Channels Engineering in Magnetic Recording: From Theory to Practice

Shayan Srinivasa Garani, *Senior Member, IEEE,* and Bane Vasić, *Fellow, IEEE*

***Abstract*—Information theory, coding theory and signal processing have significantly shaped magnetic read/write channels engineering through a chronological sequence of innovations and research advancements, cognizant of the underlying physical processes. In this magazine article, we provide an overview of magnetic recording technologies leading to the channels engineering aspects, central to this article. We survey important coding and signal processing algorithms along with some design architectures that have made it to practice within hard disk drives (HDDs) for magnetic recording channels. The push towards realizing ultra-high densities ($> 4\mathrm{Tb/in}^2$) on magnetic disks with ultra-high throughput rates ($> 10\mathrm{Gb/s}$) necessitates the development of native two-dimensional (2D) coding and signal processing algorithms and architectures within the framework of two-dimensional magnetic recording (TDMR) along with read head engineering and changes to the recording physics. We also provide an overview of novel channels engineering solutions covering all aspects of TDMR channels driven from a systems science perspective. The innovations and research advances described in this article may serve a broad engineering audience in other areas as well.**

***Index Terms*—magnetic recording technologies, read channels engineering.**

I. Introduction

THE advent of the computer age in the first half of the 20th century has propelled data storage technologies to advance in an unparalleled manner, leading to the birth, sustenance and exponential growth of technologies for both on-chip and off-chip memories. Examples of on-chip memories based on magnetics include ferroelectric random-access memories (FeRAMs), magnetoresistive random-access memories (MRAMs) etc. Hard disk drives (HDDs) are a classic example of an off-chip physical memory. The term data storage system refers to the entire system that interfaces to this off-chip physical memory, i.e., from the operating system, through the disk controller to the physical head/media subsystem.

According to International Data Corporation (IDC), it is expected that there will be more than 55 billion internet of things (IoT)-connected devices by 2025 [1]. Further, with cloud-based systems driven by datacenters and artificial intelligence (AI)-driven computing systems, several tens of billions of devices will be connected through this data-driven network. The 21st century will be the age of data storage technologies, mirroring how central processing unit (CPU)-based technologies from computer manufacturers propelled in the early part of the 20th century.

Since it is expected that nearly 175 zetta-bytes (ZB) of data will be generated every day by 2025 [2], we need state-of-the-art technologies to cater to this massive need. Along with memories with higher storage capacities, it is expected that these devices come with high throughput, shrinking form factors and low power consumption. Thus, speed, area, power and capacities ought to be within a ballpark for practical applicability of these devices. Though solid state devices (SSDs) are now part of most computer systems within desktops and enterprise solutions, catering to better read/write access times and comparable HDD storage capacities etc., one cannot ignore the role of HDDs in the data storage space since they have the cost per bit advantage [3]. Hybrid memory systems for handling cold and hot data will inevitably need HDDs. Both HDDs and SSDs are part of today's datastorage systems within datacenters.

Today, we have a thriving multibillion-dollar magnetic storage industry, which is ubiquitous and pushing itself continuously, advancing towards reaching the physical limits of magnetic storage [4]. Sustained research and development in several key multidisciplinary areas, such as magnetic materials, recording physics, tribology, channels, and systems engineering, to name a few, have enabled magnetic recording systems to reach capacities beyond 22 TB over multiple platters, using multiple read heads. HDD technology has moved from a 1D paradigm based on longitudinal and perpendicular recording to shingled recording with improvements to head/media design and recording physics. We are witnessing how with shingled magnetic recording (SMR) technology, using two readers, one could achieve storage densities beyond $1\mathrm{Tb/in}^2$ [5]. TDMR with shingled writing and 2D signal processing and coding can offer significant gains in areal densities [6] along with the necessary read head engineering and recording physics to work with TDMR.

In this magazine article, we will begin with the history of magnetic storage technologies that dates back to more than 140 years and describe how several innovations from coding theory, information theory and signal processing are continuing to shape its existence to remain competitive within the storage world.

S. S. Garani is with the Department of Electronics Systems Engineering, Indian Institute of Science, Bengaluru, KA, 560012 India e-mail: shayangs@iisc.ac.in.

