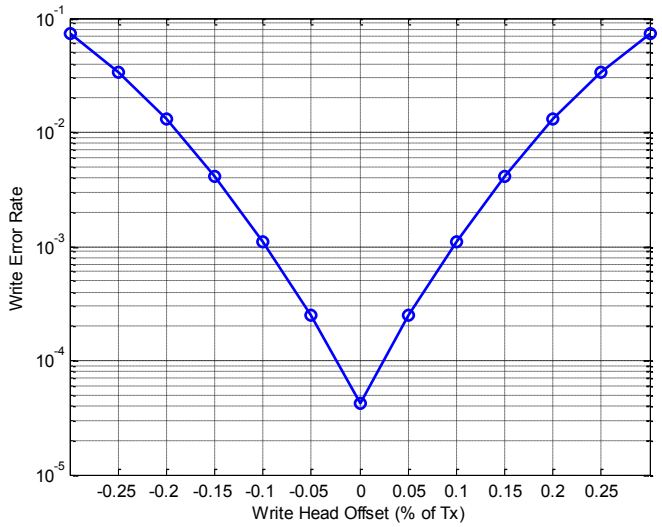

Figure 2.28 – Write error rates obtained by counting erroneously written bits as a function of the write head offset from the center of the target island. Simulations were performed with the following parameters: $\frac{dH}{dx} = 500$ Oe/nm; $H_c = 5500$ Oe; $H_{max} = 9500$ Oe; $\sigma_{Hc} = 0.07$; $\sigma_{\Delta Tx} = 0.08$; $\sigma_{Hd} = 0.04$; $T_x = 12.7$ nm.

A better strategy would be to match a set of parameters for a design to some measurements from an actual write head or sophisticated numerical model. Otherwise the variations possible for write head performance based on simulations with a given set of parameters cause the results to be somewhat questionable.

The second type of written-in error mentioned earlier can actually be the most detrimental to the channel. When write synchronization is lost, the write head can drift into a position where it starts writing on the wrong islands. Since the hardware is normally unaware of this situation, long runs of islands can be written with the incorrect bits. Written-in errors occur after the bits have been encoded and before the detection takes place, which leaves only the decoder to correct the resulting errors. Thus the written-in errors show up within the received codeword. To better quantify how insertions and deletions can affect a channel, consider the structure of a codeword c with length n,

$$c = (c_0, c_1, \cdots, c_{i-1}, c_i, c_{i+1}, \cdots, c_{n-2}, c_{n-1}) . \tag{2.30}$$

In error-free data recovery (2.30) would be both the codeword sent to and the codeword retrieved from the channel. If during the writing of the codeword, a deletion occurs at bit c_i in the codeword, then the recovered codeword would be missing this bit and all subsequent bits would be shifted one place to the left. The codeword received from the channel would then be

$$c_d = (c_0, c_1, \cdots, c_{i-1}, c_{i+1}, \cdots, c_{n-2}, c_{n-1}, c_?) , \tag{2.31}$$

and all of the bits following c_{i-1} could be in error. The bit at the end of the channel is now in an unknown state denoted by $c_?$. Now if instead bit c_j is inserted in the channel after bit c_{i-1}, then the codeword received from the channel would be

$$c_I = \left(c_0, c_1, \cdots, c_{i-1}, c_j, c_i, c_{i+1}, \cdots, c_{n-2}\right), \tag{2.32}$$

where the bits following the inserted bit c_j are all shifted one bit position to the right toward the end of the channel and the last bit c_{n-1} is lost. Now all bits following c_{i-1} could again be in error. In a BPMR channel it is also possible for insertions or deletions to occur multiple times within the same sector and in various combinations. For example, it is possible for a deletion to occur somewhere in the channel sector and then later, before reaching the end of the channel, an insertion occurs. This means that a subset of the bits comprising the codeword will likely be in error as they have effectively been shifted one bit position to the left in the codeword. The opposite situation can also take place where an insertion occurs before a subsequent deletion. With a sector size equivalent to 32Kbits, loss of write synchronization can easily cause more bit errors than conventional HDD error-correction codes can handle. As it turns out the write process for BPMR can be modeled as an interaction between three consecutive islands where the center island is the intended target to be written. When the write head possesses a write clock phase jitter, it can be modeled as a continuously accumulating variable from a Gaussian distribution with a given standard deviation based on the magnitude of the expected write clock phase error. During the course of writing to the islands in a

sector, the write head can address only a single island at a time. If the write head is in a situation where it is accumulating a drift that places it further than half of a bit period away from the target island, then an insertion or deletion can occur and one of the neighboring islands may be written to instead. As the write head accumulates phase drift, it can also start to cause substitution errors because it is closer to the center of a trench than to the center of an island. The position jitter of the islands will then increase the probability of substitution errors occurring. Five simulations of write clock phase jitter accumulation are shown in Figure 2.29.

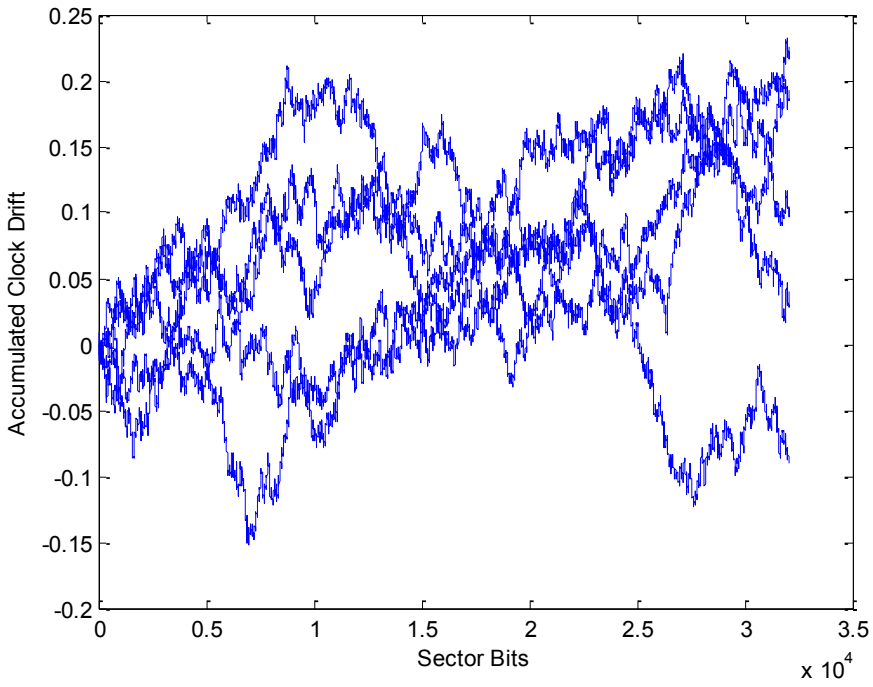

Figure 2.29 – Accumulated write clock phase error for a BPMR channel of length 32 Kbits for a total of five different simulations. The standard deviation of the write clock jitter is equal to 0.10%.

The tendency of the write clock to both accumulate and self-correct is as it passes through the channel is evident. The corresponding written-in errors for these five simulations are shown in Figure 2.30.

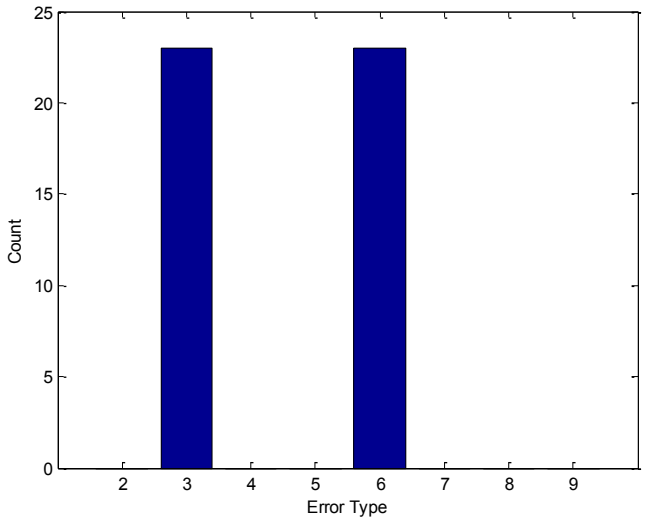

Figure 2.30 – Accumulated written-in error types for the five simulations shown in Figure 2.29. Error type 3 is an insertion and error type 6 is a deletion.

The insertions and deletions tend to occur in pairs and average more than four per sector run. Other combinations of written-in errors are possible and they do tend to occur multiple times within long sectors. The corresponding run lengths of these written-in errors will determine if the data can be recovered by the ECC [74].

2.5 Parameter Sensitivity and Selection

In developing an adequate model for BPMR a large number of parameters is required. These parameters actually help to define and support the design space utilized by engineers working on a specific media and recording head pairs. Unfortunately this also limits the scope of any modeling or simulations so that one

must be careful when examining any given results as they will be sensitive to the many parameter values specific to an areal density goal. To avoid any misinterpretations, the parameters used in subsequent sections will be limited to those specified in Table 2.1. The parameter values at 2 Tb/in^2 are from [59] and represent a more generic treatment of the BPMR readback response. The values have not been optimized to match any derived numerical pulse response. Despite this, the parameters are still widely applicable in defining a pulse response that will be heavily influenced by ISI, ITI and media noise. Thus the major degradations of the channel are adequately represented. At an area density of 4 Tb/in^2, the parameters have been optimized in some sense to a numerical model [75]. Hence the resulting pulse response and corresponding channel behavior are expected to be heavily indicative of the true channel. Regretfully, there is no way of assessing the model's accuracy since no BPMR drives exist for this density. But given these particular parameters are known, it makes sense to use them.

Chapter 3 - Detection Strategies for BPMR

Now that the write and read specifics of BPMR have been examined, it is time to introduce the signal processing responsible for processing the signal coming out of the channel. At this point the channel output is a continuous-time signal contaminated with ISI, ITI, media noise in two dimensions, electronics noise and possibly a number of written-in errors. Obviously this is far from desirable, but a number of standard detection techniques can be employed to help reshape the channel output into something more workable. At this point though, little can be done about the written-in errors as they occur outside of the scope of the detection. As shown previously in Figure 2.1, the continuous-time channel output must first be filtered and then sampled at time interval T_b to provide a digital output with signal values at the specified interval. For the initial filtering a low-pass filter matched to the channel response $h(-t)$ on the targeted track will always be used. Since the channel contains ITI, jitter noise and AWGN, these degradations are included in the signal fed to the matched filter (see Figure 3.1) where all components of the media noise are combined into a single jitter term (j_n) to help simplify the model. At this point the signal is ready for equalization and subsequent detection. The easiest method with which to begin involves equalizing a single track of data and then feeding it to a detection algorithm. In the context of TDMR techniques used with BPMR, this process is labeled single-track equalization and it forms the basis for extending conventional 1D detection into two dimensions.

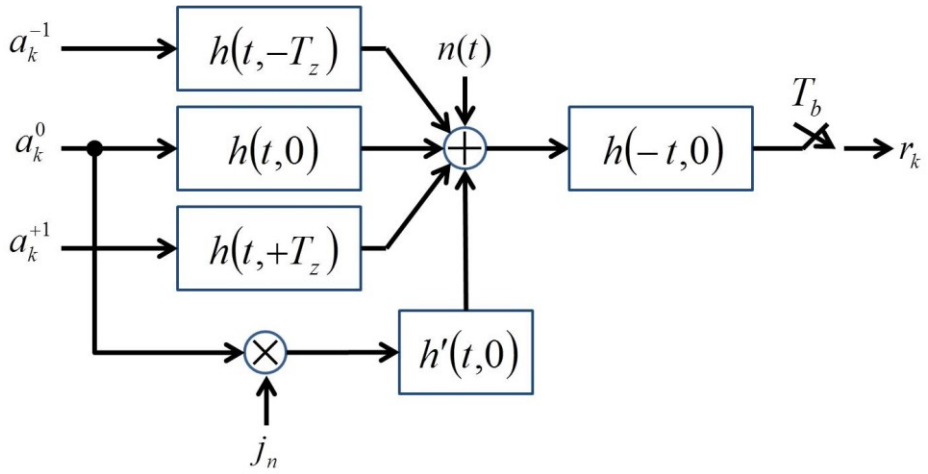

Figure 3.1 – Block diagram of a magnetic recording channel model with AWGN, first-order jitter noise, ISI, and ITI from two adjacent tracks that are input to a matched filter and then sampled at bit rate T_b.

3.1 Single-Track Equalization

Once the sampled output of the channel is available, the ISI can be dealt with by altering the signal's spectrum. In magnetic channels this is managed by using an equalizer that "equalizes" the frequency content of the sampled channel output. This is best accomplished by using a filter that maps the frequency content to a generalized-partial-response (GPR) target with non-integer coefficients [76]. The equalization process allows for some mitigation of the effects of the ISI imposed by the channel before the signal is passed off to a detector that attempts to determine which bits make up the original signal. Based on work in [77] the design of an effective equalizer begins by assuming the equalizer will be an FIR filter with $2M + 1$ coefficients (or taps) which is typically designated $\mathbf{f} = [f_{-M}, \cdots, f_0, \cdots, f_M]$. The size of the filter is often taken to be $M = 10$ which results in 21 equalizer taps, and this convention is also followed here. The goal of the equalizer is to alter the output

signal frequency to match as nearly as possible the spectrum of the desired causal target, typically designated $\mathbf{g} = [g_0, g_1, \cdots, g_{L-1}]$ which has length L. The combination of equalizer and target with the BPMR channel is shown in Figure 3.2, where the equalizer's output is designated as s_k and the target's output is denoted as $\hat{s}_k$ with k the time index.

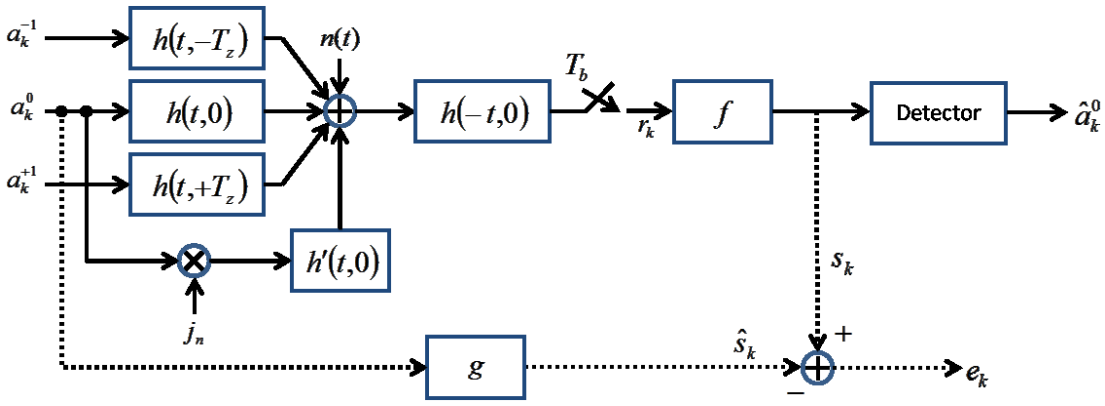

Figure 3.2 – Block diagram of BPMR under single-track equalization with AWGN, first-order jitter noise, ISI, and ITI from two adjacent tracks. The equalizer is represented by f and the GPR target is shown as g. The equalizer error is designated e_k and the APP detector output is shown as an estimation of the channel input.

The difference between the equalizer output and the target response represents the equalizer error, $e_k = s_k - \hat{s}_k$. If the equalizer were perfect, it could reproduce the target response exactly and the error would always be zero. Unfortunately this is hardly the case so the equalization must seek to minimize the error, e_k, under some constraint. As it turns out there are many ways to accomplish the minimization. Here the error will be minimized in the mean-squared-error sense and follows [4]. To begin, the equalizer mean-squared-error can be written as

$$\begin{aligned}\xi_{\mathrm{MSE}} &= E[e_k^2] \\ &= E[(s_k - \hat{s}_k)^2] \\ &= E[s_k^2] - 2E[s_k\hat{s}_k] + E[\hat{s}_k^2].\end{aligned} \tag{3.1}$$

Given that the input bits are independent, the last term can be simplified as

$$\begin{aligned}E[\hat{s}_k^2] &= E\left[\sum_{i=0}^{L}\sum_{j=0}^{L} g_i g_j a_{k-i} a_{k-j}\right] \\ &= \sum_{i=0}^{L}\sum_{j=0}^{L} g_i g_j \cdot E\left[a_{k-i} a_{k-j}\right] \\ &= \delta_{ij}\sum_{i=0}^{L}\sum_{j=0}^{L} g_i g_j \\ &= \sum_{i=0}^{L} g_i^2 = \mathbf{g}^T\mathbf{g},\end{aligned} \tag{3.2}$$

where the T of $\mathbf{g}^T$ denotes the usual matrix transpose and δ_{ij} is the Kronecker delta, equal to one with $i = j$ and zero otherwise. Now considering the middle term of the expansion of ξ_{MSE},

$$\begin{aligned}E\left[s_k\hat{s}_k\right] &= E\left[\sum_{i=-M}^{M} f_i r_{k-i}\right] E\left[\sum_{j=0}^{L} g_j a_{k-j}\right] \\ &= \sum_{i=-M}^{M}\sum_{j=0}^{L} f_i g_j \cdot E\left[r_{k-i} a_{k-j}\right].\end{aligned} \tag{3.3}$$

The expectation between the two signals r and a can be represented as a covariance matrix $\mathbf{T}$ defined as

$$\mathbf{T}_{ij} = E\left[r_{k-i}a_{k-j}\right] \quad \forall\, i = (-M, \cdots, M)\ \&\ j = (0, \cdots, L). \tag{3.4}$$

Thus the middle term of ξ_{MSE} can be expressed compactly as

$$E\left[s_k \hat{s}_k\right] = \mathbf{f}^T \mathbf{T} \mathbf{g}\,, \tag{3.5}$$

and the first term of ξ_{MSE} follows along the same lines as the last term of (3.1),

$$\begin{aligned} E[s_k^2] &= E\left[\sum_{i=-M}^{M}\sum_{j=-M}^{M} f_i f_j r_{k-i} r_{k-j}\right] \\ &= \sum_{i=-M}^{M}\sum_{j=-M}^{M} f_i f_j \cdot E\left[r_{k-i} r_{k-j}\right], \end{aligned} \tag{3.6}$$

where similar to before, the expectation between the two different signals r can be represented as a covariance matrix $\mathbf{R}$ defined as

$$\mathbf{R}_{ij} = E\left[r_{k-i}r_{k-j}\right] \quad \forall\, i,j = (-M, \cdots, M). \tag{3.7}$$

Now the first term of (3.1) can be expressed as

$$E[s_k^2] = \mathbf{f}^T \mathbf{R} \mathbf{f}. \tag{3.8}$$

This leaves the full but compact expression for the mean-squared-error as

$$\xi_{\mathrm{MSE}} = \mathbf{f}^T \mathbf{R} \mathbf{f} - 2\mathbf{f}^T \mathbf{T} \mathbf{g} + \mathbf{g}^T \mathbf{g}. \tag{3.9}$$

To minimize the above expression a Lagrange multiplier, λ, will be used with a monic constraint on $\mathbf{g}$ to avoid the all-zero solution and enforce minimization on

$$\xi_{\mathrm{MSE}} = \mathbf{f}^T\mathbf{R}\mathbf{f} - 2\mathbf{f}^T\mathbf{T}\mathbf{g} + \mathbf{g}^T\mathbf{g} - 2\lambda(\mathbf{C}\mathbf{g} - 1), \tag{3.10}$$

where the last term uses $\mathbf{C} = [1,0,\ldots,0]$ to enforce $g_0 = 1$, and the factor of 2 acts only to rescale lambda. Now it is only necessary to take the derivative with respect to f_i, g_i and λ separately and then solve each expression at zero in order to complete the minimization.

$$\frac{\partial \xi_{\mathrm{MSE}}}{\partial f_i} = 2\sum_j R_{ij}f_j - 2\sum_j T_{ij}g_j = 0\,. \tag{3.11}$$

In matrix form this can be expressed as

$$\begin{aligned} 2\mathbf{R}\mathbf{f} - 2\mathbf{T}\mathbf{g} &= 0 \\ \mathbf{R}\mathbf{f} &= \mathbf{T}\mathbf{g} \\ \mathbf{f} &= \mathbf{R}^{-1}\mathbf{T}\mathbf{g}. \end{aligned} \tag{3.12}$$

The process continues by minimizing with respect to g_i

$$\frac{\partial \xi_{\mathrm{MSE}}}{\partial g_i} = -2\sum_j f_j T_{ij} + 2g_i - 2\lambda C_i = 0, \tag{3.13}$$

where again in matrix form and with substitution for $\mathbf{f}$ from before,

$$-2\mathbf{T}^T\mathbf{f} + 2\mathbf{g} - 2\lambda\mathbf{C}^T = 0 \tag{3.14}$$

$$-2\mathbf{T}^T\mathbf{R}^{-1}\mathbf{T}\mathbf{g} + 2\mathbf{g} - 2\lambda\mathbf{C}^T = 0$$

$$\mathbf{g} - \mathbf{T}^T\mathbf{R}^{-1}\mathbf{T}\mathbf{g} = \lambda\mathbf{C}^T$$

$$\mathbf{g} = (\mathbf{I} - \mathbf{T}^T\mathbf{R}^{-1}\mathbf{T})^{-1}\lambda\mathbf{C}^T,$$

where $\mathbf{I}$ is just an identity matrix. Finally the last term is computed for the Lagrange multiplier with $\mathbf{g}$ substituted from the previous result,

$$\frac{\partial \xi_{\text{MSE}}}{\partial \lambda} = -2(Cg - 1) = 0$$

$$\mathbf{C}\mathbf{g} = 1$$

$$\mathbf{C}(\mathbf{I} - \mathbf{T}^T\mathbf{R}^{-1}\mathbf{T})^{-1}\lambda\mathbf{C}^T = 1 \tag{3.15}$$

$$\lambda = \frac{1}{\mathbf{C}(\mathbf{I} - \mathbf{T}^T\mathbf{R}^{-1}\mathbf{T})^{-1}\mathbf{C}^T}.$$

Thus the expressions for $\mathbf{f}$, $\mathbf{g}$ and λ provide the MMSE solution for the equalizer and target under the monic constraint based on the inputs $\mathbf{r}$ and $\mathbf{a}$.