B. Vasić is with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, .

Manuscript received March 19, 2023; revised Aug. 26, 2023.

A. History of Magnetic Recording: From a Technology Perspective

Magnetic recording research has witnessed a sustained development of more than 144 years from the early works of a magnetic recording apparatus conceived by Oberlin Smith in 1878 [7]. In his concept, a coil of wire could magnetize a medium, such as steel. By inducing a voltage, the same coil could be used for playback of the recorded pattern on the medium. This simple idea was demonstrated in practice by Valdemar Poulsen, and was eventually made commercially viable in the 1920s by Kurt Stille. Early advancements with electronic amplification, alternating current (AC)-bias recording, etc. helped in reducing the noise and distortion from the recorded signals during playback. Way back in 1928, Pfleumer developed a new recording medium using thin layers of metal powder. Later, magnetite was developed for coating paper. These simple ideas and further improvements to the recording media led to the birth of the magnetic tape that fuelled the entertainment industry. With subsequent media improvements, HDDs were conceived to realize random access that ensured a switch from the conventional sequential batch data processing paradigm. For a comprehensive historical overview of this celebrated technology, the reader is referred to [7].

The International Business Machine (IBM) 350 disk drive was the first commercial HDD with a capacity of 5 MB, weighing 500 lbs and having 50 24’ disks. The disks were vertically stacked and rotated at 1200 rpm. A constant gap separation with fluid bearing of 800 $\mu\mathrm{in}$ was maintained between the flying read head and the media to sense the data from the medium. The data format was also very simple, using non-return-to-zero, inverted (NRZI) codes with amplitude detection. The timing was ensured by having odd parities augmented with the NRZI bit stream, ensuring the synchronization of the open-loop oscillators. There was no sophisticated signal processing or error control codes within those systems. Several technical advancements led to the first removable HDD, the IBM 1311.

Later versions of HDDs had self-clocking codes coupled with peak detection circuits for sensing the magnetic transitions for decoding the data. A sequence of further technological innovations in the head designs, such as ferrite heads, metal-in-gap (MIG) heads, thin film inductive heads, etc. propelled an exponential growth in areal densities $\sim 30\%$, compounded annually. The introduction of (d, k) runlength-limited (RLL) codes, such as the $(2, 7)$ RLL code by IBM in 1979 provided improved gains in areal densities (ADs) since these codes could handle intersymbol-interference (ISI) and synchronization issues directly via the encoded data stream. The recording codes for magnetic storage were suited to peak detection [8]. The most important advancement in head technology was to switch from inductive heads to magneto-resistive (MR) heads that provided higher signal strengths independent of the linear velocity, making it ideal for smaller disks that had lower linear velocities [1]. These MR heads were commercially made available in 1985 [9]. For more details on the theory behind magnetic recording, the reader is referred to [10].

Trying to lower the gap between the head and the recording medium, and having improved RLL codes within the peak detection scheme could only saturate the AD gains. To stay competitive, with the ever increasing demand for higher areal densities, one required alternative strategies[2]. The innovative ideas from Kobayashi and Tang [11] on partial response signaling conceived as early as 1975 were just ripe to be tested out for a new generation of channels. Instead of finding ISI baneful, the PRML scheme allowed controlled ISI that could be tackled using Viterbi detection, which was already well-known by then for decoding of convolutional codes. In 1990, the first partial response maximum likelihood (PRML) detector was introduced by IBM, replacing the peak detection circuits. It was advantageous using MR heads, and the partial response maximum-likelihood (PRML) scheme provided additive gains. Very large scale integration (VLSI) technology was already mature by then to build such read channel integrated circuits (ICs) on a single chip. With low-cost and high-yielding chips that could be integrated within the disk controller systems, the PRML channel was a success along with MR heads. The use of analog and digital equalization techniques, PRML detector, RLL codes, and error correcting codes (ECCs) had HDDs geared up towards realizing $1\mathrm{Gb/in}^2$ areal densities. In fact, during the early 1990s, only a few IBM drives were PRML-based. The IBM 0681 drive with PRML technology achieved a capacity of $\sim 900\mathrm{MB}$. By early 1990s, the compounded annual growth rate in the ADs were $\sim 60\%$ significantly higher than the 30% growth rate seen for the past 3 decades before 1990s. This further increased with the introduction of giant magneto-resistive heads (GMRs) [12].