3.2 The BCJR Algorithm

The BCJR algorithm originally proposed in [50] is a maximum *a posteriori* (MAP) probability detection scheme. It operates on a trellis that represents the possible state transitions for the symbols input to the channel. For binary transmission individual symbols and bits are one and the same. So for a given binary input a_k at time k, the BCJR algorithm calculates the *a posteriori* probability (APP) that bit a_k is

either -1 or $+1$ based on knowledge of the trellis, the received signal at the output of the equalizer, and any *a priori* probability information made available on the bits in the codeword. It accomplishes this by computing a log likelihood ratio (LLR) for every bit within the received codeword. The LLR can be defined as

$$\mathrm{LLR}_k \triangleq \log\left[\frac{P(a_k = +1|\mathbf{r})}{P(a_k = -1|\mathbf{r})}\right]. \tag{3.16}$$

Based on knowledge of the trellis, the LLR can be restated in more applicable terms,

$$\mathrm{LLR}_k = \log\left[\frac{\sum_{S+} p(s_{k-1} = s', s_k = s,\mathbf{r})p(\mathbf{r})}{\sum_{S-} p(s_{k-1} = s', s_k = s,\mathbf{r})p(\mathbf{r})}\right]. \tag{3.17}$$

Here s_k is the trellis state at time k, s_{k-1} is the previous state at time k, $\mathbf{r}$ is the received codeword, S^+ is the set containing all pairs (s', s) that coincide with all possible state transitions in the trellis of $(s_{k-1} = s') \rightarrow (s_k = s)$ under the assumption that $a_k = +1$, and S^- is the equivalent set based on the assumption that $a_k = -1$. The common term $p(\mathbf{r})$ in the summations can be cancelled leaving the shorthand pdf expression $p(s', s,\mathbf{r})$ in the numerator and denominator. Through more applications of Bayes' rule, this joint probability can be further factored into the product of three terms,

$$p(s', s,\mathbf{r}) = \alpha_{k-1}(s')\gamma_k(s', s)\beta_k(s). \tag{3.18}$$

The term $\alpha_{k-1}(s')$ represents a forward recursion over the previous states prior to the current state and can be computed as

$$\alpha_k(s) = \sum_{s'} \gamma_k(s', s)\alpha_{k-1}(s'). \tag{3.19}$$

The term $\beta_k(s)$ represents a backward recursion over the future states ahead of the current state and can be calculated as

$$\beta_{k-1}(s') = \sum_{s} \gamma_k(s', s)\beta_k(s). \tag{3.20}$$

The term $\gamma_k(s', s)$ represents the branch transition probability that must be computed for every branch connecting consecutive states within the trellis. It can be written as

$$\begin{aligned} \gamma_k(s', s) &= \frac{P(s', s)}{P(s')} \cdot \frac{p(s', s, r_k)}{P(s', s)} \\ &= P(s|s')p(r_k|s', s), \end{aligned} \tag{3.21}$$

where $P(s|s')$ is the *a priori* probability that bit a_k was the input that resulted in a transition from state s_{k-1} to state s_k. Figure 3.3 illustrates an example section for a trellis with four states showing the relationship between the forward and backward metrics and the branch transition. In the presence of AWGN with variance $\sigma^2 = \frac{N_0}{2}$,

$$p(r_k|s', s) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\frac{\left\|r_k - r_{s',s}\right\|^2}{\sigma^2}\right], \tag{3.22}$$

where $r_{s',s}$ represents the expected output from the transition connecting branch $s' = s_{k-1}$ to $s = s_k$ in the absence of noise assuming a_k was the corresponding

input of the channel. Thus the LLRs can be computed by first initializing the α's and the β's, calculating all branch metrics, computing the forward and backward metrics, and then calculating LLRs for each bit in the codeword.

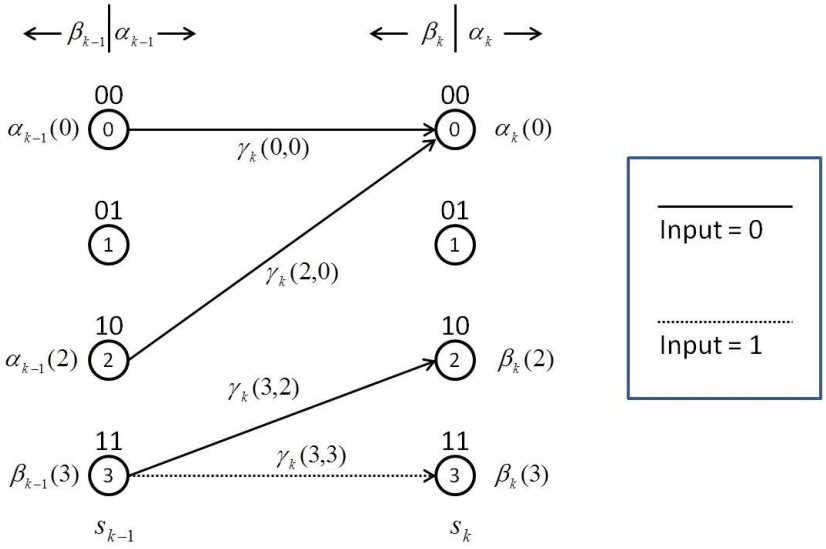

Figure 3.3 – Example of forward and backward recursions shown on a reduced 4-state trellis for binary inputs with discrete time unit k. Not all possible branches and corresponding branch transitions are shown.

Typically the BCJR algorithm is performed in the logarithm domain to avoid numerical instability while also simplifying implementation. This slightly modifies the earlier expressions for calculating the LLRs:

$$\breve{\alpha}_k(s) = \log \sum_{s'} \exp[\breve{\gamma}_k(s', s) + \breve{\alpha}_{k-1}(s')] \tag{3.23}$$

$$\breve{\beta}_{k-1}(s') = \log \sum_{s} \exp\left[\breve{\gamma}_k(s', s) + \breve{\beta}_k(s)\right] \tag{3.24}$$

$$\check{\gamma}_k(s',s) = \log[P(s|s')] - \frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2}\frac{\left\|r_k - r_{s',s}\right\|^2}{\sigma^2}, \tag{3.25}$$

where $\check{\alpha}_k$, $\check{\beta}_k$ and $\check{\gamma}_k$ designate the log-domain versions of α_k, β_k and γ_k. During implementation the algorithm is usually constrained to start and stop at the zero state of the trellis and the forward and backward metrics are initialized by,

$$\check{\alpha}_0(s) = \begin{cases} 0, & s = 0 \\ -\infty, & s \neq 0 \end{cases} \tag{3.26}$$

$$\check{\beta}_T(s) = \begin{cases} 0, & s = 0 \\ -\infty, & s \neq 0' \end{cases} \tag{3.27}$$

where the subscript T on $\check{\beta}$ denotes the length of the expanded trellis terminating at the zero state and the $(-\infty)$ value is substituted with some suitably large negative value. Substituting with the log domain versions $\check{\alpha}_k$, $\check{\beta}_k$ and $\check{\gamma}_k$ in conjunction with the appropriate initializations, the LLRs are calculated as

$$\begin{aligned} \mathrm{LLR}_k &= \log\left[\frac{\sum_{S+}\alpha_{k-1}(s')\gamma_k(s',s)\beta_k(s)}{\sum_{S-}\alpha_{k-1}(s')\gamma_k(s',s)\beta_k(s)}\right] \\ &= \log\left[\sum_{S+}\exp\left(\check{\alpha}_{k-1}(s')+\check{\gamma}_k(s',s)+\check{\beta}_k(s)\right)\right] \\ &\quad - \log\left[\sum_{S-}\exp\left(\check{\alpha}_{k-1}(s')+\check{\gamma}_k(s',s)+\check{\beta}_k(s)\right)\right]. \end{aligned} \tag{3.28}$$

As a decoder the BCJR algorithm can return the LLRs as the soft information on the bits based on the channel output or it can make a hard decision based on the LLRs. For hard decisions on the bits, the BCJR algorithm's outputs are made by

$$\hat{a}_k = \text{sign}(L_k), \tag{3.29}$$

where $\hat{a}_k = +1$ when $[P(\hat{a}_k = +1|\mathbf{r}) > P(\hat{a}_k = -1|\mathbf{r})]$ and $\hat{a}_k = -1$ otherwise. While the full implementation of the BCJR algorithm is preferred, there have been several attempts to reduce its overall complexity without significantly sacrificing decoder performance [78], [79]. In this work the log-domain version is the preferred implementation.

3.3 TDMR with Multi-Track Detection Techniques

In a high ITI environment like BPMR, the interference of the surrounding tracks in combination with ISI can substantially degrade the detection ability of a one-dimensional approach. Thus it only makes sense to explore extending 1D detection strategies into 2D equivalent methods. As it turns out, this is precisely one of the strategies behind TDMR which was introduced in Chapter 1. One obvious way to proceed is to expand the 1D equalization and detection for a single track to cover multiple tracks. There are also a number of different ways in which this can be done. The development here will follow the methods outlined in [46] and [66] as these sources outline the most natural progression from the 1D single-track equalization previously discussed. As noted in [46] the expansion from 1D into 2D detection is best done by utilizing multi-track detection (MTD) techniques where information from the adjacent side tracks must be acquired prior to processing the readback signal from the target track. This dictates a redesign of the physical read head so that multiple-track sensing is possible. While this seems like a difficult requirement,

TDMR may likely be required to expand with this functionality anyway [80] and multiple-head designs for HDDs have been considered as a future technology option for some time now [81]. In terms of detection performance, it was shown in [46] that using TDMR techniques on data acquired from only a single track produces substantially less performance gain when compared to MTD. Thus, MTD will be used here where the goal is only to recover the bits on a single targeted track and not to attempt readings with data recovery on multiple tracks simultaneously. Multiple-track data recovery is beyond the scope of the current discussion.

3.3.1 MTD with 2D Equalization

To perform 2D equalization with information from multiple tracks, a substantial expansion from the 1D equalization depicted in Figure 3.2 will be required. Given that previous modeling of the ITI showed that its influence spreads across two rows of adjacent tracks, it is only necessary to sense five adjacent tracks (the target track plus two sets of side tracks) in order to adequately capture the extent of the ITI influence on the center track. Thus an expansion to 2D equalization would appear as shown in Figure 3.4. With the help of this diagram, a derivation of the optimum equalizers and targets in the mean-square-error sense can be formulated as an extension of the single-track equalization case. Here the 2D equalizer error can be expressed as

$$e_k = \mathbf{f}^T\mathbf{r} - \mathbf{g}^T\mathbf{a}\,, \tag{3.30}$$

where the notation follows from before and $\mathbf{f}$, $\mathbf{r}$, $\mathbf{g}$ and $\mathbf{a}$ are all vectors in matrix form.

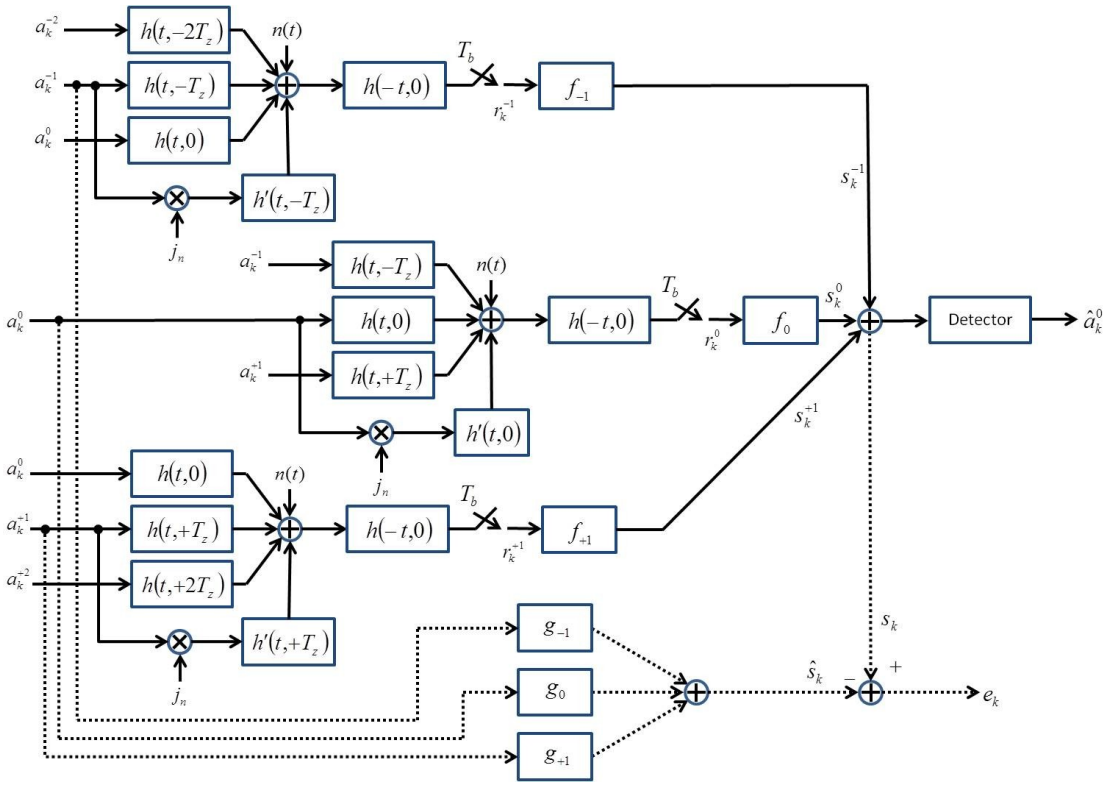

Figure 3.4 – Block diagram of BPMR under multi-track detection of five adjacent tracks with 2D equalization, AWGN, first-order jitter noise, ISI, and ITI from two adjacent tracks. The equalizers are represented by f and the GPR targets are shown as g. The equalizer error is designated e_k and the APP detector output is shown as an estimation of the channel input.

By concatenating the various 1D equalizer and target coefficients describing their respective FIR filters, the representative vectors in 2D become

$$\mathbf{f}_{-1} = \left[f_{-1,-M}, \cdots, f_{-1,0}, \cdots, f_{-1,M}\right] \tag{3.31}$$

$$\mathbf{f}_{0} = \left[f_{0,-M}, \cdots, f_{0,0}, \cdots, f_{0,M}\right] \tag{3.32}$$

$$\mathbf{f}_{+1} = \left[f_{+1,-M}, \cdots, f_{+1,0}, \cdots, f_{+1,M}\right] \tag{3.33}$$

$$\mathbf{f} = [\mathbf{f}_{-1} \quad \mathbf{f}_0 \quad \mathbf{f}_{+1}] \tag{3.34}$$

for the equalizers where their corresponding lengths are taken to be $2M + 1$ taps as usual. Then for the targets

$$\mathbf{g}_{-1} = [g_{-1,0}, \cdots, g_{-1,L_2-1}] \tag{3.35}$$

$$\mathbf{g}_0 = [g_{0,0}, \cdots, g_{0,L_1-1}] \tag{3.36}$$

$$\mathbf{g}_{+1} = [g_{+1,-M}, \cdots, g_{+1,L_2-1}] \tag{3.37}$$

$$\mathbf{g} = [\mathbf{g}_{-1} \quad \mathbf{g}_0 \quad \mathbf{g}_{+1}], \tag{3.38}$$

where the lengths of the side-track targets are designated L_2 and the length of the center-track target is designated L_1. Note that L_2 does not have to equal L_1. The respective channel inputs passed to each target are also arranged in a corresponding format

$$\mathbf{a}_{-1} = [a_{-1,k}, \cdots, a_{-1,k-L_2+1}] \tag{3.39}$$

$$\mathbf{a}_0 = [a_{0,k}, \cdots, a_{0,k-L_1+1}] \tag{3.40}$$

$$\mathbf{a}_{+1} = [a_{+1,k}, \cdots, a_{+1,k-L_2+1}] \tag{3.41}$$

$$\mathbf{a} = [\mathbf{a}_{-1} \quad \mathbf{a}_0 \quad \mathbf{a}_{+1}], \tag{3.42}$$

where k is the time index. The outputs of the channel that are passed to the equalizers follow

$$\mathbf{r}_{-1} = \left[r_{-1,k+M}, \cdots, r_{-1,k}, \cdots, r_{-1,k-M}\right] \tag{3.43}$$

$$\mathbf{r}_{0} = \left[r_{0,k+M}, \cdots, r_{0,k}, \cdots, r_{0,k-M}\right] \tag{3.44}$$

$$\mathbf{r}_{+1} = \left[r_{+1,k+M}, \cdots, r_{+1,k}, \cdots, r_{+1,k-M}\right] \tag{3.45}$$

$$\mathbf{r} = \left[\mathbf{r}_{-1} \quad \mathbf{r}_{0} \quad \mathbf{r}_{+1}\right], \tag{3.46}$$

where k is the same time index as for the GPR target inputs. With the given model the 2D equalizer error, $e_k = \mathbf{f}^T\mathbf{r} - \mathbf{g}^T\mathbf{a}$, needs to be minimized in the same way as for the 1D case in single-track detection. Thus the minimization of the mean-squared-error takes the form

$$\xi_{\mathrm{MSE}} = E[e_k^2] = \mathbf{f}^T\mathbf{R}\mathbf{f} - 2\mathbf{f}^T\mathbf{T}\mathbf{g} + \mathbf{g}^T\mathbf{g}, \tag{3.47}$$

where $\mathbf{R} = E[\mathbf{r}\mathbf{r}^T]$ and $\mathbf{T} = E[\mathbf{r}\mathbf{a}^T]$. A monic constraint is again enforced with $g_{0,0} = 1$ such that

$$\mathbf{C} = [0, \cdots, 1, \cdots, 0], \tag{3.48}$$

with length equal to $L_1 + 2L_2$ so as to match the size of $\mathbf{g}$. The use of a Lagrange multiplier λ to enforce minimization results in

$$\begin{aligned} \lambda &= \frac{1}{\mathbf{C}(\mathbf{I} - \mathbf{T}^T\mathbf{R}^{-1}\mathbf{T})^{-1}\mathbf{C}^T} \\ \mathbf{g} &= (\mathbf{I} - \mathbf{T}^T\mathbf{R}^{-1}\mathbf{T})^{-1}\lambda\mathbf{C}^T \\ \mathbf{f} &= \mathbf{R}^{-1}\mathbf{T}\mathbf{g}, \end{aligned} \tag{3.49}$$

which is also consistent with the work of [82]. This development allows for different length GPR targets between the center and side tracks. Before passing the 3-track equalized channel output, $\mathbf{r} = [\mathbf{r}_{-1} \quad \mathbf{r}_0 \quad \mathbf{r}_{+1}]$ to the BCJR detector, the symmetry of the channel outputs imposed by the equalizers can be used to reduce the expansion of the states needed by the BCJR algorithm.

3.3.2 Expanded-State BCJR for TDMR

When the equalization makes use of 2D equalizers, 2D targets or both, the standard 1D BCJR algorithm must be modified to accommodate the added dimensionality. Since the BCJR operates on a trellis which models the possible state transitions for the polar binary inputs $\{-1, +1\}$, the trellis must be expanded for the additional inputs from tracks a_{-1} and a_{+1}. The symmetry of the channel model results in identical equalizers and targets for both adjacent tracks a_{-1} and a_{+1}, meaning that $\mathbf{f}_{-1} = \mathbf{f}_{+1}$ and $\mathbf{g}_{-1} = \mathbf{g}_{+1}$. This implies that their respective contributions to the detection can be combined as $a_+ = a_{-1} + a_{+1} = \{-2,0,2\}$ which is now a single ternary input as opposed to two separate binary inputs [46]. Thus, instead of expanding the number of states in the trellis to a total of $2^{L_1-1} \times 2^{L_2-1} \times 2^{L_2-1}$ states for inputs a_0, a_{-1} and a_{+1}, the trellis needs to only be expanded to a total of $2^{L_1-1} \times 3^{L_2-1}$ states for inputs a_0 and a_+. The number of required branches connecting consecutive states will also be reduced from eight $(2 \times 2 \times 2)$ per state to only six (2×3) per state. Given the BCJR algorithm is fairly complex when only one input is expected, the opportunity to reduce states is well advised for the 2D trellis.

This still leaves the 1D BCJR algorithm to be modified to correctly implement the expanded trellis. Modifications to the standard BCJR algorithm based on an expanded trellis for 2D equalization targets where explored in [83] and the same methodology is used here. For a trellis with the two inputs a_0 and a_+ only the branch metric calculation within the BCJR needs to be modified, which is typical of other trellis state expansions such as those used in noise-predictive techniques [84]. The modified branch transition probability takes the form

$$\gamma(s_{k-1}, s_k) = P\left(a_{0,k}, a_{+,k}\right) \cdot P\left(r_k | s_{k-1}, s_k, a_{0,k}, a_{+,k}\right), \tag{3.50}$$

where (s_{k-1}, s_k) indicates the previous and next state of a specific trellis branch and $P\left(a_{0,k}, a_{+,k}\right)$ is the *a priori* probability with $P\left(a_{0,k}, a_{+,k}\right) = P\left(a_{0,k}\right)P\left(a_{+,k}\right)$. For the ternary input the following holds by definition

$$\begin{aligned} a_{+,k} = -2 \Longrightarrow P\left(a_{+,k} = -2\right) \\ = P\left(a_{-1,k} = -1\right)P\left(a_{+1,k} = -1\right) \end{aligned} \tag{3.51}$$

$$\begin{aligned} a_{+,k} = +2 \Longrightarrow P\left(a_{+,k} = +2\right) \\ = P\left(a_{-1,k} = +1\right)P\left(a_{+1,k} = +1\right) \end{aligned} \tag{3.52}$$

$$\begin{aligned} a_{+,k} = 0 \Longrightarrow P\left(a_{+,k} = 0\right) \\ = P\left(a_{-1,k} = -1\right)P\left(a_{+1,k} = +1\right) \\ + P\left(a_{-1,k} = +1\right)P\left(a_{+1,k} = -1\right), \end{aligned} \tag{3.53}$$

The *a posteriori* probability for the two inputs is then given by

$$P(a_{0,k}, a_{+,k}|\mathbf{r}) = \frac{1}{P(\mathbf{r})} \sum_{s_{k-1}} \alpha_{k-1}(s_{k-1}) \times \gamma(s_{k-1}, s_k) \times \beta(s_k), \tag{3.54}$$

with $\mathbf{r}$ being the corresponding block of the equalizer output. The APP is then marginalized over the ternary input to compute the desired APP for $a_{0,k}$ as

$$\text{APP}(a_{0,k}) = P(a_{0,k}|\mathbf{r}) = \sum_{a_{+,k}} P(a_{0,k}, a_{+,k}|\mathbf{r}) \,. \tag{3.55}$$

This completes the detector design and implementation from the 2D equalizer and target through to the output of the detector's modified BCJR algorithm where one would expect improved performance. In [46] a substantial detector performance gain was obtained using 2D equalization where it was also noted that full 2D equalization only performs slightly better than a less complex version known as joint-track equalization.

3.3.3 Joint-Track Equalization with MTD

In joint-track equalization the equalizer is one-dimensional while the targets are two-dimensional. Thus the 1D equalizer is forced to mimic the frequency content of a desired 2D target response [85]. This simplifies the overall detector design as the equalizers for the side-track data labeled as f_{-1} and f_{+1} in Figure 3.4 for 2D equalization are zeroed out. Thus, equations (3.47) – (3.49) still hold for the design

of the equalizer. In joint-track equalization the target response can be viewed as an equivalent 3×3 matrix **G**

$$\mathbf{G} = \begin{bmatrix} g_{-1} \\ g_0 \\ g_{+1} \end{bmatrix} = \begin{bmatrix} g_{-1,-1} & g_{-1,0} & g_{-1,+1} \\ g_{0,-1} & g_{0,0} & g_{0,+1} \\ g_{+1,-1} & g_{+1,0} & g_{+1,+1} \end{bmatrix}, \tag{3.56}$$

used in conjunction with a corresponding representation for the equalizer **F** in the MMSE process,

$$\mathbf{F} = \begin{bmatrix} f_{-1} \\ f_0 \\ f_{+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ f_{0,-1} & f_{0,0} & f_{0,+1} \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.57}$$

where the equalizer's 1D constraint is apparent. Care must be taken when using joint-track equalization because it does not result in better detection performance than single-track equalization when *a priori* information from the side tracks is not provided. Thus it must be used with some form of MTD so as to provide prior knowledge on the adjacent tracks before initiating detection on the target track [46], [86]. When MTD is paired with joint-track equalization, the technique provides near 2D equalization detection performance even when an ML detector is used [87].