With the PRML in place, there were several notable innovations in the channels front, such as the noise-predictive maximum likelihood (NPML) [13] engine that used whitening filters to further improve the performance of Viterbi detectors by overcoming the media noise due to transitions from the magnetic domains. These ideas coupled with the design of generalized partial response (GPR) equalization, i.e., a pre-target selection and adaptive equalization techniques, helped realize ADs beyond tens of $\mathrm{Gb/in}^2$. To achieve higher ADs, one has to reduce the magnetic grain sizes for recording a bit. Reducing the grain sizes cannot continue unabated since recorded bits can be erased due to the *superparamagnetic* effect [14]. Mindful of these physical constraints, recording physics, head/media engineering and channels engineering had to be conceived to advance the state-of-the-art. We will discuss these technological aspects later in this section. Since a lot

[1] The reader must note that the throughput is directly linked to the linear velocities. To sustain a constant throughput, the spinning of the disk at the inner diameter must be higher than at the outer diameter, irrespective of the constant angular velocity on the solid disk.

[2] The reader must note that there has been a constant push for innovations in the HDD industry towards achieving higher ADs. These are driven by innovations in all the three major subsystems, namely (a) improved head design, (b) improved media fabrication, and (c) improved channels engineering. Given a head/media combination, efforts in the channels engineering side are pushed towards achieving higher AD gain. When gains from innovations in channels engineering tend to saturate, the head and media are pushed for improved designs, leading to next-generation channels development suited for those head/media designs. This constant push for innovations across all the three major subsystems within the HDD technology has helped constantly improve AD gains and stay competitive in the market.

of channels engineering depends on the underlying recording physics, we will next discuss the physics behind longitudinal and perpendicular recording schemes.

B. Physics behind Magnetic Recording Channels

Hysteresis is the key physical property for magnetic memories to hold information. Coercivity of the medium determines its stability against external magnetic fields and thermal issues. Put in simple words, when the magnetic grain sizes are small, they do not hold enough magnetic energy per unit volume, leading to poor signal quality during readback. From a physical perspective, an upper bound for the data storage density is determined by the magnetic *quadrilemma* [4], which we shall describe below:

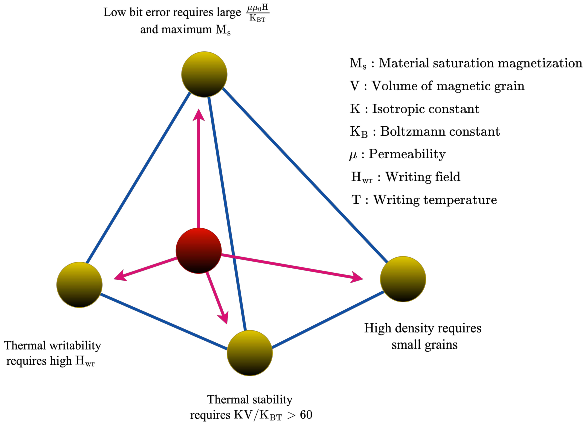

Low bit error requires large $\frac{\mu\mu_0 \mathrm{H}}{\mathrm{K_{BT}}}$
and maximum $\mathrm{M_s}$
$\mathrm{M_s}$: Material saturation magnetization
V : Volume of magnetic grain
K : Isotropic constant
$\mathrm{K_B}$: Boltzmann constant
μ : Permeability
$\mathrm{H_{wr}}$: Writing field
T : Writing temperature
Thermal writability requires high $\mathrm{H_{wr}}$
High density requires small grains
Thermal stability requires $\mathrm{KV/K_{BT}} > 60$

Figure 1. High ADs require smaller grain sizes. Thermal instability leads to grains switching their magnetization since the magnetic energy per unit volume is less. Further, write instability issues require high writing fields to get the desired level of anisotropy. Providing high writing fields over small grain sizes under high temperatures is not easy to realize. These conflicting requirements lead to the magnetic quadrilemma that dictates the achievable AD from a physics perspective. Adapted from [4].