3.4 Noise Prediction

While TDMR techniques have shown considerable promise for BPMR, they require much greater complexity and in the case of multi-track detection can dictate difficult redesigns of the read head. They also concentrate on managing the high levels of ITI and ISI in the channel while mostly ignoring negative contributions of the media

noise and AWGN. A better strategy would be to also incorporate some sort of correction for the media noise and AWGN. As it turns out, a very powerful technique known as noise prediction emphasizes signal correction based on the characteristics of the noise after equalization. Noise prediction was originally employed as a maximum likelihood improvement to the computation of the branch metric in the Viterbi detector as a means of correcting the coloration of the noise by the equalizer [88], [89]. For a zero-mean Gaussian-Markov signal-dependent noise process, [90], [91] derived the optimum maximum-likelihood sequence detection (MLSD) solution for the VA branch metric. In [92] this method was thoroughly expounded in light of pattern-dependent noise prediction (PDNP) where it was paired with soft-output detectors, most notably the BCJR algorithm. Since a BCJR detector is used here, the work of [92] is closely followed to apply PDNP to the case of single-track equalization for BPMR.

To begin, the output of the equalizer is assumed to be $\mathbf{y}(\mathbf{b}) = \mathbf{s}(\mathbf{b}) + \mathbf{n}(\mathbf{b})$ where $\mathbf{s}(\mathbf{b})$ is the signal output of the equalizer and $\mathbf{n}(\mathbf{b})$ is the corresponding noise. Both signal and noise are considered to be dependent on the input bit pattern $\mathbf{b}$. For MLSD an input bit sequence is found that maximizes the conditional probability of receiving $\mathbf{y}(\mathbf{b})$ given $\mathbf{b}$ where $\mathbf{y}(\mathbf{b})$, $\mathbf{s}(\mathbf{b})$ and $\mathbf{n}(\mathbf{b})$ all have length N,

$$p(\mathbf{y}|\mathbf{b}) = p(y_N, \cdots, y_1|\mathbf{b}) \,. \tag{3.58}$$

If the noise is assumed Gaussian-Markov then the mean of $\mathbf{y}(\mathbf{b})$ can be substituted by the conditional probability mean which in turn can be determined by the optimal

linear prediction of the noise $\mathbf{n}(\mathbf{b})$. The variance follows the same argument and this allows for

$$p(\mathbf{y}|\mathbf{b}) = \frac{1}{\sqrt{2\pi\sigma_p^2(\mathbf{b})}} \exp\left[-\frac{1}{2}\frac{\left(y_k - \mu_p(\mathbf{b})\right)^2}{\sigma_p^2(\mathbf{b})}\right], \tag{3.59}$$

where $\mu_p(\mathbf{b})$ is the optimal linear prediction of y_k for time index k and $\sigma_p^2(\mathbf{b})$ is the error variance of the predictor. Thus the linear prediction and error variance must be determined. For Markov noise of order J they are given as follows,

$$\mu_p(\mathbf{b}) = \hat{n}_k(\mathbf{b}) + s_k = \mathbf{q}(\mathbf{b})\boldsymbol{n}^T + s_k, \tag{3.60}$$

with $\mathbf{q(b)}$ the row vector of J predictor coefficients and $\boldsymbol{n}^T = \left[n_{k-1}, \cdots, n_{k-J}\right]^T$. To solve for the predictor coefficients one must apply the Yule-Walker equation to each specified bit pattern $\mathbf{b}$ such that

$$\mathbf{q(b)} = \mathbf{c}^T(\mathbf{b})\mathbf{R}(\mathbf{b})^{-1}, \tag{3.61}$$

where $\mathbf{c(b)} = E[n_k\mathbf{n}]$ and $\mathbf{R(b)} = E[\mathbf{n}\mathbf{n}^T]$. The corresponding predictor error variance can be found from

$$\sigma_p^2(\mathbf{b}) = \sigma^2(\mathbf{b}) - \mathbf{c}^T(\mathbf{b})\mathbf{R}(\mathbf{b})^{-1}\mathbf{c}(\mathbf{b}), \tag{3.62}$$

with $\sigma^2(\mathbf{b}) = E[n_k^2]$. This development, while for MSLD, easily extends into soft-output detectors where the APP on each bit is the desired metric [93]. This is accomplished quite easily by modifying the branch metric from the BCJR algorithm where, as before, the derivation of [92] is preferred. This is very similar to the

expansion of the trellis within MTD for BPMR where two-dimensional information was passed to a trellis with expanded states. When the trellis was previously expanded for MTD, the forward and backward recursions of the BCJR algorithm did not require modification and only the branch metric necessitated changes. The same situation applies here where the BCJR branch metric can be computed as

$$\begin{aligned}&\gamma_k(s_{k-1}, s_k)\\&= \frac{1}{\sqrt{2\pi\sigma_p^2(s_{k-1}, s_k)}} \exp\left[-\frac{1}{2}\frac{(y_k - s_k - \hat{n}_k(s_{k-1}, s_k))^2}{\sigma_p^2(s_{k-1}, s_k)}\right]\\&\times P(b_k)\,,\end{aligned} \tag{3.63}$$

where (s_{k-1}, s_k) indicates the previous and next state of a specific trellis branch, $\hat{n}_k(s_{k-1}, s_k)$ and $\sigma_p^2(s_{k-1}, s_k)$ represent the noise prediction and its error variance for branch $\gamma_k(s_{k-1}, s_k)$ and bit pattern b_k. As before, a substitution is made for $\hat{n}_k(s_{k-1}, s_k)$,

$$\hat{n}_k(s_{k-1}, s_k) = \sum_{j=1}^{J} q_j\,(s_{k-1}, s_k) n_{k-j}(s_{k-1}, s_k), \tag{3.64}$$

so that the noise predictors and corresponding error variances can be obtained from (3.20) and (3.21) for each branch transition of the trellis. To use noise prediction in practice, the length of the Markov noise process J, and an effective spread of past and future bits, denoted by P and F, must be determined to make the method effective. This means that parameters for each must be established and these

parameters will determine the expansion of the trellis used in the BCJR algorithm. Normally the standard 1D BCJR algorithm requires a trellis with 2^L states where L is one less than the memory of the GPR target paired with the equalizer. With noise prediction this expands to $2^{\max(P,L+J)+F}$ states as determined by the length of the corresponding bit pattern. When the underlying noise processes are not Gaussian-Markov of order J, the noise prediction is suboptimal but it can still produce considerable gains in detector performance [94], [95] depending on the true nature of the noise in the channel. Even in the situation where the noise from the channel possesses a strong characteristic such as a mean value, PDNP can be modified to capture this characteristic of the channel and still provide substantial detection improvement [96]. The overall complexity of the noise-predictive method depends on the number of predictor taps, the number of predictors selected and the number of trellis branches. Since each of these choices plays a part in the expansion of the overall trellis, sub-optimal selections are often made to reduce the complexity added to the detection design at the cost of detector performance. When noise prediction is extended to two-dimensions, the problem of additional complexity becomes even more significant as the detector's 2D equalization and APP scheme are already sub-optimized to make them tractable for implementation [97], [98]. Thus, productive implementation of noise-predictive techniques in multiple dimensions remains an open problem.

3.5 Simulations with MTD

In order to assess the various detection techniques previously discussed on the BPMR channel, a set of simulations were designed to compare different strategies. Initially the areal density is held at 1.61 Tb/in^2 and coincides to square islands with 11 nm side lengths and bit period and track pitch equal to 22 nm. The remaining parameters affecting the read head and readback pulse response coincide with the first column of Table 2.1. A fair comparison between single-track equalization, MTD and a bound are used in the various simulations. In single-track equalization the 1D target has length three and is denoted SE-3. For MTD the span of the ITI coverage consists of five adjacent tracks with target track a_0 and two virtual side tracks $a_{\pm 1}$ and $a_{\pm 2}$. The GPR target has length three on the center track, length two on the first virtual track and length one on the second virtual track, which is denoted MTD-321. For MTD the center track is always equalized using joint-track equalization while the side tracks use single-track equalization. To establish a basis with which to compare the performance of MTD on the BPMR channel, a bound is estimated by using MTD with equivalent target lengths assuming all side-track information is error free. Initially only ITI, ISI and AWGN are present. Then media noise is imposed on the channel. First, a determination on the difference between the hexagonal and rectangular patterns is attempted to see if one performs better than the other under the various detection schemes. This is managed by starting with the initial areal density and then reducing T_x and T_y while maintaining a BAR = 1, which effectively scales the density at the cost of increasing the ISI and ITI present in the channel. In

Figure 3.5 it is worth noting that the rectangular and hexagonal patterns both perform similarly under single-track equalization, but as the ITI and ISI become increasingly problematic, single-track equalization does not work well for staggered islands (see Figures 3.6 – 3.8).

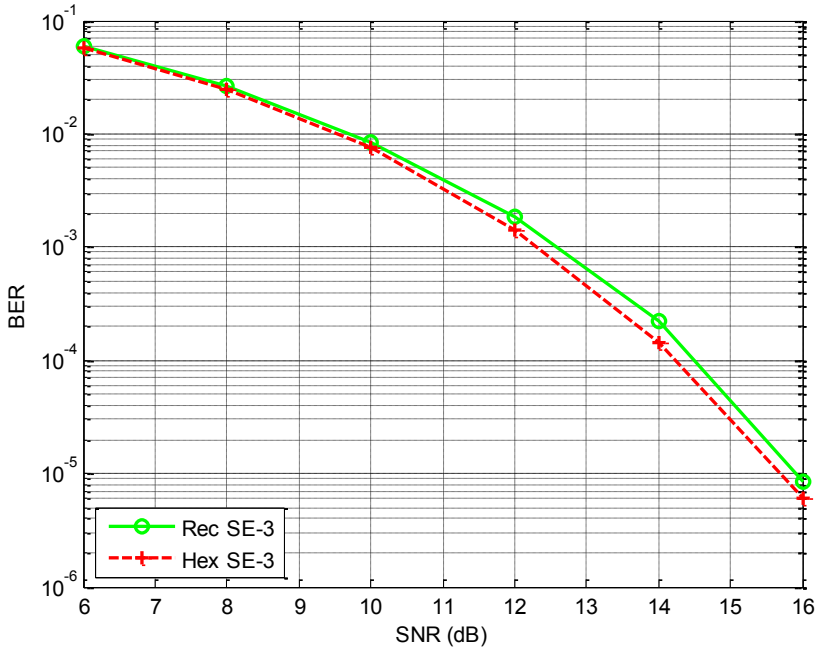

Figure 3.5 – BPMR channel detection with single-track equalization at an areal density of 1.61 Tb/in^2 $\left(T_x = T_y = 20\text{nm}\right)$ in the presence of ISI, ITI and AWGN. “Rec” refers to a rectangular array of islands while “Hex” refers to a hexagonal array.

In Figures 3.6 – 3.8 some interesting patterns emerge. Despite the poor performance under single-track equalization for high interference environments, the hexagonal island pattern always outperforms the rectangular arrangement under MTD although this gap does not increase substantially as the channel interference increases. Also the bounds between the two patterns are nearly the same with the rectangular pattern only just slightly better than the hexagonal pattern.

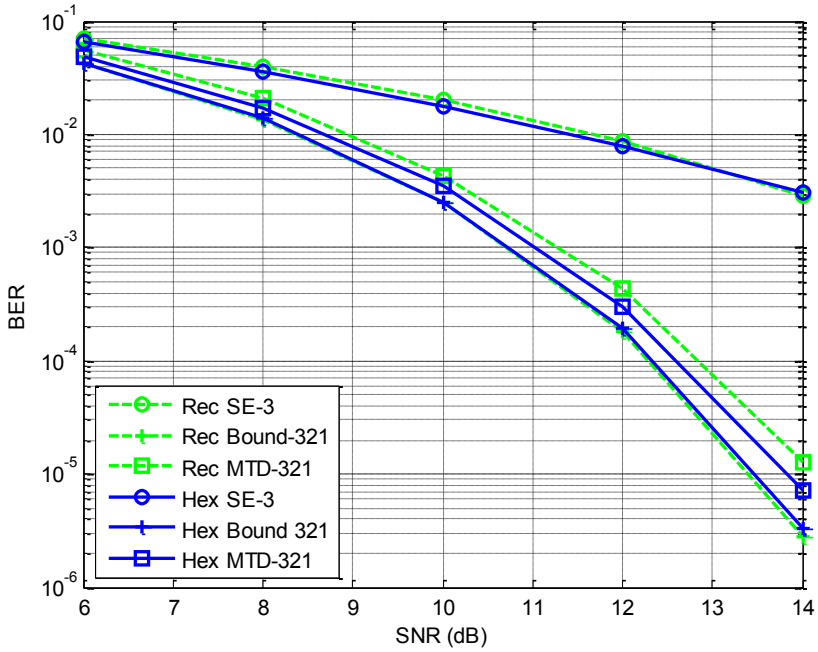

Figure 3.6 – BPMR channel detection with single-track equalization, MTD and a bound at an areal density of 2 Tb/in^2 $\left(T_x = T_y = 18\text{nm}\right)$ in the presence of ISI, ITI and AWGN for both rectangular and hexagonal island geographies.

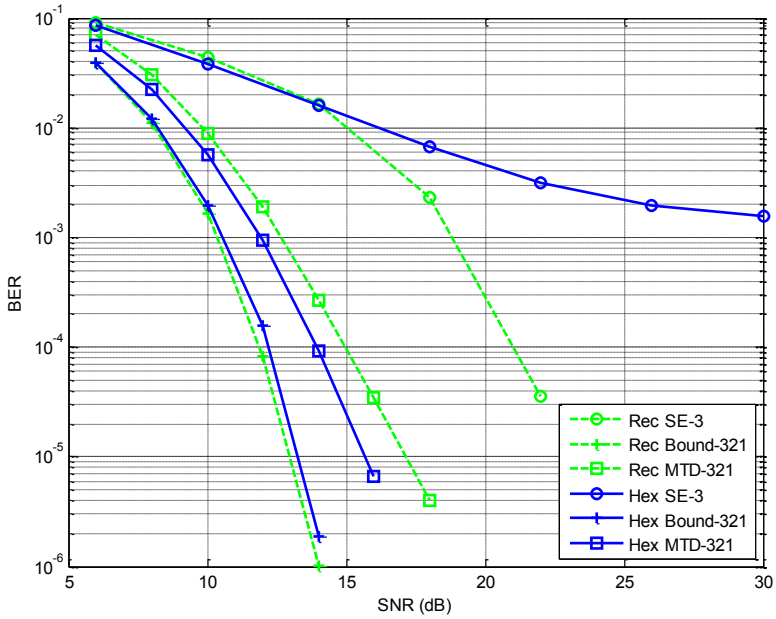

Figure 3.7 – BPMR channel detection with single-track equalization, MTD and a bound at an areal density of 2.52 Tb/in^2 $\left(T_x = T_y = 16\text{nm}\right)$ in the presence of ISI, ITI and AWGN for both rectangular and hexagonal island geographies.

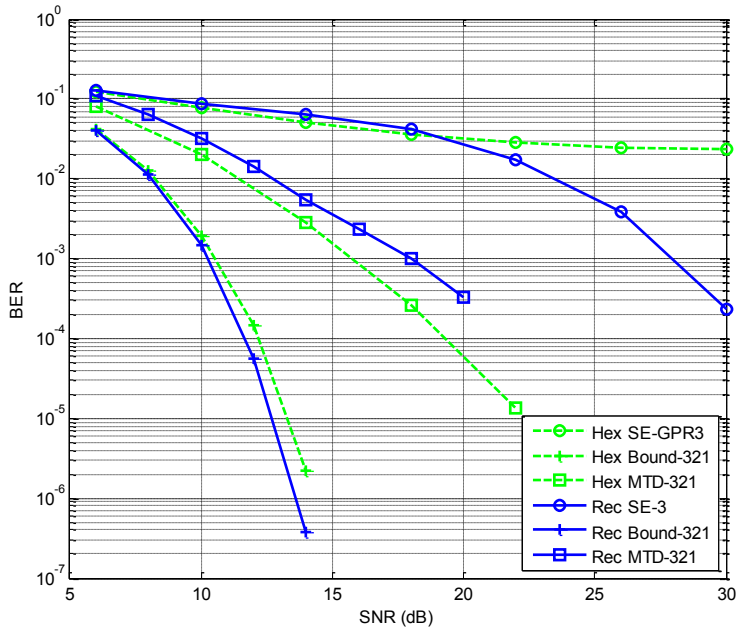

Figure 3.8 – BPMR channel detection with single-track equalization, MTD and a bound at an areal density of 3.29 Tb/in^2 $\left(T_x = T_y = 14\text{nm}\right)$ in the presence of ISI, ITI and AWGN for both rectangular and hexagonal island geographies.

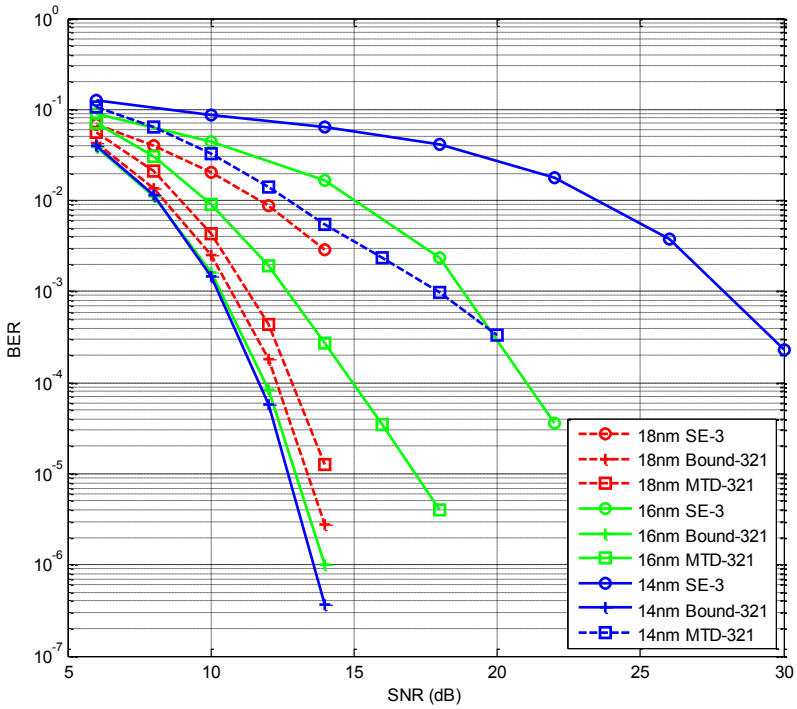

Figure 3.9 – BPMR channel detection with single-track equalization, MTD and a bound at areal densities from 2 Tb/in^2 to 3.29 Tb/in^2 in the presence of ISI, ITI and AWGN for a rectangular island geography.

As the areal density increases so does the dispersion between the curves for a given island configuration which is shown explicitly in Figure 3.9 for islands in the rectangular pattern. With MTD the bound is nearly achieved as long as the island size is kept at half the bit length. Beyond that the increased ITI and ISI tend to deteriorate the technique's performance gains. Since the hexagonal pattern under MTD performs better than a rectangular pattern in the presence of high interference conditions and it displays a nice bound, it makes sense to proceed assuming this configuration. When the areal density is scaled to reach the desired 4 Tb/in^2 level with the parameters in Table 2.1, MTD performs exceptionally well on the uncoded channel while single-track equalization still lags by almost 9 dB (see Figure 3.10). The performances shown in the previous graphs would increase slightly if longer targets were used, so the target for MTD and the bound on the first virtual track was increased to length three. This should only result in marginal increases to the overall complexity. In Figure 3.11 the performance of the channel is degraded by the addition of 10% media noise. With an island position jitter in conjunction with island size variation of 10%, the detection suffers considerable performance loss. For MTD on the staggered islands the loss is more than 4 dB at around a BER of 10^{-4}. The bound is not greatly affected as the media noise on the side tracks does not impact the detection. The single-track equalization is also heavily penalized by the presence of the media noise. But this can be somewhat accounted for by incorporating noise-prediction into the detection (see Figure 3.12).

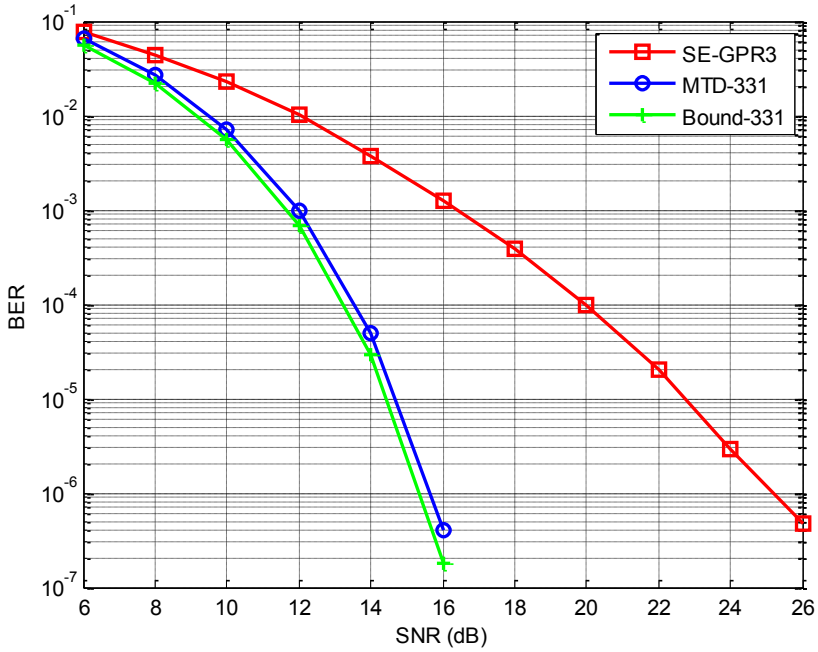

Figure 3.10 – Uncoded BPMR channel with areal density 4 Tb/in^2 with ITI, ISI and AWGN.

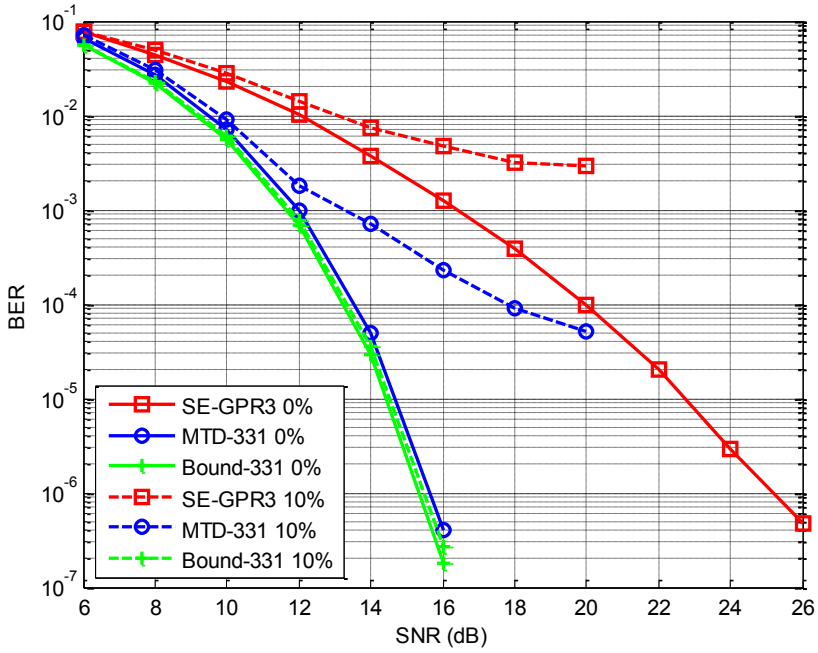

Figure 3.11 – Uncoded BPMR channel with areal density 4 Tb/in^2 with ITI, ISI, 10% media noise and AWGN.

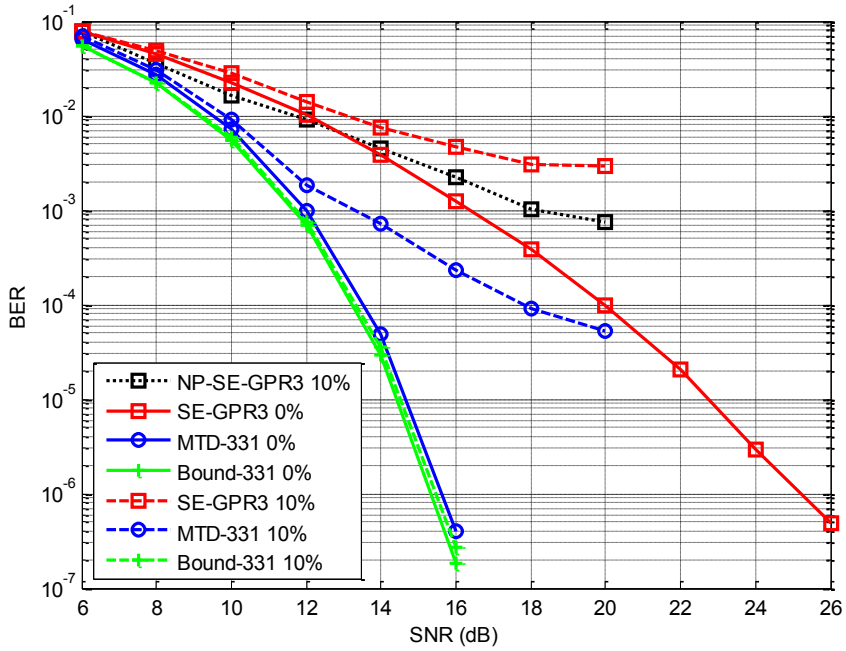

Figure 3.12 – Uncoded BPMR channel with areal density 4 Tb/in^2 with ITI, ISI, 10% media noise and AWGN.

While noise prediction results in an impressive performance improvement in single-track equalization, it cannot compete with MTD. Its contribution might improve further if it is later determined that the true BPMR media noises possess some form of a pattern.

Chapter 4 - Possible Coding Enhancements for BPMR

While MTD techniques have shown promise in dealing with the ITI, ISI, media noise and AWGN present in the BPMR channel, detection alone cannot ensure that error-free recovery of the stored bits will be reliable at the bit error rates necessary for high-density HDDs. In addition to these problems, BPMR suffers from the inclusion of written-in errors which occur outside of the scope of the detection. Such errors are the result of island position jitter, variations in the switching field of an individual island, the disturbing fields of neighboring islands, and variations in the write clock phase. Most often written-in errors occur only as single-bit errors mixed with an occasional double-bit error where both types are only substitution errors. But when the write clock phase is allowed to drift without timing recovery, insertion and deletion errors can occur that result in a shift of all subsequent bits within the sector. Given that such error events can occur multiple times within a long sector of length 32 Kbits before resynchronization of the write clock takes place, large portions of the recovered sector may be in error. Thus a powerful ECC is needed to assist in recovering data from a high-density BPMR channel.

There has been some recent work dealing with written-in errors for BPMR, but none have managed a workable solution that takes into account all of the inherent problems. In [99] a very powerful non-binary LDPC code was used to successfully recover bits in the presence of single-bit written-in errors, but such a code is not yet practical for hardware implementation and it is not clear how such a non-binary code would handle insertions and deletions. In [68] Reed-Solomon (RS)

codes were employed within a picket-shift code design specifically targeted to recover data subject to multiple insertions and deletions with good results if the code rate and write clock phase drift were both decreased to low levels. The resulting tradeoff on the code rate was unfortunately lower than what is normally required for a HDD. The Levenshtein code has been shown to be effective at correcting single insertion or deletion events within a simplified channel [100], [101] and this work has been extended by others to include both double and multiple error events in the same channel block [102], [103]. Unfortunately, these developments did not include the full list of impairments specific to BPMR. Also as the number of multiple insertions and deletions grows, the constraints imposed by the Levenshtein-based codes necessitate further drops in the code rate which is already too low for HDDs. This makes them operationally impractical for BPMR. Marker codes have also been explored for insertion and deletion environments, where markers within the code allow for the correction of errors caused by a loss of synchronization, but again the code rate penalty is high [104], [105]. Thus the search for a solution to the written-in errors continues. Here, an attempt is made to apply practical ECCs and decoding techniques to the adversities of BPMR to help gain insight into possible solutions. In terms of practical implementation and performance, cyclic LDPC codes based on finite geometries are known to perform well on a variety of channels. So it makes sense to start with these codes to determine if they can provide enough error-correction capability to mitigate the multiple degradations of the BPMR storage channel.

4.1 Low-Density Parity-Check Codes

In the past several years interest in low-density parity-check (LDPC) codes has become more prevalent due to their ability to perform close to capacity on various channels when iteratively decoded [72]. As a set of error-correction codes, they possess many rich properties that enable them to be practically implemented across a variety of communication channels. Since LDPC codes are linear block codes, they can be represented by either a generating matrix, $\mathbf{G}$, or a corresponding parity-check matrix, $\mathbf{H}$. Either representation is sufficient and basically equivalent. To differentiate LDPC codes from codes with denser parity-check matrices, the null space of an $m \times n$ parity check matrix, $\mathbf{H}$, designates an LDPC code of length n if $\mathbf{H}$ is sparse having w_c ones per column and w_r ones per row where both w_c and w_r are small compared to the size of the code *n*. This definition is somewhat ambiguous, but understandably so, as there is no true density that separates a low versus a medium density code. An LDPC code is *regular* if all rows have equivalent weight ρ and all columns have equivalent weight γ. Otherwise the LDPC code is considered *irregular*. Decoding of LDPC codes typically makes use of an iterative algorithm that accepts soft information from the channel in the form of LLRs on each bit. It then performs a number of operations with the LLRs to estimate the symbol for each bit. The algorithm chosen is normally a form of belief propagation (BP) known as the sum-product algorithm (SPA). This algorithm was originally proposed by Gallager as the soft-decoding method for his LDPC codes [106]. The algorithm is fairly robust and possesses many variations, most of which trade off performance for a less

complex process. BP algorithms typically perform on a parity-check matrix's bipartite Tanner graph representation [107] (see Figure 4.1). When the Tanner graph contains few short cycles, a BP algorithm can usually achieve near optimum performance. The smallest possible size for a cycle has length 4 (also shown in Figure 4.1) and these cycles can substantially degrade the decoding performance of a BP algorithm [108].

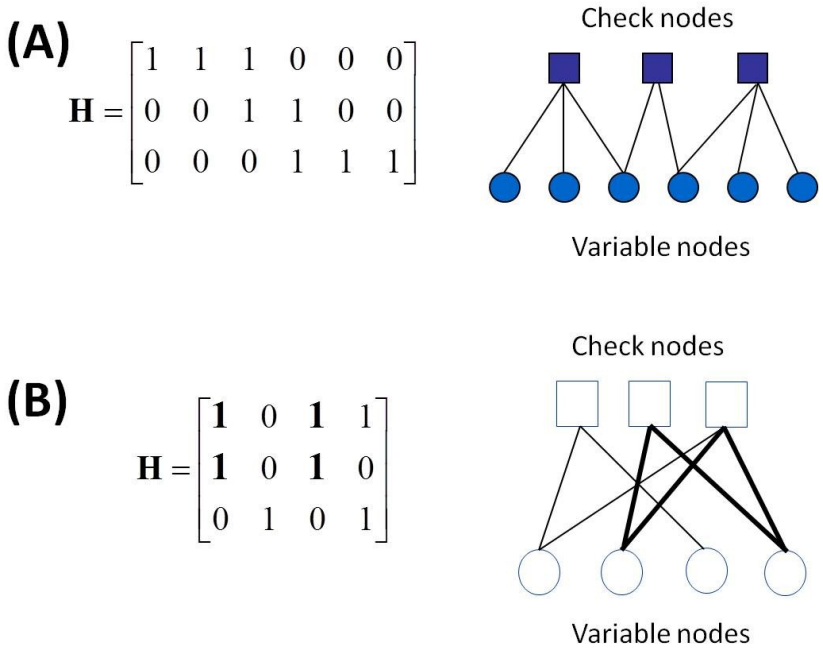

Figure 4.1 – Illustration of a parity-check matrix and its corresponding Tanner graph representation in (A) where no cycles exist and (B) where one cycle of length 4 is shown with both bold entries in the parity-check matrix and bold lines in the Tanner graph.

Thus when choosing a BP decoder, it is best to pair it with a version of the parity-check matrix which contains few if any short cycles. When the decoder succeeds at converging to a valid codeword, $\mathbf{c} = [c_1, \cdots, c_n]^T$ of length n, then

$$\mathbf{Hc} = 0\,. \tag{4.1}$$

Since BP algorithms perform iterative decoding, the decoder will either converge to a valid codeword or reach a preset number of maximum iterations where it declares a decoding failure. In the worst case scenario, the decoder can converge to the wrong codeword which results in an undetectable decoding failure.

4.2 Cyclic Linear Block Codes

While the density of the parity check matrix offers one distinction for a code, another useful designation describes an automorphic characteristic. When a linear block code C possesses the property by which any cyclic shift, right or left, of a codeword in C results in a different valid codeword in C, the linear block code C is said to be cyclic:

$$c = [c_0, c_1, \cdots, c_{n-1}] \rightarrow c' = [c_{n-1}, c_0, c_1, \cdots, c_n] \,. \tag{4.2}$$

Cyclic linear block codes can also be represented by either a generator matrix $\mathbf{G}$ or a parity-check matrix $\mathbf{H}$. In the proper form each one of these matrices will display the cyclic property within their respective rows. For example if C is the binary cyclic $[7, 4]$ Hamming code with generator polynomial $1 + x + x^3$, then it has the generator matrix,

$$\mathbf{G}_{\text{Hamming}(7,4)} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}, \tag{4.3}$$

where each subsequent row is simply one shift to the right of the preceding row. Cyclic codes are often desirable because they can be easily encoded using shift

registers with feedback connections. They can also be used to generate a larger class of quasi-cyclic codes which are themselves often desirable codes because they can be easily implemented in both the encoder and decoder [109].

4.3 Finite Geometry Codes as Cyclic LDPC Codes

Among the many different types of LDPC codes that have been constructed or designed, there is yet only a single class of cyclic LDPC codes. These codes are constructed from finite geometries which include both Euclidean and projective geometries. In such geometries the presence of only a finite number of points allows for some interesting mathematics. When considering finite geometries it is helpful to recognize the following base axioms shared with typical geometry:

1. Any two points can be connected by a line.
2. Any two lines must either intersect at a single point or not intersect at all.
3. Any two lines intersecting on more than one point must be the same line.

These simple axioms form the basics of finite geometries whose representation can be accomplished using the algebra of finite fields, which is often used to formulate linear block codes. Even though finite geometry codes were formulated many years ago, it was not until recently that Kou, Lin, and Fossorier [110] recast them as cyclic LDPC codes that could be iteratively decoded using belief propagation with near capacity performance. The associated cyclic parity-check matrices of these finite geometry codes are constructed from the incidence vectors of their many lines. They can be subjected to iterative decoding as their corresponding Tanner graphs are free of cycles of length four and thus have a minimum girth of length at least six. It has

been shown that these codes have large minimum distance and do not possess trapping sets of weight smaller than their minimum distance. Because of these properties finite geometry codes tend to perform exceptionally well on many different channels. In fact they are often designated as baseline codes in terms of BER performance when designing comparable LDPC codes.

4.3.1 Euclidean Geometry as a Realization of a Code

A number of parameters and notations need to be established in order to develop the structures within the finite geometry. As with typical geometry, finite geometry can be multidimensional. For the current context, only dimensions two and higher are useful. If *m* is used to designate the dimension of the geometry, its value must be two or greater. Let GF(*q*) be the standard notation designating a finite field of *q* elements with *q* defined as a power of a prime. When using finite fields to design linear block codes it is helpful to write *q* as

$$q = p^s; s \geq 1\,. \tag{4.4}$$

With this notation a finite Euclidean geometry of *m* dimensions over the finite field GF(*q*) can be designated EG(*m*, *q*) or as EG(*m*, p^s). This geometry will constitute points, lines and flats. To establish these structures, each point will be represented as an *m*-tuple over GF(*q*). From *q* elements the vector space of all *m*-tuples over GF(*q*) defines the total number of points, *n*, of the geometry. Thus,

$$n = q^m\,, \tag{4.5}$$

and a point can be viewed as either a subspace or a coset of a subspace of the vector space of m-tuples of subspace dimension zero. With a set of points defined in the finite geometry, lines can be constructed to connect between various points. These lines can be viewed as one-dimensional structures that form a subspace or a coset of a subspace of the vector space of m-tuples with subspace dimension 1. A finite geometry with q^m points that does not violate the three stated axioms is characterized by

$$F(m,1) = \frac{q^{m-1}(q^m - 1)}{q - 1} \tag{4.6}$$

unique lines with q points connected by each line. The set of all lines for EG(m, q) can be partitioned into groups of either parallel or intersecting lines.

By further generalizing the definitions of points and lines, higher dimensional structures can be formed within the geometry which are referred to as flats. Given the vector space over GF(q) can be partitioned into either various dimensional subspaces or into cosets of various dimensional subspaces, the dimensionality m of the geometry limits the extent of the partitioning and thus the dimensions for flats. Flats are often denoted as μ-flats where for EG(m, q) μ-flats can range from $0 \leq \mu \leq m$. Thus the number of flats possible for EG(m, q) is

$$F(m,\mu) = q^{m-\mu} \prod_{i=1}^{\mu} \frac{q^{m-i+1} - 1}{q^{\mu-i+1} - 1}. \tag{4.7}$$

For EG(2,q) the finite geometry is limited to points (0-flats), lines (1-flats) and 2-flats. In order to formulate these structures into a genuine linear block code, it is best to

make use of the properties of finite fields by introducing an extension field of GF(q). When doing so, an extension field GF(q^m) of GF(q) represents a vector space of dimension m of all possible m-tuples over GF(q) where any given element in the extension field is an m-tuple over GF(q). Thus the relationships of the extension field and the finite geometry to GF(q) are equivalent and GF(q^m) becomes a realization of EG(m, q). Given a primitive element α of GF(q^m), its various powers can be mapped to the various n points of the finite geometry:

$$\begin{matrix} \alpha^{-\infty} = 0 \\ \alpha^{0} = 1 \\ \vdots \\ \alpha^{q^m-2} = n-2 \end{matrix} , \tag{4.8}$$

where the point mapped to alpha at negative infinity is the origin of the geometry. It is helpful to also define a set of q points passing through the origin of EG(m, q) by

$$\beta\alpha^{n_i} : \beta \in \mathrm{GF}(q). \tag{4.9}$$

This arrangement allows for the following distinction

$$\alpha^{n_j} + \beta\alpha^{n_i} : i \neq j; \beta \in \mathrm{GF}(q). \tag{4.10}$$

Since zero is one of the values for beta, the statement above describes a set of q points which form a line passing through both points represented by α^n. To make the jump from finite geometry to cyclic linear block code, it is necessary to remove the origin point. This effectively limits the set to (q^m – 1) points and it also removes all lines passing through the origin point leaving

$$F(m,1) = \frac{(q^{m-1}-1)(q^m-1)}{q-1}, \tag{4.11}$$

lines within the geometry - now more appropriately referred to as the subgeometry of EG(m, q). Any line in the subgeometry can be represented by q distinct powers of the primitive element α,

$$F_1(m,1) = \{\alpha^{p_1}, \alpha^{p_2}, \cdots, \alpha^{p_q}\}; p \in \mathrm{GF}(q), \tag{4.12}$$

as each line still passes through q points. If this line, or any given line in the subgeometry for that matter, is multiplied by any allowable representation of a non-origin point,

$$\alpha^i; 0 \leq i < q^m - 1\,. \tag{4.13}$$

The result is simply a shifted version of the original line taken modulo $q^m - 1$. For example,

$$\begin{aligned} \alpha^6 F_1(m,1) &= \{\alpha^{p_1+6}, \alpha^{p_2+6}, \cdots, \alpha^{p_q+6}\} \\ &= F_6(m,1); p \in \mathrm{GF}(q) \end{aligned} \tag{4.14}$$

represents a different line in the finite geometry and reveals the inherent cyclic structure. All of the lines in the subgeometry can be partitioned into

$$\frac{(q^{m-1}-1)}{q-1} \tag{4.15}$$

cyclic classes. A binary vector $\mathbf{v}_L$ of length $q^m - 1$ can be constructed from any given line of the subgeometry by taking the powers of alpha as the column indices where ones should be located and filling the remaining vector positions with zeros,

$$\mathbf{v}_L^{\mathrm{EG}} = \left(v_0, v_1, \cdots, v_{q^m-2}\right), \tag{4.16}$$

where the point designated α^i is either on line L if $v_i = 1$ or not on line L if $v_i = 0$. This binary vector can then be used as the first row of a parity check matrix of a linear block code. Its subsequent $q^m - 2$ right circular shifts result in the remaining rows of the parity check matrix.

From the preceding development consider a finite two-dimensional Euclidean plane denoted as EG(2, q) with parameter $q \equiv 2^S$ with s some positive integer. For any choice of s the given plane will contain q^2 points and $q^2 + q$ lines. Each point will lie on $q + 1$ lines and each line will contain q points. For the resulting code to be cyclic, the point at the origin (the zero point) is removed along with the $q + 1$ lines passing through it. This leaves $q^2 - 1$ points and $q^2 - 1$ lines in the plane. The set of points and lines are used to construct a corresponding parity-check matrix **H** by letting the set of points coincide with the columns and the set of lines correspond to the rows. By appropriately ordering the lines and columns, the resulting parity-check matrix is cyclic and a Type-1 EG(2, q) code emerges with the characteristics shown in Table 4.1. By taking subsequent values of s, Table 4.2 can be constructed to show the actual values for the parameters. The code rate for a given code is denoted R and is equivalent to the code dimension divided by the code length. Because these codes form the only class of binary cyclic LDPC codes, it is necessary to examine the structure behind this class. Furthermore, only binary codes corresponding to two-dimensional geometries are considered here.

Table 4.1 – Various descriptive parameters for Type-1 EG(2, q) cyclic LDPC codes.

Parameter	
size of **H**	$n \times n$
code length	$n = q^2 - 1$
code dimension	$k = n - 3^s + 1$
minimum distance	$d_{min} = q + 1$
row weight of **H**	$\rho = q$
column weight of **H**	$\gamma = q$

Table 4.2 – Examples of Type-1 EG(2, q) cyclic LDPC codes.

s	(n, k)	d_{min}	ρ ; γ	R
2	(15, 7)	5	4	0.4667
3	(63, 37)	9	8	0.5873
4	(255, 175)	17	16	0.6863
5	(1023, 781)	33	32	0.7634
6	(4095, 3367)	65	64	0.8222
7	(16383, 14197)	129	128	0.8666
8	(65535, 58975)	257	256	0.8999

Higher finite dimensional geometries $(m > 2)$ exist and can be used to generate non-binary and quasi-cyclic codes. In addition to Type-1 Euclidean geometry (EG) codes, Type-1 projective geometry (PG) codes will also be included as they share very similar structure and properties with the Type-1 EG codes.

4.3.2 Projective Geometry as a Realization of a Code

A Euclidean geometry is not the only finite geometry capable of formulating a linear block code. Projective geometries have also been shown to be an effective tool for

constructing codes. Other finite geometries exist and may yet have a useful role to play in the future of error control codes [111]–[113], but that particular problem is not covered here. Projective geometries stem from Euclidean geometries and thus they share a number of commonalities. A projective geometry contains points, lines and flats that are usually formed by projecting higher dimensional constructs from a Euclidean geometry onto a lower dimensional space. Thus a two-dimensional projective geometry can be constructed by projecting two-dimensional lines from a Euclidean geometry into one-dimensional points for the projective geometry. This also works for lines and flats in the projective geometry. Remember that a line in Euclidean geometry is also a 1-flat and that a 2-flat is just a construct one dimension greater than a 1-flat, or line. So 2-flats in a Euclidean geometry can be projected into 1-flats for the projective geometry which effectively establishes its lines. To generalize, any m-flat with $m > 0$ in a Euclidean geometry can be projected as an (m – 1)-flat for a projective geometry. One interesting point worth noting, a projective geometry does not possess an origin simply because individual points from a Euclidean geometry cannot be projected as they have zero dimension. Thus, all lines passing through the origin of a Euclidean geometry project to non-origin points within a projective geometry.

Once the points of a projective geometry are established, they can be used to construct lines that can then be used to construct 2-flats which can then be used to construct flats of higher dimension and so forth. If the projective geometry is held to the axioms for finite geometries, then the projective geometry behaves in many

respects like its Euclidean geometry counterpart. Its points can also be represented using an extension field covering a base finite field. Following from Euclidean finite geometries, for dimension m and finite field GF(q) with q elements, a projective geometry with m dimensions can be constructed. This geometry will be designated PG(m, q). Each point in PG(m, q) can be represented as an (m + 1)-tuple over the finite field GF(q) where any given primitive element of GF(q) along with all of its multiples represent the same point. Thus the GF(q^{m+1}) extension field over GF(q) can be used to designate the points within PG(m, q). Based on the number of lines in a EG(m+1, q), which would project as points into PG(m, q), the number of points for PG(m, q) will be

$$n = \frac{q^{m+1} - 1}{q - 1}. \tag{4.17}$$

A primitive element of the extension field, α, represents the points in PG(m, q). With two distinct points from PG(m, q), designate them α^x and α^y, a line can be drawn connecting them and it has the form

$$L_{PG} = \mathrm{S}_i \alpha^x + \mathrm{S}_j \alpha^y \,, \tag{4.18}$$

with S_i and S_j scalars from the base field GF(q). There exist (q^2 – 1) possible choices for the scalars and of these (q – 1) of them are simply multiples of the given points. For a line of this form there will be

$$\frac{q^2 - 1}{q - 1} = q + 1 \tag{4.19}$$

points per line. The number of lines intersecting at a point in PG(m, q) can be found by determining the number of 2-flats intersecting at a 1-flat in EG(m + 1, q). For any EG(m, q) the number of intersecting 2-flats on a line can be found from

$$I_{\mathrm{EG}}(m, \mu, \mu - 1) = \frac{q^{m-\mu+1} - 1}{q - 1}, \tag{4.20}$$

which for lines intersecting points in PG(m, q) becomes

$$I_{\mathrm{EG}}(m, \mu, \mu - 1) = \frac{q^{m-\mu+1} - 1}{q - 1} \Longrightarrow I_{\mathrm{EG}}(m + 1,2,1)\,, \tag{4.21}$$

$$I_{\mathrm{EG}}(m + 1,2,1) = I_{\mathrm{PG}}(m, 1,0)\,, \tag{4.22}$$

$$I_{\mathrm{PG}}(m, 1,0) = \frac{q^{m+1-2+1} - 1}{q - 1} = \frac{q^m - 1}{q - 1}. \tag{4.23}$$

Now that the number of points, the definition of a line, the number of points per line, and the number of lines intersecting at a point are known for PG(m, q), the total number of distinct lines can be found as follows:

$$\begin{aligned} F_{\mathrm{PG}}(m, 1) &= \frac{(q^{m+1} - 1)}{q - 1} \times \frac{1}{q + 1} \times \frac{(q^m - 1)}{q - 1} \\ &= \frac{(q^{m+1} - 1)}{(q^2 - 1)} \frac{(q^m - 1)}{(q - 1)}. \end{aligned} \tag{4.24}$$

Unlike its Euclidean counterpart, to develop a cyclic linear block code from PG(m, q) there is no need to form a subgeometry by removing the origin point because PG(m, q) has no origin. Any point in PG(m, q) can be represented by one of the n distinct powers of the primitive element α of GF(q^{m+1}). The points incident on a given line can be designated by the powers of α where $\left(\alpha \to \alpha^j\right)$. These powers then

constitute the column indices where ones should be placed within a vector of length *n*. This vector contains only zeros in all other positions,

$$\mathbf{v}_L^{\mathrm{PG}} = (v_0, v_1, \cdots, v_{n-1}) \,, \tag{4.25}$$

where the point designated α^j is either on line L if $v_i = 1$ or not on line L if $v_i = 0$. The vector can then be used as a row of a parity check matrix. Any right or left cyclic shift of this vector results in another line within PG(*m*, *q*). Thus a parity check matrix of size (*n* x *n*) can be constructed which represents a cyclic linear block code.

Based on the previous development, the corresponding PG codes of interest can be established. First consider a finite projective plane designated PG(2, *q*). This plane will contain $q^2 + q + 1$ points and $q^2 + q + 1$ lines. Each line will contain $q + 1$ points and each point lies on $q + 1$ lines. A parity-check matrix **H** can be formulated by using the set of points as the columns and the set of lines as the rows. Again with appropriate ordering the parity-check matrix will be cyclic. This results in a Type-1 PG(2, *q*) LDPC code with parameters listed in Table 4.3.

Table 4.3 – Various descriptive parameters for Type-1 PG(2, *q*) cyclic LDPC codes.

Parameter	
size of **H**	$n \times n$
code length	$n = q^2 + q + 1$
code dimension	$k = n - 3^s - 1$
minimum distance	$d_{min} = q + 2$
row weight of **H**	$\rho = q + 1$
column weight of **H**	$\gamma = q + 1$

For subsequent values of *s*, Table 4.4 can be enumerated to show the parameters for Type-1 PG codes. The code rate for a given code is again denoted R.

Table 4.4 – Examples of Type-I PG(2, *q*) cyclic LDPC codes.

s	(n, k)	d_{min}	ρ ; γ	R
1	(7, 3)	4	3	0.4286
2	(21, 11)	6	5	0.5238
3	(73, 45)	10	9	0.6164
4	(273, 191)	18	17	0.6996
5	(1057, 813)	34	33	0.7692
6	(4161, 3431)	66	65	0.8246
7	(16513, 14326)	130	129	0.8676
8	(65793, 59231)	258	257	0.9003

Note that these PG codes in combination with their corresponding EG codes are the best iteratively decodable cyclic LDPC codes currently known. There are in fact other cyclic EG codes, but these codes contain a larger number of ones in their parity-check matrices making them less supportive of iterative decoding. The cyclic parity-check matrices for EG and PG codes are all square ($n \times n$) and can be formulated as a Tanner graph with n variable nodes and n check nodes. A square **H** matrix is not typically a characteristic of LDPC codes, but it is characteristic of cyclic codes. All of the cyclic **H** matrices from these codes contain no cycles of length four. The following shows an example of a parity-check matrix (not in systematic form) from each geometry.

$$H_{PG(7,3)} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.26}$$

$$H_{EG(15,7)} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.27)$$

It is worth noting that the smallest PG(2, q) code above is equivalent to the generating matrix of the [7, 4] Hamming code with generator polynomial $1 + x + x^3$ shown in (4.3). BP handles matrices such as the ones above easily and with good decoding results [110]. But before using a finite geometry LDPC code on a BPMR channel, it is necessary to create another type of LDPC code of the same length and code rate to assist in establishing the performance of a finite geometry LDPC code.

4.3.3 Progressive Edge Growth LDPC Codes

For the finite geometry codes previously discussed, there is basically a single procedure for designing the code. While finite geometry codes tend to perform well, LDPC codes based on computer-based searches have often been associated with better performance on a channel [114]. Unfortunately, these types of LDPC codes do not possess inherent structures that allow for simple encoding, although there have

been some proposed methods that offer improved encoding [115]. But their use is still currently limited. Here an LDPC design method known as progressive edge growth (PEG) will be used to generate a code of the same length n and code rate R which can be used in comparison with finite geometry codes on the BPMR channel.

The PEG algorithm functions by attempting to construct a Tanner graph with a large girth which typically provides good iterative decoding performance by avoiding small cycles [116]. It is initiated with a set of n variable nodes N and a set of m check nodes M. The sets of nodes N and M are not connected so that the graph starts out with no edges. Edges are then progressively added to the graph based on the algorithm's edge-selection procedure. Since the procedure progresses locally, the PEG algorithm keeps track of a neighborhood within depth l of a given variable node n_i. This neighborhood set is denoted N_i^l and it has a complementary set defined as $\bar{N}_i^l = M \backslash N_i^l$. As edges are added, they are selected in such a way as to maximize the local girth of the connecting node. Selection of an edge to a variable node n_i is performed by constructing a path tree R_i with n_i as the root node and then growing the tree to a depth of l where one of two conditions occurs:

(1) The cardinality of N_i^l stops increasing and is less than the number of check nodes, m, or

(2) $\bar{N}_i^l \neq \emptyset$ but $\bar{N}_i^{l+1} = \emptyset$.

Under condition (1) the check node m_i in N_i^l with the smallest degree that connects n_i and m_i is selected as the edge. Under condition (2) an edge is added that connects n_i to the check node at the $(l+1)$-level that has the largest distance from

n_i. The procedure then progresses to the next variable node n_i until the entire Tanner graph is constructed with maximized local girth.

4.3.4 Simulations on the BPMR Channel

Following the results of the previous chapter, simulations are arranged to take advantage of the hexagonal island configuration. The density of interest is 4 Tb/in^2 and coincides to square islands with 6.35 nm side lengths and bit period and rack pitch equal to 12.7 nm. Sector size is set at 512 bytes. As before, single-track equalization, MTD and a bound are selected for comparison. In single-track equalization the 1D target has length three and is denoted SE-GPR3. For MTD the span of the ITI coverage consists of five adjacent tracks with target track a_0 and two virtual side tracks $a_{\pm 1}$ and $a_{\pm 2}$. The GPR targets have lengths of three on the center and first virtual track and length one on the second virtual track which is denoted MTD-331. For MTD the center track is always equalized using joint-track equalization while the side tracks use single-track equalization. The bound is estimated using MTD with equivalent target lengths by assuming all side-track information is error free. Initially only ITI, ISI and AWGN are present. Then media noise and written-in errors are imposed on the channel. In Figure 4.2 the uncoded channel is shown along with a constructed PEG code whose characteristics have been matched to those of the EG(4095, 3367) code displayed in Figure 4.3. Both LPDC codes have length 4095 and a code rate equal to 0.8222. The EG code is cyclic, has a large minimum distance ($d_{min} = 65$), and a large column weight of 64.

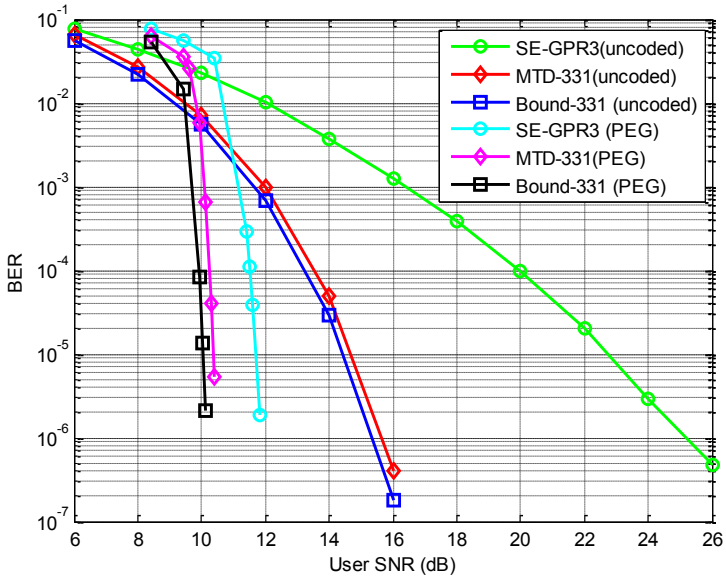

Figure 4.2 – BPMR channel with areal density 4 Tb/in^2 with ITI, ISI and AWGN. BER curves represent both the uncoded channel and the performance of a PEG code of length 4095 with code rate 0.8222.

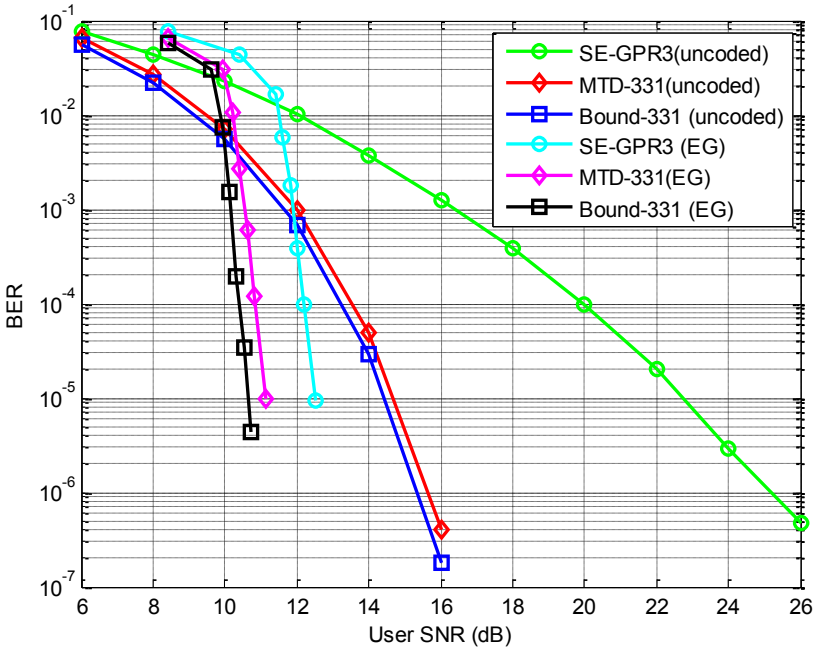

Figure 4.3 – BPMR channel with areal density 4 Tb/in^2 with ITI, ISI and AWGN. BER curves represent both the uncoded channel and the performance of a EG(4095, 3367) with code rate 0.8222.

The PEG code is not cyclic, has a much smaller column weight of four and has a less dense parity-check matrix than the EG code. Both codes result in large coding gains for both single-track equalization and MTD. The gain in performance for single-track equalization is so large that it can compete with the MTD.

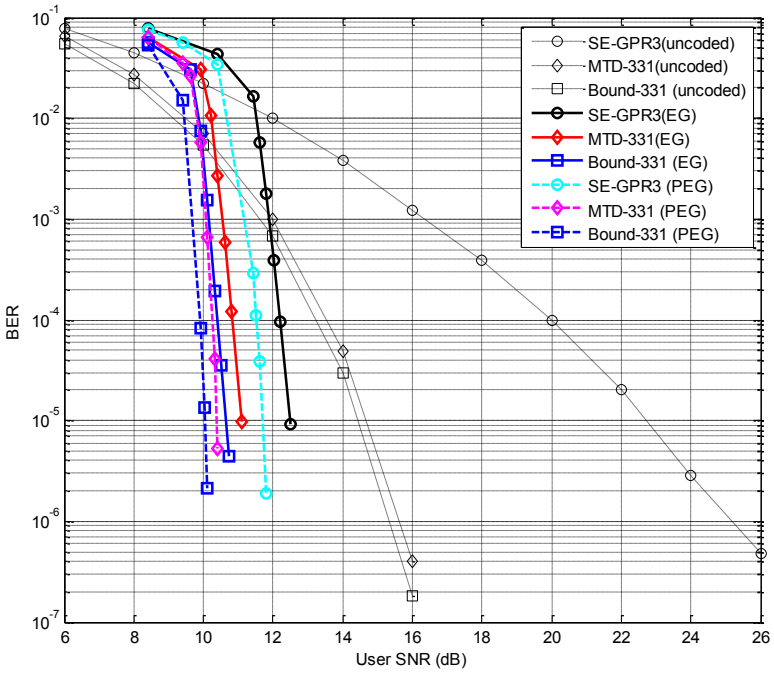

Figure 4.4 – BPMR channel with areal density 4 Tb/in^2 with ITI, ISI and AWGN. BER curves represent the uncoded channel and the performance of both the PEG code and EG(4095, 3367) codes both with code rate 0.8222.

In Figure 4.4 a close comparison between the two codes can be made. The PEG code outperforms the EG code by more than 0.5 dB for MTD. The PEG code's MTD and single-track equalization curves are also slightly closer to their bound than the corresponding curves for the EG code. While the PEG code's performance is worth noting, it is only a measure of comparison for the EG code whose cyclic structure coupled with its impressive performance makes it attractive. Since it can handle the

ITI, ISI and AWGN in the channel for both single-track equalization and MTD, it is left to determine whether or not it can also handle the more problematic case caused by the presence of both media noise and written-in errors.

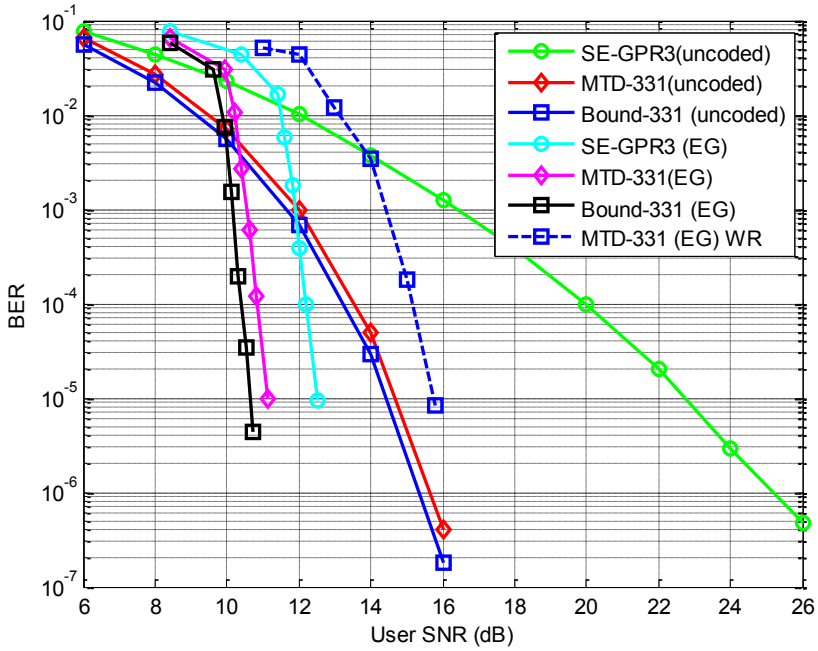

Figure 4.5 – BPMR channel with areal density 4 Tb/in^2 with ITI, ISI, 10% media noise, AWGN and written-in errors. Written-in errors are substitution type errors only. BER curves represent both the uncoded channel and the performance of a EG(4095, 3367) with code rate 0.8222.

Based on the results in Figure 4.5, it is apparent that a finite geometry code is capable of effectively recovering data from the BPMR channel even when many of the degradations of the channel are introduced. The coding loss is substantial for MTD and amounts to almost 5 dB at a BER of 10^{-5}, but this is somewhat expected given the presence of 10% media noise and substitution error rate of 1.5×10^{-3}. Unfortunately, the EG(4095, 3367) code is not capable of solving the problems caused by insertions and deletions. On a long sector of length 32 Kbits, whenever an insertion or deletion is inserted anywhere except near the end of the sector, the

decoder fails to converge to a codeword. Thus it is necessary to look for another option of dealing with the insertions and deletions. One option would be to utilize the finite geometry code in a more judicious way that offers significant decoding gains. Since finite geometry codes are cyclic, there may be a way to utilize some of their automorphic properties to augment their decoding performance.

4.4 Permutation Decoding

The ability to morph from one codeword to another within a code *C* can be utilized in a decoding environment. One of the first such applications was proposed by F.J. MacWilliams in a 1964 paper that initiated a method called permutation decoding [117]. While MacWilliams mainly made use of the two known automorphisms of cyclic codes, her methods were easily extendable to include any known automorphisms. One of the best descriptions of the permutation algorithm is provided by Huffman and Pless [118] in which they prove a couple of necessary theorems for the algorithm. The overall mechanics of the algorithm are explained below.

For a codeword in systematic form with corresponding parity-check matrix $\boldsymbol{H} = [A \mid I]$ and generating matrix $\boldsymbol{G} = [I \mid A^T]$, the transmitted information bits are in order within the codeword followed by the parity bits. The premise of permutation decoding is to move the errors in the received codeword into the parity positions so that all the information bits are correct. Then the information bits can be retrieved, error free, and the permuted codeword can be un-permuted to

recover the information bits actually sent over the channel. This can be accomplished with a known set of automorphisms given that the number of bits in error in the received codeword does not exceed the error correction capability of the code. Thus multiple automorphisms are required to move not more than t error positions within the n bit positions $\{0, 1, \ldots, n-1\}$ into the parity bit positions indexed by $\{k, k+1, \ldots, n-1\}$. The process of permutation decoding can be stated succinctly as the following set of steps.

1. Find a set of valid automorphisms that form a PD-set of C with cardinality $|\sigma_i| = x$
2. For the received codeword y, calculate the weight of the syndromes: $\mathbf{H}(y\sigma_i)^T$ for $i = 1, 2, \ldots, x$ until such weight is t or less or until x is exceeded.
3. If no weight is t or less declare a decoding failure and stop.
4. For the codeword $(y\sigma_i)$ with $i \leq x$, extract the information bits.
5. Apply the parity check equations to the information bits to obtain the correct parity bits for the morphed codeword.
6. Use the correct information and parity bits to construct a valid codeword for $(y\sigma_i)$.
7. Apply the inverse automorphism to $(y\sigma_i)$ to obtain codeword y.

Since automorphisms are in practice difficult to find, the need for a minimum set of automorphisms to effect the changes in the received codeword serves dual purposes. A minimum set is less complex to implement and hopefully easier to determine. The following, taken from and proven in [118], is a lower bound on the size of a minimum PD-set.

> Bound: A PD-set of size P for a t-error correcting $[n, k]$ code with redundancy $r = n - k$ satisfies
>
> $$P \geq \left\lceil \frac{n}{r} \left\lceil \frac{n-1}{r-1} \cdots \left\lceil \frac{n-t+1}{r-t+1} \right\rceil \cdots \right\rceil \right\rceil. \tag{4.28}$$

For the well-known extended Golay code [24, 12, 8], the bound requires a set of no less than 14 automorphisms. Given that this code has been extensively studied, at least one PD-set for it is known. However, for a given code there is no guarantee that such a set exists nor that the set AUT(*C*) contains enough automorphisms to construct such a set. There has been much work on finding PD-sets and s-PD-sets as mentioned in a survey by Key [119], but nothing yet that sheds light on decoding long codes such as LDPC codes. MacWilliams did show in her paper that the set of automorphisms for a cyclic code, σ_C, will be a PD-set for *C* when $k < \frac{n}{t}$. Thus, in a strict sense permutation decoding cannot be used with an LDPC code of any great length as the number of required automorphisms is too high. But the concepts of applying automorphisms to the parity-check matrices of a linear block code can still be put to use.

4.5 Automorphism Decoding

For the specific purpose of error-control coding, a linear block code *C* with code words represented as code vectors, an automorphism of *C* is any permutation of the column indices that maps a given codeword to another codeword in *C*. So if one starts with a codeword of *C*, then applying an automorphism simply results in a different valid codeword of *C*. Automorphisms allow individual pieces of a codeword to be manipulated in terms of position without invalidating the structure or properties of the code *C*. This flexibility is being used at the decoding end of communication systems to help recover the information transmitted across the

channel. As the knowledge of AUT(*C*) is limited for any given code, the idea of using automorphisms in the context of decoding error-control codes automatically meets with considerable challenges. Luckily, for the class of cyclic linear block codes there exist two well-known automorphisms of AUT(*C*). The first automorphism is easily recognized when referenced to cyclic codes. By the very definition, the cyclic property represents a valid automorphism which can be stated as follows:

$$\sigma C : i \rightarrow i + 1 \bmod n , \tag{4.29}$$

where the index *i* represents the position, or column index, in a codeword vector *c*. Also, any power of the cyclic automorphism represents a repeated application of a single shift and can be expressed as

$$\sigma^2 C : i \rightarrow i + 2 \bmod n , \tag{4.30}$$

$$\sigma^3 C : i \rightarrow i + 3 \bmod n , \tag{4.31}$$

$$\therefore \sigma^n C : i \rightarrow i + n \bmod n = i . \tag{4.32}$$

The second automorphism for cyclic codes can be expressed as the following:

$$\sigma_{A-F} : i \rightarrow 2i \bmod n \tag{4.33}$$

Repeated applications of this automorphism are also represented as powers of $\sigma_{A\text{-}F}$. In the early work by MacWilliams [117] she showed that automorphisms can be decomposed into products of disjoint cycles. These disjoint cycles are invariant under the given automorphism. For the cyclic automorphism σ_C the entire set of shifts forms a single disjoint set. For $\sigma_{A\text{-}F}$ the number of disjoint cycles equals the number of cosets of the *n*-roots of unity. It is important to keep these cycles in mind

when using a particular automorphism since they basically represent a sub-cycle of the overall automorphism. Thus, the cyclic automorphism does not contain internal looping that maps columns back onto themselves before the *n*th cyclic shift. In contrast the Artin-Frobenius transform does. For a cyclic code of length seven, three consecutive applications of the automorphism are all that is required to reproduce the original matrix. This is illustrated in Table 4.5.

Table 4.5 – Permutation of column indices for $\sigma_{A\text{-}F}$ with $n = 7$.

σ^1	σ^2	σ^3	
0 →	0 →	0 →	0
1 →	4 →	2 →	1
2 →	1 →	4 →	2
3 →	5 →	6 →	3
4 →	6 →	1 →	4
5 →	2 →	3 →	5
6 →	3 →	5 →	6

Table 4.6 – Permutation of column indices for $\sigma_{A\text{-}F}$ with $n = 15$.

σ^1	σ^2	σ^3	σ^4	
0 →	0 →	0 →	0 →	0
1 →	8 →	4 →	2 →	1
2 →	1 →	8 →	4 →	2
3 →	9 →	12 →	6 →	3
4 →	2 →	1 →	8 →	4
5 →	10 →	5 →	10 →	5
6 →	3 →	9 →	12 →	6
7 →	11 →	13 →	14 →	7
8 →	4 →	2 →	1 →	8
9 →	12 →	6 →	3 →	9
10 →	5 →	10 →	5 →	10
11 →	13 →	14 →	7 →	11
12 →	6 →	3 →	9 →	12
13 →	14 →	7 →	11 →	13
14 →	7 →	11 →	13 →	14

For such a short code the only column that does not experience a shift is the zero (first) column. This will always be true when applying the Artin-Frobenius automorphism. Other cycles can appear as *n* gets larger as can be seen in Table 4.6. The highlighted numbers in Table 4.6 reveal the sub-cycle (5, 10) where these two column indices switch back and forth between consecutive applications of the automorphism. A recent development makes use of the Artin-Frobenius automorphism for the purpose of decoding medium-density parity-check codes.

4.5.1 Auto-Diversity Decoding

Ouzan and Be'ery [120] proposed a method of decoding medium-density parity check (MDPC) codes that made use of automorphisms of cyclic codes. They refer to it as auto-diversity (AD) decoding and showed that it could be used to effectively decode codes that were typically denser than standard LDPC codes. The algorithm is basically an extension of the standard iterative belief propagation method which functions as follows:

- The AD decoder requires a parity-check matrix $\mathbf{H}$, a known set of automorphisms from the automorphism group of the code *Aut(C)*, the LLR input vector representing the soft information about the channel, the number of iterations for the SPA, and the number of iterations for the AD decoder.
- It applies the standard Sum Product Algorithm and attempts to decode the channel vector through a set number of iterations.
- If a valid codeword is found, the process is stopped.
- If the output of the SPA is not a valid codeword, then the AD decoder randomly selects an automorphism from the code's automorphism group *Aut(C)* and applies it to the interim vector. This has the effect of shifting the bits upon which the given error patterns are affecting.

- This interim vector is then submitted once more to the SPA and the process repeats until either a valid codeword is obtained or a set number of iterations for the AD decoder is reached.
- If the decoder does not converge to a valid codeword, the decoder estimates the codeword using a least metric selector.

To help illustrate the functioning of the AD decoder the following algorithmic description is provided and follows that of Ouzan and Be'ery.

Supports: parity-check matrix H and set of automorphisms
Inputs: LLR_{in} (soft input vector, length n)
I_{spa} (maximum number of iterations for sum product algorithm)
N_{ad} (maximum number of iterations for AD decoder)
Process:
for $1 \le j \le N_{ad}$
  for $1 \le i \le I_{spa}$
    perform SPA based on LLR_{in} and H
    return estimated codeword, c_i
  end for loop (SPA)
  if $c_i \times H^T = 0$ then
    $c = c_i$
    return codeword c and STOP
  else
    randomly choose σ from $Aut(C)$
    $H = \sigma H$ (apply the automorphism)
  end else
  end if
end for loop (AD)
if valid codeword not found
  estimate codeword c with least metric selector
  return codeword c
end if
STOP
Output: codeword c

When examining the AD decoding algorithm there are a number of points worth making. Since this algorithm makes use of only one known automorphism of a cyclic code (the Artin-Frobenius automorphism), it can be applied to any cyclic code. The

strategy involved by AD is to avoid error patterns that cause BP to not converge during the decoding process by permuting the bit positions of the parity-check matrix. The AD decoder attempts to provide a different arrangement of bits for the error pattern, which could prove decodable by a further run of the BP algorithm. When actually running this algorithm, the parity-check matrix $\mathbf{H}$ is not permuted by the automorphism σ in practice. Instead the equivalent operation of permuting the LLR vector with σ^{-1} is performed and the output of the decoder is subsequently un-permuted to avoid any rearrangement of $\mathbf{H}$ inside the BP algorithm. Gains in decoding performance via AD are at a cost of increased complexity, which will be at most N_{ad} times the complexity of running straight BP.

4.5.1.1 Remarks on AD Decoding

The true attractiveness of AD decoding lies in its simplicity. If one is already running BP decoding on a cyclic code, then advancing to an AD decoder is relatively straight forward. However the implementation of the AD algorithm leaves much to be desired. Applying automorphisms in a random fashion is a hit-or-miss strategy. One does not know if the resulting error patterns are easier to decode or not. It is also possible for the algorithm to burn one or more outer iterations by applying subsequent automorphisms that result in cycling back to a previously used version of the parity-check matrix. Albeit, the probability of this happening is small with codes of even moderate length and highly improbable for practical LDPC codes. Given such considerations one has to wonder if there is not a smarter way to apply the automorphisms used in AD decoding so as to increase the algorithms

effectiveness. It just so happens that another form of automorphism decoding called automorphism group decoding deals with these issues but it requires knowledge of a code's set of minimum codewords [121] which cannot be practically acquired for long LDPC codes. Since the EG(4095, 3367) code used earlier on the BPMR channel is cyclic and its parity-check matrix is relatively dense for an LDPC code, implementing AD decoding may provide the gains needed to hopefully deal more effectively with the written-in errors of the channel.

4.5.1.2 AD Decoding on the BPMR Channel

To test if the AD decoder will provide significant gains when used over a BPMR channel, simulations are arranged using a hexagonal island configuration with an areal density of 4 Tb/in^2. The sector size is set at 512 bytes. Single-track equalization, MTD and a bound are all used for comparison where the GPR target lengths are set to match the previous simulations. Initially only ITI, ISI and AWGN are present so that a fair comparison can also be made with the corresponding PEG code. The results are shown in Figure 4.6 in terms of both BER and frame error rate (FER) for EG(4095, 3367). The code simulations were performed using first a standard BP decoder and then once more using an AD decoder. In Figure 4.6 the result is clear, AD decoding does not provide any decoding performance gain when used with the EG code on the BPMR channel. The decoder's performance is exactly the same as running straight BP decoding. Now the question remains as to why. If the AD decoder is functioning properly, why is there no change in the decoding

performance? This question can be answered if one looks into how the AD decoder's application of the automorphisms affects the EG code.

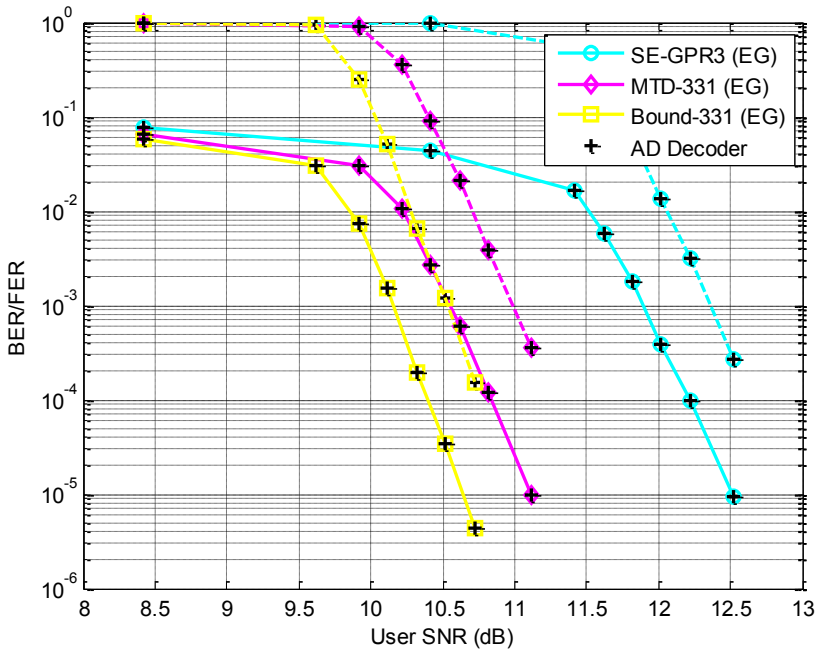

Figure 4.6 – BPMR channel with areal density 4 Tb/in^2 with ITI, ISI and AWGN. BER/FER curves represent both the BP and AD decoding performance with a EG(4095, 3367).

4.5.2 Automorphism Decoding with Finite Geometry LDPC Codes

As was noted previously, finite geometry LDPC codes based on a two-dimensional geometry is normally designated by its parity-check matrix. Since these codes are cyclic, each subsequent row of the parity-check matrices is just a cyclic shift of the initial row. All EG and PG parity-check matrices exhibit this same cyclic symmetry. Thus AD decoding can be applied successfully provided the automorphisms of $\sigma_{A\text{-}F}$ do not result in row images of the parity-check matrix. This would manifest itself as a simple reshuffling of the given rows of the parity-check matrix. To determine if in fact this is the case, both geometries will need to be checked individually.

If the small PG(7, 3) parity-check matrix is subjected to a single shift of the Artin-Frobenius automorphism, denoted $\sigma_{A\text{-}F}{}^{1}$, the resulting parity-check matrix becomes

$$\sigma_{A-F}^{1} H_{PG(7,3)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \tag{4.34}$$

and it can be seen that the rows of this **H** matrix for the PG code simply switch positions under the automorphism. Any automorphism representing a power of $\sigma_{A\text{-}F}{}^{1}$ that is subsequently applied to this **H** matrix will simply produce more row shuffling. In fact for this code with n = 7, only three consecutive permutations, $\sigma_{A\text{-}F}{}^{3}$, are necessary to arrive back at the original matrix. If a row-reduced version (row-reduced here simply implies that some of the rows have been removed) of the original matrix is used instead, such as the one below,

$$H_{PG(7,3)} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}, \tag{4.35}$$

then three consecutive applications of the automorphism will reconstitute the original matrix again. Any number of applied consecutive permutations less than three only acts to reproduce the rows that were removed from the original matrix,

$$\sigma_{A-F}^{1} H_{PG(7,3)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}, \tag{4.36}$$

$$\sigma_{A-F}^{2} H_{PG(7,3)} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \tag{4.37}$$

Note that the second row is immobile under these automorphisms. Obviously one has to be very careful when applying automorphisms. Since all Type-1 PG codes incorporate the same structure regardless of the code length (specified by the parameter *s*) none of them can be used with the AD decoder unless the format of the parity-check matrix is changed. In the case of PG codes, shifted versions of the first row in the parity-check matrix cannot be used in any constructions of a different parity-check matrix for the same code where the Artin-Frobenius automorphism is to be utilized.

After examining PG codes under the Artin-Frobenius automorphism, it is quite possible that the structure incorporated into the Type-1 EG codes may also make them essentially immune to the permutations. Following the same procedure as before, the parity-check matrix for the EG(15, 7) code is subjected to various powers of the $\sigma_{A\text{-}F}$ automorphism

$$\sigma_{A-F}^{1} H_{EG(15,7)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (4.38)$$

Again the rows are simply shuffled by an application of the automorphism and the second row of the **H** matrix is immobile. Row-reducing this matrix to its equivalent full row rank size and then applying the $\sigma_{A\text{-}F}$ automorphism once more yields the following

$$\sigma_{A-F}^{2} H_{EG(15,7)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}. \quad (4.39)$$

It appears that Type-1 EG codes exhibit the same behavior as Type-1 PG codes when the Artin-Frobenius automorphism is applied to the standard form of their parity-check matrices. This means that the AD-version of automorphism decoding cannot be used with the parity-check matrices for the cyclic EG and PG codes. Even if the parity-check matrices are reduced to full-row rank by removing the bottom

redundant rows, there will be no gains from using AD decoding! In order to use automorphisms on these codes either a parity-check matrix must be constructed that does not contain rows which are shifts of any other row or a new set of automorphisms that form a PD-set must be found. The odds of finding a new PD-set are slim and there is no guarantee that such a set exists. It might prove more beneficial to utilize some of the ideas from Hehn et al. to attempt to construct parity-check matrices for these finite geometry codes that can make use of an AD decoder. This would involve finding optimum cyclic orbit generators, which for most short cyclic codes may be feasible. But for codes of even moderate length it is not possible to find all of the minimum weight codewords of the dual codes. In the case of an [n, k, d] cyclic EG or PG code the best that can be done is to generate some low weight codewords based on the parity-generating polynomial. Out of the codewords that can possibly be generated, a set of distinct cyclic orbit generators can be chosen with which to construct a new full row rank matrix **H** for the AD decoder. Unfortunately such a process would involve a computer generated search that may only result in a version of the parity-check matrix that exhibits less decoding performance than the standard form for **H**. As an alternative, there is yet another class of Euclidean geometry codes that are cyclic which can be formulated based on the points and lines in the finite geometry. They possess a much larger density of 1's in their respective parity-check matrices, so they must be considered at least an MDPC code.

4.6 Another Class of Finite Geometry as Cyclic MDPC Codes

Recall that an EG(m, q) over the finite field GF(q) contains

$$F(m,1) = \frac{q^{m-1}(q^m - 1)}{q - 1}, \tag{4.40}$$

lines, each consisting of q points. If point p is contained on line L, then line L is said to intersect at point p. For every point p there are

$$\frac{q^m - 1}{q - 1}, \tag{4.41}$$

lines that intersect it. All of these lines intersecting at point p form an intersecting bundle of lines which are orthogonal at point p. Any other point in EG(m, q) can appear on only one of the lines in the intersecting bundle of lines. Within the subgeometry of EG(m, q) there will be

$$\frac{q^m - 1}{q - 1} - 1\,, \tag{4.42}$$

lines in each intersecting bundle. Lines in EG(m, q) can also be partitioned based on their parallel structure. Lines that do not intersect at a point and that do not share common points can be considered parallel lines. For each line L in EG(m, q) there will be

$$q^{m-1} - 1\,, \tag{4.43}$$

lines that will be parallel to line L. All of the lines in EG(m, q) can be sorted into

$$\frac{q^m - 1}{q - 1}, \tag{4.44}$$

parallel bundles of lines, each containing q^{m-1} parallel lines. For the subgeometry of EG(m, q) this leaves $q^{m-1} - 1$ lines per parallel bundle. This parallel structure can be utilized to formulate another class of cyclic codes.

4.6.1 *Two-fold* EG Codes

While Type-1 EG and PG codes can be used to construct cyclic LDPC codes, there is another class of cyclic finite geometry codes that stems from the same geometries but results in a denser parity check matrix. Such codes make use of the parallel structure of lines in a Type-1 EG code and are called *two-fold* EG codes. These finite geometry codes are cyclic, dense, and possess good minimum distance properties [122]. Starting with a parallel bundle from the subgeometry of EG(m, q), two lines from this bundle will have no points in common. This pairing of two distinct parallel lines constitutes a (1, 2)-frame within the subgeometry and consists of 2q points, q points from each line. By taking a third line from the same parallel bundle, two distinct (1, 2)-frames can be formed, $\{L1, L2\}$ and $\{L1, L3\}$ that intersect at line $L1$. Thus these two (1, 2)-frames both share the q points of line $L1$. From a given parallel bundle in the subgeometry, any selected line can be used to form (1, 2)-frames with all of the other lines in the bundle, resulting in

$$q^{m-1} - 2\,, \tag{4.45}$$

(1, 2)-frames that intersect on the given line. All of the (1, 2)-frames incident on this line are said to be orthogonal on the line as they share no other points in common. Thus a point not on the incident line can only be part of one of the orthogonal (1, 2)-frames. For a given parallel bundle a total of

$$\frac{(q^{m-1}-1)(q^{m-1}-2)}{2}, \tag{4.46}$$

(1, 2)-frames that are distinct can be constructed. This implies that for a given subgeometry of EG(*m*, *q*), the total number of (1, 2)-frames that can be constructed is given by the expression

$$\frac{(q^{m-1}-1)(q^{m-1}-2)(q^{m}-1)}{2(q-1)}. \tag{4.47}$$

To establish the cyclic property of the resulting *two-fold* EG code, take a single (1, 2)-frame from the subgeometry and designate it

$$K_i = \{L_j, L_k\}; j \neq k\,. \tag{4.48}$$

For a binary code, the geometry is taken over GF(2) and an incidence vector $v(K_i)$ for K_i can be defined as a (q^m – 1)-tuple over GF(2),

$$v(K_i) = \left(v_0, v_1, v_2, \cdots, v_{q^m-2}\right). \tag{4.49}$$

The (q^m – 2) vector components correspond to zeros and ones with the ones corresponding to the points contained in K_i. Since (1, 2)-frames are formed from the lines in EG(*m*, *q*) and these lines possess a cyclic structure, the (1, 2)-frames will also

possess the cyclic property. This can be shown as follows. If L is a line from EG(*m*, *q*) and α a primitive element of the extension field used to represent points in EG(*m*, *q*), then the cyclic nature of the lines in EG(*m*, *q*) assures that αL is also a line in EG(*m*, *q*). This property translates directly to (1, 2)-frames within EG(*m*, *q*) and assures that

$$\alpha K_i = \{\alpha L_j, \alpha L_k\}; j \neq k\ , \tag{4.50}$$

is also a (1,2)-frame in EG(*m*, *q*). This means that the left or right shift of any incidence vector $v(K_i)$ for a (1, 2)-frame is a valid incidence vector for a different (1, 2)-frame. To utilize this cyclic structure as a linear block code, a binary parity check matrix is formed from the incidence vectors of all (1, 2)-frames within the subgeometry of EG(*m*, *q*). With the following two designations,

$$\#\ \text{of rows} = r = \frac{(q^{m-1}-1)(q^{m-1}-2)(q^m-1)}{2(q-1)}, \tag{4.51}$$

$$\#\ \text{of columns} = n = \ q^m - 1\ . \tag{4.52}$$

The null space of the *r* x *n* parity check matrix formed from the incidence vectors of all the (1, 2)-frames produces a cyclic code of length *n*. Codes of this kind are referred to as *two-fold* EG codes. In short, *two-fold* EG codes are constructed by partitioning the parallel lines found in the corresponding EG(2, *q*). These parallel lines then form a (1, 2)-*frame* of EG(2, *q*). The incidence vectors of these (1, 2)-frames can be used as the rows of a parity-check matrix **H**. Since the right-cyclic shift of one of these incidence vectors corresponds to another valid incidence vector of

another (1, 2)-frame, the frames are cyclic in nature and so is the corresponding parity-check matrix. It is only necessary to generate one incidence vector and then shift it *n* times to construct a valid **H** matrix. When compared with standard EG codes, *two-fold* EG codes are slightly higher rate codes which are characterized by slightly smaller minimum distances. The following table of parameters provides a way of comparing them to the standard cyclic EG and PG codes.

Table 4.7 – Examples of cyclic *two-fold* EG(2, *q*) codes.

s	(n, k)	d_{min}	ρ ; γ	R
2	(15, 11)	3	8	0.7333
3	(63, 45)	7	16	0.7143
4	(255, 191)	15	32	0.7490
5	(1023, 813)	31	64	0.7947
6	(4095, 3431)	63	128	0.8379

Even though *two-fold* EG codes are all very similar to the usual EG codes, their parity-check matrices are denser and plagued by cycles of length four. This makes them unsuitable for decoding based on belief propagation and would normally only be majority-logic decodable. However, auto-diversity decoding works well with codes characterized by parity-check matrices that are typically categorized as MDPC codes. As can be seen from Table 4.7 above, the parity-check matrices of *two-fold* EG codes contain twice the number of ones as their contemporary EG code's **H** matrices. It is because of this fact that the Artin-Frobenius automorphism can be used to some advantage because the applied permutations do not strictly result in shuffling of the rows.

4.6.2 More Simulations with AD Decoding

Armed with the knowledge of how AD decoders work, some simulations involving an AD decoder used with both Type-1 EG and *two-fold* EG codes can be attempted. All simulations that follow were performed on a binary input additive white Gaussian noise channel. The definition of the signal-to-noise ratio (SNR) follows as well as the adjustment from SNR to E_b/N_o:

$$\mathrm{SNR} = \frac{1}{\sigma^2}\,, \tag{4.53}$$

$$\frac{E_b}{N_o} = \mathrm{SNR} - 10\log(2R)\,, \tag{4.54}$$

where R represents the code rate. The following simulation results (see Figures 4.7 – 4.9) show the decoding performances in terms of the BERs and FERs of the BP decoder in the form of the sum-product algorithm and the AD decoder for different *two-fold* EG codes. A comparison between the performances of SPA and AD decoding of *two-fold* EG(63,45), *two-fold* EG(255,191), and *two-fold* EG(1023,813) reveals that AD decoding provides a noticeable gain over SPA (~ 1dB at 10^{-5} BER) for the larger codes. For the smaller *two-fold* EG(63,45) there is very little observable difference in the performance of the two algorithms. The reason for the gain in the larger codes is that the AD decoder better handles the dense parity-check matrices of *two-fold* EG codes compared to strict SPA decoding. The combination of shifting the error patterns of the received codeword vector inside the AD decoder and

running the SPA through many more iterations in the process, results in the better decoding performance.

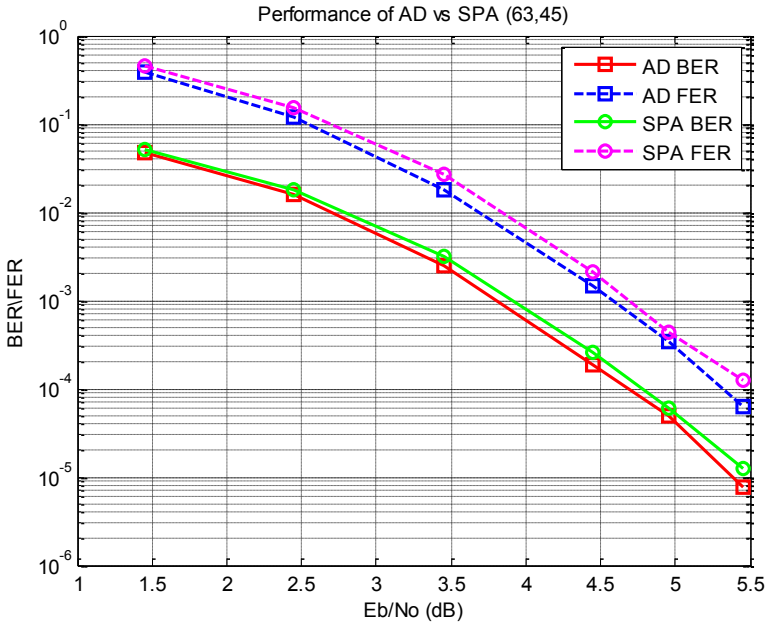

Figure 4.7 – BER/FER decoding performance curves for both the SPA and AD decoders on an AWGN channel with the *two-fold* EG(63, 45) code.

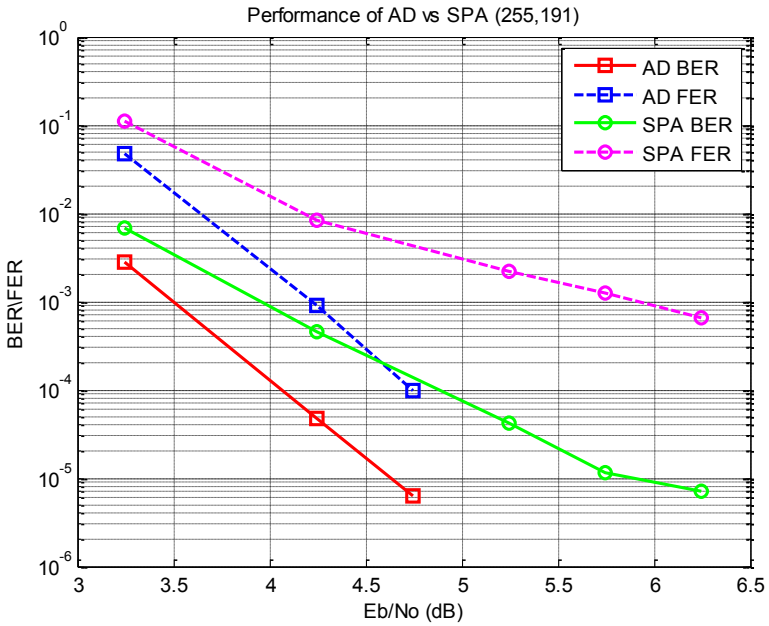

Figure 4.8 – BER/FER decoding performance curves for both the SPA and AD decoders on an AWGN channel with the *two-fold* EG(255, 191) code.

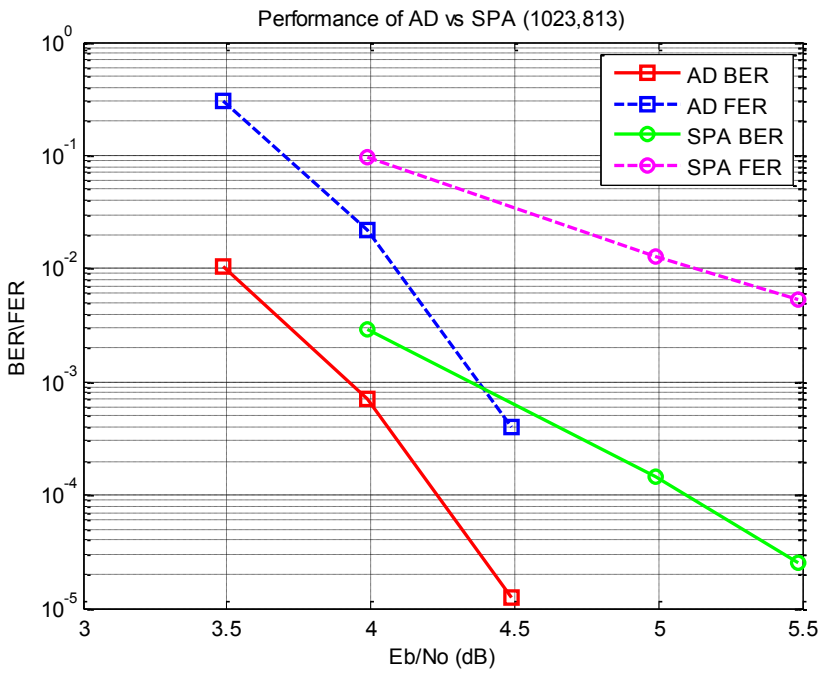

Figure 4.9 – BER/FER decoding performance curves for both the SPA and AD decoders on an AWGN channel with the *two-fold* EG(1023, 813) code.

Based on previous arguments about changing the parity-check matrix of a cyclic finite geometry code to possibly enhance decoding performance, the initial "square" parity-check matrix of the *two-fold* EG(255,191) code was modified by reducing it to a full-rank matrix. For such dense codes this could prove advantageous to the decoder if the performance loss is negligible when compared to the decoding iterations saved by utilizing far fewer checks in the parity-check equations. In the simulations both square and full-rank versions of the **H** matrix are used with the AD and SPA decoders. The results in terms of BER only are shown in Figure 4.10. It can be noted that AD decoding still outperforms SPA decoding, but the "full-rank" matrix representation does not perform better than the "square" matrix representation.

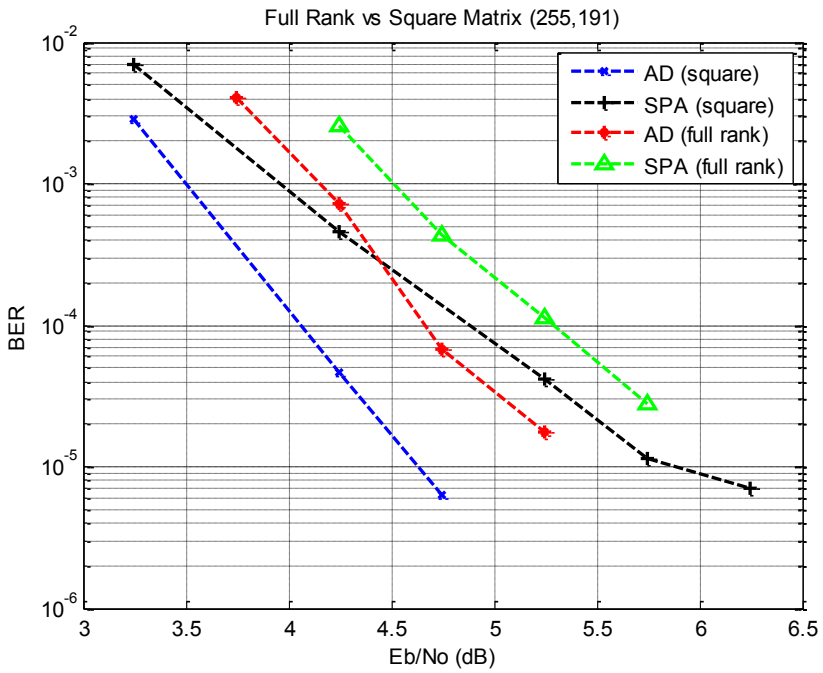

Figure 4.10 – BER decoding performance curves for both the SPA and AD decoders on an AWGN channel with the *two-fold* EG(255, 191) code with both "square" and "full-rank" versions of the parity-check matrix.

It seems the "full-rank" version suffers from the trimming away of too many rows to compete with the "square" version. In essence the matrix has not been modified in a beneficial manner. It is possible that the "full-rank" matrix is nothing more than a truncated version of the "square" matrix so the same performance gain is provided to both matrices under AD decoding. The greater number of rows in the "square" matrix in combination with the structure in the code is providing the larger gain. The same sort of experiment can be used with a Type-1 EG code as well. In the following simulations the performance of a *two-fold* EG code and an EG code are shown (see Figure 4.11). The EG code has been divided into both a full-rank (denoted "FR") version and the standard "square" version based only on differences in parity-check matrices.

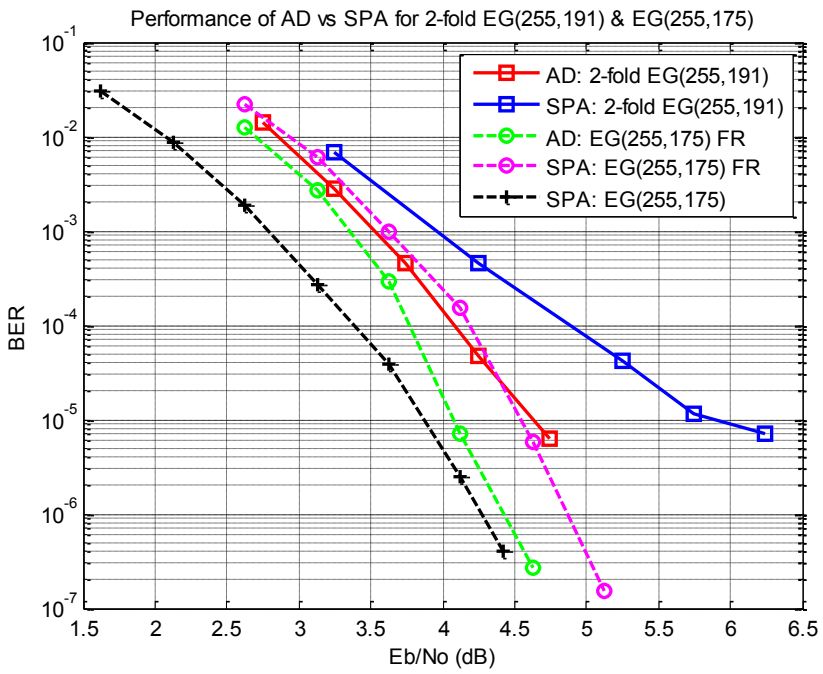

Figure 4.11 – BER decoding performance curves for both the SPA and AD decoders on an AWGN channel with both the *two-fold* EG(255, 191) code and the EG(255, 175) code. A "square" and a "full-rank" version of the parity-check matrix is utilized by the EG code.

In the curves above there is a leveling off in the waterfall region of the "full-rank" parity-check matrix before reaching the performance of the "square" matrix version for the EG code. The difference in performance between the two curves at a BER of 10^{-6} is less than 0.25 dB. Based on the analysis of how the Artin-Frobenius automorphism works with Type-1 EG and PG codes, the green and black curves are close to converging because the AD decoder is recreating virtual rows of the parity-check matrix that were chopped off during the reduction to a "full-rank" matrix. In the extreme where the AD decoder runs through all possible shifts of the parity-check matrix under the Artin-Frobenius automorphism, the green curve would lie on top of the black curve, indicating identical performance. Strictly speaking this would

only be true at the iteration point where further runs of the SPA would produce no additional gains in the decoding performance.

While two-fold EG codes possess a number of nice properties, they do not appear to outperform their cyclic EG LDPC contemporaries on an AWGN channel. This does not mean that their usefulness on the BPMR channel is wholly predetermined. But given the number of structural similarities between the types of codes, it is unlikely that the AD decoder paired with a two-fold EG code will outperform a similar EG or PG code on any given channel. Unfortunately, none of the coding methods previously described solves the problem caused by the presence of insertions and deletions within the BPMR write process.

Chapter 5 – Conclusion

The challenges faced by engineers attempting to bring BPMR from concept to profitable product are significant and multifaceted. Not the least of which are its modeling and signal processing problems. For BPMR to progress as a possible avenue for future HDDs, the technology must successfully circumvent its many standing problems.

5.1 Concluding Remarks on BPMR

One of the most significant problems faced by BPMR resides in the requirement to synchronize the write clock to the locations of individual islands to within a tiny fraction of an error. At the areal recording densities of interest, current servo motor tolerances in combination with other electrical and mechanical limitations cannot place the write head to within the margins required by the write process. Thus the introduction of insertion and deletion errors into the storage channel is currently assured. While error-correction codes are available, none can currently handle the level of written-in errors expected within the channel at coding rates appropriate for HDDs. The solution of this problem still awaits in the future and will likely be solved by a combination of decreasing write head design tolerances and targeting ECCs that can handle any remaining levels of written-in errors.

For the most part, this dissertation was able to identify and model the BPMR channel in conjunction with all of its major error sources. The use of a 2D Gaussian pulse to model the channel response was shown to work well within simulations

while incorporating the effects of the fabrication jitter which manifested itself as both position jitter and island size variations. The simulations with multi-track detection revealed its ability to greatly neutralize the detrimental effects of both high levels of ISI and ITI present in the channel at high areal densities, but it suffers greatly in high levels of media noise present in BPMR. Given that the noise prediction used with single-track equalization provided some improvement to the 1D method's dismal performance in high ITI conditions, it should be incorporated in some fashion into the MTD technique. An attempt to model the written-in errors was made which identified the extent to which these kinds of errors can degrade the overall performance of the channel. Despite various coding techniques, the worst of the written-in errors still cannot be effectively handled. The class of cyclic finite geometry LDPC codes was tapped to show the impressive levels of decoding performance possible with these codes for the BPMR channel when faced with all but insertion and deletion errors. Automorphism decoding in the form of auto-diversity decoding was explored with various finite geometry codes, including a lesser-known class of cyclic code known as *two-fold* EG codes. It was shown that AD decoding improves the performance of these codes and allows for iterative decoding of this type of MDPC code. It was also discovered that the cyclic finite geometry codes are not compatible with current methods of automorphism decoding as there structure invalidates the two known applicable automorphisms for cyclic codes. It was also determined by investigation that reduced versions of the parity-check

matrices of finite geometry codes do not offer any benefit to the decoder in automorphism decoding.

5.2 Suggestions for Future Research

During the formulation of this dissertation a number of interesting questions arose out of the many thought-provoking challenges which ensued. Several of these instances are worth mentioning and hopefully will inspire someone to investigate a problem that needs solving.

1) The insertion and deletion types of written-in errors are extremely difficult to handle when they occur in multiple random positions throughout a long sector. Low rate codes can be devised to correct for multiple instances of these errors, but this does not help for an application where extended latency should be avoided. One has to wonder if an algorithm can be devised that attempts to shift the bits in local neighborhoods and then checks the outcomes against the resulting parity checks to determine if significantly fewer bits are in error. The algorithm would only need to localize a given error and then treat the neighborhood of surrounding bits as an erasure.
2) While the Auto-Diversity decoder works with only cyclic codes, many quasi-cycle codes are simply formed from circulants of cyclic codes. It seems reasonable that a quasi-cyclic version of the AD decoder could be formulated that would operate on the individual circulants while also decoding the

entire codeword as a whole. This would certainly increase the decoder's applicability.

3) Noise prediction is a powerful tool for extending the detection algorithm's performance and it works best when the properties of the noise are known. Thus a better description of the fabrication media noise introduced as jitter into the channel would likely help a great deal in noise predictive techniques applied to the BPMR channel.
4) TDMR techniques are an obvious forward step for signal processing. Unfortunately, the complexity of many of the one-dimensional algorithms used in detection grows exponentially when expanded into two dimensions. Thus these algorithms can only be approximated in any real sense under TDMR. Can efficient alternatives to the 2D BCJR algorithm be formulated that effectively allow for noise prediction in MTD?
5) Only a few binary LDPC finite geometry codes can be constructed which are also cyclic codes. From the known cyclic FG codes, descendent codes can be constructed that are also cyclic and less dense than the parent code. Are these codes subject to the same constraint as the parent codes with regard to their use in AD decoding?
6) The switching field distribution in BPMR can have a significant detrimental impact on the writing process. But as of yet there has been no work describing how the distribution is affected by different island patterns across various increases in areal densities with respect to BAR = 1 or BAR = 2. This

could heavily impact a hexagonal island arrangement where more islands are in greater proximity.

Bibliography

Chapter 1 – Advances in Magnetic Storage Channels

1.1 Current State of HDD Magnetic Recording

[1] R. E. Rottmayer, "Head design techniques for recording devices," in *Coding and Signal Processing for Magnetic Recording Systems*, B. Vasic and E. M. Kurtas, Eds. Boca Raton, FL: CRC Press, LLC, 2005, pp. 4-1 – 4-17.

[2] H. Sawaguchi, Y. Nishida, H. Takano, and H. Aoi, "Performance analysis of modified PRML channels for perpendicular recording systems," *J. Magnetism Magn. Materials*, vol. 235, pp. 265–272, Oct. 2001.

[3] W. Chang, "Advanced signal processing for magnetic recording on perpendicularly magnetized media," Ph.D. dissertation, Dept. Elect. and Comput. Eng., The Univ. of Oklahoma, Norman, 2010.

[4] R. M. Todd, "A model for a perpendicular magnetic recording channel and equalizer," CSP Lab, The Univ. of Oklahoma, Tech. Rep., 2009.

1.2 Limiting Factors for Magnetic Storage Channels

[5] H. J. Richter, "Recent advances of the recording physics of thin-film media," *J. Physics D: Appl. Physics*, vol. 32, pp. R147–R168, 1999.
and H. J. Richter, Erratum to "Recent advances of the recording physics of thin-film media," *J. Physics D: Appl. Physics*, vol. 32, p. 3092, 1999.

[6] H. J. Richter, "The transition from longitudinal to perpendicular recording," *J. Physics D: Appl. Physics*, vol. 40(9), pp. R149–R177, 2007.

[7] R.Wood, "The feasibility of magnetic recording at 1 terabit per square inch," *IEEE Trans. Magn.*, vol. 36, no. 1, pp. 36–42, Jan. 2000.

[8] R. Wood, M. Williams, A. Kavcic, and J. Miles, "The feasibility of magnetic recording at 10 terabits per square inch on conventional media," *IEEE Trans. Magn.*, vol. 45, no. 2, pp. 917–923, Feb. 2009.

1.3 Shingled Write Recording with TDMR

[9] S. J. Greaves, H. Muraoka, and Y. Kanai, "Simulations of recording media for 1 Tb/in^2," *J. Magnetism Magn. Materials*, vol. 320, pp. 2889–2893, 2008.

[10] S. Greaves, Y. Kanai, and H. Muraoka, “Shingled recording for 2-3 Tbit/in^2,” *IEEE Trans. Magn.*, vol. 45, no. 10, pp. 3823–3829, Oct. 2009.

[11] W. Tan and J. R. Cruz, “Evaluation of detection algorithms for perpendicular recording channels with intertrack interference,” *J. Magnetism Magn. Materials,* vol. 287, pp. 397–404, 2005.

[12] A. R. Krishnan et al., “Two-dimensional magnetic recording: Read channel modeling and detection,” in *Proc. IEEE Int. Magnetics Conf.*, Sacramento, CA, 2009, vol. 45, pp. 3830–3836.

[13] A. Amer, D. D. E. Long, E. L. Miller, J.-F. Paris, and T. Schwarz, “Design issues for a shingled write disk system,” in *Proc. 26th IEEE Symp. Massive Storage Systems and Technologies*, 2010 © IEEE. doi: 10.1109/MSST.2010.5496991

[14] K. S. Chan, J. J. Miles, E. Hwang, B. V. K. VijayaKumar, J.-G. Zhu, W.-C. Lin, and R. Negi, “TDMR platform simulations and experiments,” *IEEE Trans. Magn.*, vol. 45, no. 10, pp. 3837–3843, Oct. 2009.

[15] B. Kurkoski, “Towards efficient detection of two-dimensional intersymbol interference channels,” *IEICE Trans. Fundamentals*, vol. E91-A, no. 10, pp. 2696–2703, Oct. 2008.

1.4 Heat-Assisted Magnetic Recording

[16] R. Radhakrishnan, B. Vasic, F. Erden, and C. He, “Characteristics of heat-assisted magnetic recording channels,” in *Advances in Information Recording*, DIMACS Series vol. 73, P. Siegel, E. Soljanin, A. J. van Wijngaarden and B. Vasic, Eds. Providence, RI: American Mathematical Society, 2008, pp. 23–39.

[17] M. F. Erden, T. Rausch, and W. A. Challener, “Cross-track location and transition parameter effects in heat-assisted magnetic recording,” *IEEE Trans. Magn.*, vol. 41, no. 6, pp. 2189–2194, Jun. 2005.

[18] T. Rausch, “Thermal Williams-Comstock model for predicting transition length in a heat assisted magnetic recording system,” *IEEE Trans. Magn.*, vol. 40, no. 1, pp. 137–147, Jan. 2004.

[19] M. H. Kryder, E. C. Gage, T. W. McDaniel, W. A. Challener, R. E. Rottmayer, G. Ju, Y.-T. Hsia, and M. F. Erden, “Heat assisted magnetic recording,” *IEEE Trans. Magn.*, vol. 96, no. 11, pp. 1810–1835, Nov. 2008.

[20] T. W. McDaniel, W. A. Challener, and K. Sendur, “Issues in heat-assisted perpendicular recording,” *IEEE Trans. Magn.*, vol. 39, no. 4, pp. 1972–1979, Jul. 2003.

[21] Seagate Technology LLC. (2012, Mar. 19). *Seagate reaches 1 terabit per square inch milestone in hard drive storage with new technology demonstration*. Press release [Online]. Available: http://media.seagate.com/

[22] H. Muraoka and S. J. Greaves, "Statistical modeling of write error rates in bit patterned media for 10 Tb/in^2 recording," *IEEE Trans. Magn.*, vol. 47, no. 1, pp. 26–34, Jan. 2011.

1.5 Microwave-Assisted Magnetic Recording

[23] J.-G. Zhu, X. Zhu, and Y. Tang, “Microwave assisted magnetic recording,” *IEEE Trans. Magn.*, vol. 44, no. 1, pp. 125–131, Jan. 2008.

[24] Y. Nozaki, A. Kato, K. Noda, Y. Kanai, T. Tanaka, and K. Matsuyama, "Micromagnetic study on microwave-assisted magnetic recording in perpendicular medium with intergrain exchange coupling," *J. Appl. Physics*, vol. 109, pp. 123912–1:6 , 2011.

[25] S. Okamoto, N. Kikuchi, and O. Kitakami, “Magnetization switching behavior with microwave assistance,” *Appl. Physics Lett.*, vol. 93, pp. 102506–1:3, 2008.

[26] Y. Shiroishi, K. Fukuda, I. Tagawa, H. Iwasaki, S. Takenoiri, H. Tanaka, H. Mutoh, and N. Yoshikawa, “Future options for HDD storage,” *IEEE Trans. Magn.*, vol. 45, no. 10, pp. 3816–3822, Oct. 2009.

1.6 Bit-Patterned Magnetic Recording

[27] T. R. Albrecht, O. Hellwing, R. Ruiz, M. E. Schabes, B. D. Terris, and X. Z. Wu, “Bit-patterned magnetic recording: nanoscale magnetic islands for data storage,” in *Nanoscale Magnetic Materials and Applications*, J. P. Liu et al., Eds. New York, NY: Springer Science + Business Media, LLC, 2009, pp. 237–273.

[28] B. D. Terris and T. Thomson, "Nanofabricated and self-assembled magnetic structures as data storage media," *J. Physics D: Appl. Physics*, vol. 38, pp. R199–222, 2005.

[29] M. K. Grobis, O. Hellwig, T. Hauet, E. Dobisz, and T. R. Albrecht, “High-density bit patterned media: Magnetic design and recording performance,” *IEEE Trans. Magn.*, vol. 47, no. 1, pp. 6–10, Jan. 2011.

1.6.1 Bit Aspect Ratio and Grain Volume

[30] S. Y. Chou, P.R. Krauss, and L.S. Kong, "Nanolithographically defined magnetic structures and quantum magnetic disk," *J. Appl. Physics*, vol. 79(8), pp. 6101–6106, 1996.

[31] R. M. H. New, R. F. W. Pease, and R. L. White, "Lithographically patterned single-domain cobalt islands for high-density magnetic recording," *J. Magnetism Magn. Materials*, vol. 155(1–3), pp. 140–145, 1996.

[32] B. D. Terris, “Fabrication challenges for patterned recording media,” *J. Magnetism Magn. Materials*, vol. 321, pp. 512–517, 2009.

[33] J. Kalezhi, J. J. Miles, and B. D. Belle, “Dependence of switching fields on island shape in bit patterned media,” *IEEE Trans. Magn.*, vol. 45, no. 10, pp. 3531–3534, Oct. 2009.

[34] R. L. White, R. M. H. New and R. F. W. Pease, “Patterned media: A viable route to 50 Gbit/in and up for magnetic recording,” *IEEE Trans. Magn.*, vol. 33, no. 1, pp. 990–995, Jan. 1997.

[35] G. F. Hughes, “Read channels for patterned media,” *IEEE Trans. Magn.*, vol. 35, no. 5, pp. 2310–2312, Sep. 1999.

[36] D. Weller, A. Moser, L. Folks, M. E. Best, W. Lee, M. F. Toney, J.-U. Thiele, and M. F. Doerner, “High K_u materials approach to 100 Gbit/in^2,” *IEEE Trans. Magn.*, vol. 36, no. 1, pp. 10–15, Jan. 2000.

1.6.2 Writing to BPMR Islands

[37] J. Hu, T. M. Duman, E. M. Kurtas, and M. F. Erdan, "Bit-patterned media with written-in errors: Modeling, detection, and theoretical limits," *IEEE Trans. Magn.*, vol. 43, no. 8, pp. 3517–3524, Aug. 2007.

[38] J. Hu, T. M. Duman, M. F. Erden, and A. Kavcic, “Achievable information rates for channels with insertions, deletions and intersymbol interference with i.i.d. inputs,” *IEEE Trans. Commun.*, vol. 58, no. 4, pp. 1102–1111, Apr. 2010.

[39] A. R. Iyengar, P. H. Siegel, and J. K. Wolf, “Write channel model for bit-patterned media recording,” *IEEE Trans. Magn.*, vol. 47, no. 1, pp. 35–45, Jan. 2011.

[40] M. E. Schabes, "Micromagentic simulations for Tb/in^2 recording systems," *J. Magnetism Magn. Materials*, vol. 320(22), pp. 2880–2884, 2008.

[41] H. J. Richter, A. Y. Dobin, K. Z. Gao, R. J. M. v. d. Veerdonk, R. T. Lynch, J. Xue, D. Weller, P. Asselin, M. F. Erden, and R. M. Brockie, “Recording on bit-patterned media at densities of 1 Tb/in^2 and beyond,” *IEEE Trans. Magn.*, vol. 42, no. 10, pp. 2255–2260, Oct. 2006.

[42] P. Chicoine, M. Hassner, E. Grochowski, S. Jenness, M. Noblitt, G. Silvus, C. Stevens, and B. Weber, "Hard Disk Drive Long Data Sector White Paper," IDEMA, Tech. Rep., Apr. 2007.

[43] S. Zhang, K. S. Chai, K. Cai, B. Chen, Z. Qin, and S. M. Foo, “Write failure analysis for bit-patterned-media recording and its impact on read channel modeling,” *IEEE Trans. Magn.*, vol. 46, no. 6, pp. 1363–1365, Jun. 2010.

[44] Y. Nakamura, Y. Okamoto, H. Osawa, H. Aoi, and H. Muraoka, “A study of LDPC coding and iterative decoding system in magnetic recording system using bit-patterned medium with write error,” *IEEE Trans. Magn.*, vol. 45, no. 10, pp. 3753–3756, Oct. 2009.

1.6.3 Reading from BPMR Islands

[45] P. W. Nutter, I. T. Ntokas, and B. K. Middleton, “An investigation of the effects of media characteristics on read channel performance for patterned media storage,” *IEEE Trans. Magn.*, vol. 41, no. 11, pp. 4327–4334, Nov. 2005.

1.7 Dissertation Overview

[46] W. Chang and J. R. Cruz, “Inter-track interference mitigation for bit-patterned magnetic recording,” *IEEE Trans. Magn.*, vol. 46, no. 11, pp. 3899–3908, Nov. 2010.

[47] K. Cai, Z. Qin, S. Zhang, Y. Ng, K. S. Chai, and R. Rathnakumar, “Modeling, detection and LDPC codes for bit-patterned media recording,” in *IEEE Global Communications Conf., Exhibition & Industry Forum Workshops*, Miami, FL, 2010, pp. 1910–1914.

Chapter 2 - Modeling the BPMR Channel

2.1 Elements of a Magnetic Recording System

[48] A. J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Trans. Inf. Theory*, vol. IT-13, pp. 260–269, Apr. 1967.

[49] G. D. Forney, Jr., “The Viterbi algorithm,” *Proc. IEEE*, vol. 61, pp. 268–278, Feb. 1973.

[50] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inf. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[51] J. Hagenauer and P. Hoeher, “A Viterbi algorithm with soft-decision outputs and its applications,” in *Proc. IEEE Global Communications Conf., Exhibition & Industry Forum*, Dallas, TX, 1989, pp. 1680–1686.

2.2 The Readback Signal

[52] S. Khizroev, Y. Liu, K. Mountfield, M. H. Kryder, and D. Litvinov, “Physics of perpendicular magnetic recording: Writing process,” *J. Magnetism Magn. Materials*, vol. 246, pp. 335–344, 2002.

[53] K. Yamakawa, H. Muraoka, K. Fudano, S. J. Greaves, Y. Ohsawa, K. Ise, and Y. Nakamura, “High field-gradient design of single-pole writehead with planar pole structure,” *IEEE Trans. Magn.*, vol. 46, no. 3, pp. 730–737, Mar. 2010.

[54] J.-G. Zhu, D. Z. Bai, and A. F. Torabi, “The role of SUL in readback and effect on linear density performance for perpendicular recording,” *IEEE Trans. Magn.*, vol. 39, pp. 1961–1966, Jul. 2003.

[55] S. Nabavi, B. V. K. V. Kumar, and J. A. Bain, “Two-dimensional pulse response and media noise modeling for bit-patterned media,” *IEEE Trans. Magn.,* vol. 44, no. 11, pp. 3789–3792, Nov. 2008.

[56] P. W. Nutter, D. M. McKirdy, B. K. Middleton, D. T. Wilton, and H. A. Shute, "Effect of island geometry on the replay signal in patterned media storage," *IEEE Trans. Magn.*, vol. 40, no. 6, pp. 3551–3558, Nov. 2004.

[57] S. W. Yuan and H. N. Bertram, “Off-track spacing loss of shielded MR heads,” *IEEE Trans. Magn.*, vol. 30, no. 3, pp. 1267–1273, May 1994.

[58] S. W. Yuan and H. N. Bertram, “Correction to ‘Off-track spacing loss of shielded MR heads’,” *IEEE Trans. Magn.*, vol. 32, no. 4, Jul. 1996.

[59] S. Nabavi, “Signal processing for bit-patterned media channels with inter-track interference”, Ph.D. dissertation, Dept. Elect. Eng. and Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, 2008.

[60] S.H. Zhang, Z.L. Qin, and X.X. Zou, “Signal modeling for ultra-high density bit-patterned media and performance evaluation,” in Dig. Tech. Papers *IEEE Int. Magnetics Conf.*, Madrid, Spain, 2008, pp. 1516–1517.

[61] J. J. M. Ruigrok, *Short-Wavelength Magnetic Recording: New Methods and Analyses*. Oxford, U.K.: Elsevier, 1990.

[62] K. Wiesen and B. Cross, “GMR head side-reading and bit aspect ratio,” *IEEE Trans. Magn.*, vol. 39, no. 5, pp. 2609–2611, Nov. 2003.

[63] D. A. Lindholm, “Magnetic fields of finite track width heads,” *IEEE Trans. Magn.*, vol. 13, pp. 1460–1462, Sep. 1977.

[64] I. T. Ntokas, P. W. Nutter, C. J. Tjhai, and M. Z. Ahmed, "Improved data recovery from patterned media with inherent jitter noise using low-density parity-check codes," *IEEE Trans. Magn.*, vol. 43, pp. 3925–3929, Oct. 2007.

[65] 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 Trans. Magn.*, vol. 40, no. 1, pp. 148–156, Jan. 2004.

2.2.1 Modeling with a 2D Gaussian Pulse

[66] W. Chang and J. R. Cruz, “Intertrack interference mitigation on staggered bit-patterned media,” *IEEE Trans. Magn.*, vol. 47, no. 10, pp. 2551–2554, Oct. 2011.

[67] Y. Ng, K. Cai, B. V. K. Vijaya Kumar, S. Zhang, and T. C. Chong, “Modeling and two-dimensional equalization for bit-patterned media channels with media noise,” *IEEE Trans. Magn.*, vol. 45, no. 10, pp. 3535–3538, Oct. 2009.

[68] Y. Ng, B. V. K. V. Kumar, K. Cai, S. Nabavi, and T. C. Chong, “Picket-shift codes for bit-patterned media recording with insertion/deletion errors,” *IEEE Trans. Magn.*, vol. 46, no. 6, pp. 2268–2271, Jun. 2010.

2.2.2 Media Noise and Its Effects on the Readback Response

[69] Y. J. Shi, P. W. Nutter, B. D. Belle, and J. J. Miles, "Error events due to island size variations in bit patterned media," *IEEE Trans. Magn.*, vol. 46, no. 6, pp. 1755–1758, May 2010.

2.3 Defining the SNR

[71] X. Yang and E. M. Kurtas, "Signal and noise generation for magnetic recording channel simulations", in *Coding and Signal Processing for Magnetic Recording Systems*, B. Vasic and E. M. Kurtas, Eds. Boca Raton, FL: CRC Press, LLC, 2005, pp. 6-1–6-20.

[72] W.E. Ryan and S. Lin, "*Channel Codes – Classical and Modern*," Cambridge, U. K.: Cambridge Univ. Press, 2009.

2.4 Errors in the Writing Process

[73] H. Muraoka , S. J. Greaves, and Y. Kanai "Modeling and simulation of the writing process on bit-patterned perpendicular media", *IEEE Trans. Magn.*, vol. 44, pp. 3423–3429, Nov. 2008.

[74] S. Zhang, K. Cai, M. Lin-Yu, J. Zhang, Z. Qin, K. K. Teo, W. E. Wong, and E. T. Ong, "Timing and written-in errors characterization for bit patterned media," *IEEE Trans. Magn.*, vol. 47, pp. 2555–2558, Oct. 2011.

2.5 Parameter Sensitivity and Selection

[75] K. Cai, private communication, Jul. 2010.

Chapter 3 - Detection Strategies for BPMR

3.1 Single-Track Equalization

[76] J. Fitzpatrick, J. K. Wolf, and L. Barbosa, "New equalizer targets for sampled magnetic recording systems,'' in *Proc. 25th Asilomar Conf. on Signals, Systems and Computers*, 1991, pp. 30–34.

[77] J. Moon and W. Zeng, "Equalization for maximum likelihood detectors", *IEEE Trans. Magn.*, vol. 31, no. 2, pp. 1083–1088, Mar. 1995.

3.2 The BCJR Algorithm

[78] D. Fertonani, A. Barbieri, and G. Colavolpe, "Reduced-complexity BCJR algorithm for turbo equalization," *IEEE Trans. Commun.*, vol. 55, pp. 2279–2287, Dec. 2007.

[79] G. Colavolpe, G. Ferrari, and R. Raheli, "Reduced-state BCJR-type algorithms," *IEEE J. Sel. Areas Commun.*, vol. 19, pp. 848–859, May 2001.

3.3 TDMR with Multi-Track Detection Techniques

[80] R. Wood, "Shingled magnetic recording and two-dimensional magnetic recording," presentation for *IEEE Magnetics Society - Santa Clara Valley Chapter*, Oct. 2010.

[81] L. C. Barbosa, “Simultaneous detection of readback signals from interfering magnetic recording tracks using array heads,” *IEEE Trans. Magn.*, vol. 26, no. 5, pp. 2163–2165, Sep. 1990.

3.3.1 MTD with 2D Equalization

[82] M. Keskinoz, “Two-dimensional equalization/detection for patterned media storage,” *IEEE Trans. Magn.*, vol. 44, no. 4, pp. 533–539, Apr. 2008.

3.3.2 Expanded-State BCJR for TDMR

[83] W. Tan and J. R. Cruz, “Signal processing for perpendicular recording channels with intertrack interference,” *IEEE Trans. Magn.*, vol. 41, no. 2, pp. 730–735, Feb. 2005.

[84] E. Chesnutt, “Novel turbo equalization methods for the magnetic recording channel,” Ph.D. dissertation, Dept. Elect. and Comput. Eng., Georgia Inst. of Technol., Atlanta, 2005.

3.3.3 Joint-Track Equalization with MTD

[85] B. G. Roh, S. U. Lee, J. Moon, and Y. Chen, “Single-head/single-track detection in interfering tracks,” *IEEE Trans. Magn.*, vol. 38, no. 4, pp. 1830–1838, Jul. 2002.

[86] G. Kong and S. Choi, "Simplified multi-track detection schemes using *a priori* information for bit patterned media recording," *J. Appl. Physics*, vol. 111, issue 7, in *Proc. 56th Annu. Conf. Magnetism and Magn. Materials - Patterned Films*, pp. 07B920-1 – 3, Apr. 2012.

[87] S. Karakulak, P. H. Siegel, J. K. Wolf, and H. N. Bertram, "Joint-track equalization and detection for bit patterned media recording," *IEEE Trans. Magn.*, vol. 46, no. 9, pp. 3639–3647, Sep. 2010.

3.4 Noise Prediction

[88] P. R. Chevillat, E. Eleftheriou, and D. Maiwald, "Noise-predictive partial response equalizers and applications," in *Proc. IEEE Int. Conf. Communications*, Chicago, IL, 1992, pp. 942–947.

[89] J. D. Coker, E. Eleftheriou, R. L. Galbraith, and W. Hirt, "Noise-predictive maximum likelihood (NPML) detection," *IEEE Trans. Magn.*, vol. 34, pp. 110–117, Jan. 1998.

[90] A. Kavcic and J. M. F. Moura, "Correlation-sensitive adaptive sequence detection," *IEEE Trans. Magn.*, vol. 34, pp. 763–771, May 1998.

[91] A. Kavcic and J. M. F. Moura, "The Viterbi algorithm and Markov noise memory," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 291–301, Jan. 2000.

[92] J. Moon and J. Park, "Pattern-dependent noise prediction in signal-dependent noise," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 4, pp. 730 –743, Apr. 2001.

[93] A. Kavcic, "Soft-output detector for channels with intersymbol interference and Markov noise memory," in *Proc. IEEE Global Communications Conf., Exhibition & Industry Forum*, Rio de Janeiro, Brazil, 1999, pp. 728–732.

[94] P. Vanichchanunt, K. Woradit, S. Nakpeerayuth, L. Wuttisittikulkij, and L. Hanzo, "Noise correlation-aided iterative decoding for magnetic recording channels," *ECTI Trans. Elect. Eng., Electronics, and Commun.*, vol.5, no.2, pp. 1–12, Aug. 2007.

[95] I. B. Djordjevic and B. Vasic, "Noise-predictive BCJR for suppression of interchannel nonlinearities," *IEEE Photonics Technol. Lett.*, vol. 18, no. 12, pp. 1317–1319, Jun. 2006.

[96] Z. Wu, P. H. Siegel, J. K. Wolf, and H. N. Bertram, “Mean-adjusted pattern-dependent noise prediction for perpendicular recording channels with nonlinear transition shift,” *IEEE Trans. Magn.*, vol. 44, no. 11, pp. 3761–3764, Nov. 2008.

[97] J. B. Soriaga, P. H. Siegel, J. K. Wolf, and M. Marrow, “On achievable rates of multistage decoding on two-dimensional ISI channels,” in *Proc. IEEE Int. Symp. Information Theory*, Adelaide, Australia, 2005, pp. 1348–1352.

[98] M. Marrow and J. K. Wolf, “Detection of 2-dimensional signals in the presence of ISI and noise,” in *Int. Symp. Information Theory and its Applications*, Parma, Italy, 2004, pp. 891–894.

Chapter 4 - Possible Coding Enhancements for BPMR

[99] Y. Nakamura, Y. Bandai, Y. Okamoto, H. Osawa, H. Aoi, and H. Muraoka, “A study on nonbinary LDPC coding and iterative decoding system in BPM R/W channel,” *IEEE Trans. Magn.*, vol. 47, no. 10, pp. 3566–3569, Oct. 2011.

[100] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[101] K. Saowapa, H. Kaneko, and E. Fujiwara, “Systematic binary deletion/insertion error correcting codes capable of correcting random bit errors,” *IEICE Trans. Fundamentals*, vol. E83-A, no. 12, pp. 2699–2705, 2000.

[102] A. S. J. Helberg and H. C. Ferreira, “On multiple insertion/deletion correcting codes,” *IEEE Trans. Inf. Theory*, vol. 48, no. 1, pp. 305–308, Jan. 2002.

[103] T. G. Swart and H. C. Ferreira, “A note on double insertion/deletion correcting codes,” *IEEE Trans. Inf. Theory*, vol. 49, no. 1, pp. 269–273, Jan. 2003.

[104] M. C. Davey and D. J. C. MacKay, “Reliable communication over channels with insertions, deletions and substitutions,” *IEEE Trans. Inf. Theory*, vol. 47, pp. 687–698, Feb. 2001.

[105] J. Xiaopeng and M. A. Armand, "Interleaved LDPC codes, reduced-complexity inner decoder and an iterative decoder for the Davey-MacKay construction," in *Proc. IEEE Int. Symp. Information Theory*, St. Petersburg, Russia, 2011, pp. 742–746.

4.1 Low-Density Parity-Check Codes

[106] R. G. Gallager, “Low-density parity-check codes,” *IRE Trans. Inf. Theory*, pp. 21–28, Jan. 1962.

[107] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, Sep. 1981.

[108] J.-L. Kim, U. N. Peled, I. Perepelitsa, V. Pless, and S. Friedland, “Explicit construction of families of LDPC codes with no 4-cycles,” *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2378–2388, Oct. 2004.

4.2 Cyclic Linear Block Codes

[109] Q. Huang, Q. Diao, S. Lin, and K. Abdel-Ghaffar, “Cyclic and quasi-cyclic LDPC codes on row and column constrained parity-check matrices and their trapping sets,” unpublished.

4.3 Finite Geometry Codes as Cyclic LDPC Codes

[110] Y. Kou, S. Lin, and M. Fossorier, “Low density parity check codes based on finite geometries: a rediscovery and new results,” *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

4.3.2 Projective Geometry as a Realization of a Code

[111] H. Tang, J. Xu, S. Lin, and K. Abdel-Ghaffar, “Codes on finite geometries,” *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 572–596, Feb. 2005.

[112] N. Kamiya, “High-rate quasi-cyclic low-density parity-check codes derived from finite affine planes,” *IEEE Trans. Inf. Theory*, vol. 53, no. 4, pp. 1444–1459, Apr. 2007.

[113] S. J. Johnson and S. R. Weller, “High-rate LDPC codes from unital design,” in *Proc. IEEE Global Communications Conf., Exhibition & Industry Forum*, San Francisco, CA, 2003, pp. 150–152.

4.3.3 Progressive Edge Growth LDPC Codes

[114] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes", *IEEE Trans. Inf. Theory*, vol. 47, pp. 619–637, Feb. 2001.

[115] T. J. Richardson and R. L. Urbanke, “Efficient encoding of low-density parity check codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 638–656, Feb. 2001.

[116] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, “Regular and irregular progressive edge-growth Tanner graphs,” *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

4.4 Permutation Decoding

[117] F. J. MacWilliams, “Permutation decoding of systematic codes,” *Bell System Tech. J.*, vol. 43, no. 1, pp. 485–505, Jan. 1964.

[118] W. Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*. Cambridge, U. K.: Cambridge Univ. Press, 2003.

[119] J. D. Key. (2006, Feb. 6). *Recent developments in permutation decoding* [Online]. Available: http://www.ces.clemson.edu/~keyj

4.5.1 Auto-Diversity Decoding

[120] S. Ouzan and Y. Be’ery, “Moderate-density parity-check codes,” unpublished.

4.5.1.1 Remarks on AD Decoding

[121] T. Hehn, O. Milenkovic, S. Laendner, and J. B. Huber, “Permutation decoding and the stopping redundancy hierarchy of cyclic and extended cyclic codes,” *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5308–5331, Dec. 2008.

4.6.1 *Two-fold* EG Codes

[122] L. Zhang, Q. Huang, and S. Lin, “Iterative decoding of a class of cyclic codes,” in *Information Theory and Applications Workshop*, San Diego, CA, 2010, pp. 1–9.

Appendix A – List of Acronyms

2D	Two-Dimensional
3D	Three-Dimensional
ABS	Air Bearing Surface
AD	Auto-Diversity
APP	*a posteriori* Probability
AWGN	Additive White Gaussian Noise
BCJR	Bahl-Cocke-Jelinek-Raviv
BEC	Binary Erasure Channel
BER	Bit Error Rate
BP	Belief Propagation
BPMR	Bit-Patterned Magnetic Recording
CCW	Counterclockwise
CW	Clockwise
ECC	Error Correcting Code
EG	Euclidean Geometry
FER	Frame Error Rate
FFT	Fast Fourier Transform
FG	Finite Geometry
FGL	Field Generating Layer
FIR	Finite Impulse Response
FR	Full Rank
GF	Galois Field
GMR	Giant Magnetoresistive
GPR	Generalized Partial-Response
HAMR	Heat-Assisted Magnetic Recording
HDD	Hard Disk Drive
ISI	Intersymbol Interference
ITI	Intertrack Interference
LDPC	Low-Density Parity-Check
LLR	Log Likelihood Ratio
MAMR	Microwave-Assisted Magnetic Recording
MAP	Maximal *a posteriori*
ML	Maximum Likelihood
MLSD	Maximum Likelihood Sequence Detector
MMSE	Minimum Mean-Squared Error
MRC	Magnetic Recording Channel
MSE	Mean-Squared Error
MTR	Maximum Transition Run
NRZ	Non-Return-to-Zero
PDNP	Pattern-Dependent Noise Predictive
PEG	Progressive Edge-Growth

PG	Projective Geometry
PMF	Probability Mass Function
PMR	Perpendicular Magnetic Recording
PMRC	Perpendicular Magnetic Recording Channel
PR	Partial-Response
PRML	Partial-Response Maximum Likelihood
RLL	Run Length Limited
RS	Reed-Solomon
SER	Sector-Error Rate
SISO	Soft-Input Soft-Output
SNR	Signal-to-Noise Ratio
SOVA	Soft Output Viterbi Algorithm
SPA	Sum Product Algorithm
SUL	Soft Under Layer
SWR	Shingled Write Recording
TDMR	Two-Dimensional Magnetic Recording
VA	Viterbi Algorithm