Consider the extreme case of storing a bit over a magnetic grain, which is the smallest magnetic domain. Figure 1 shows the various conflicting physical requirements for storing data on a magnetic grain. To obtain high storage densities, the grain size must be small. Having smaller grains leads to thermal instability since the magnetic energy per unit volume is less and the magnetization of the grain is lost over time, governed by the Neel relaxation time. Now, there are write instability issues since the external write field must be very high to induce the required anisotropy over the grain. Finally, there are write errors since it is difficult to ensure write heads can provide this field over small grain sizes and high temperatures. This leads to the issue of thermal writeability since the magnetic moment in the medium must be sufficiently large or the writing temperature is not too high. These conflicting requirements will lead to optimization of the physical parameters to assess upper bounds on the achievable density within the classical regime.

Composite media with exchange coupling using soft and hard magnetic layers [15] is used to overcome limitations of writability and thermal stability with conventional write heads. With materials such as CoCrPt-alloys for hard magnetic layers and a combination of multilayer media structures, the medium has high anisotropy, allowing room for smaller grains. Based on the thermal stability criterion, the areal density of the medium depends on the grain volume and the areal packing fraction of the storage islands. Also, from a thermal-writeability perspective, based on the physical parameters, such as the writing field, writing temperature, Curie temperature of the medium, etc. one can estimate the areal densities in order to sustain the magnetization over a grain without flipping the state under heat-assisted writing (ref. to equation (6) in [4]). It is estimated that the achievable AD for magnetic storage under heat-assisted recording is around $20\mathrm{Tb/in}^2$ [4]. The reader must note that this assumption is only over one layer of the medium. By stacking layers of magnetic medium, with appropriate recording physics, one can achieve higher ADs.

At this stage, we have not discussed anything related to the noise modeling or statistical description of the read/write processes. Mapping the physical constraints to a communication-theoretic framework will eventually lead to achievable bounds on data storage densities from an information-theoretic view point.

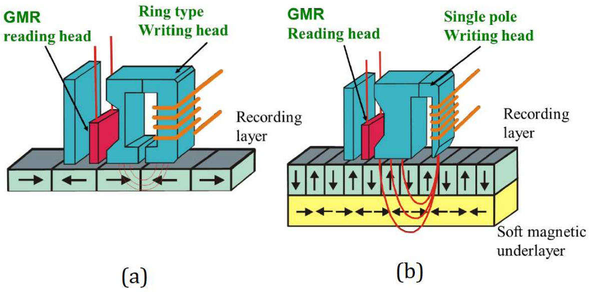


Figure 2. (a) Longitudinal magnetic recording, where the magnets are aligned in the east-west direction. (b) Perpendicular magnetic recording with a soft under layer. The magnets are aligned in the north-south direction. Figure source: Courtesy [16].

1) Longitudinal Magnetic Recording (LMR): In longitudinal magnetic recording (LMR), the magnetic anisotropy is oriented in the thin film plane i.e., in the east-west direction. Figure 2(a) shows the alignment of the magnetic grains. Around $140\mathrm{Gb/in}^2$ ADs were demonstrated in laboratory setups.

2) Perpendicular Magnetic Recording (PMR): The pioneering work of Iwasaki in 1975 [16] led to perpendicular magnetic recording (PMR), where the magnetic anisotropy is perpendicular to the thin film plane i.e., the magnetic domains are oriented in the north-south direction as shown in Figure 2(b). Present day HDDs are based on PMR technology, which provides significantly higher ADs than LMR up to $1\mathrm{Tb/in}^2$. In PMR, there is a soft magnetic underlayer (SUL) which provides a return flux path. Thus, there is a stronger write field gradient using the same head material as the LMR, allowing the medium coercivity to be higher. Larger write field gradient results in smaller transition jitter, thereby improving the signal-to-noise ratio (SNR).

The read head design is the same irrespective of the recording scheme.

C. New Technologies for Higher Areal Densities

Though PMR-based HDDs have been highly successful since 2005, there is a saturation to the AD gains around $1\mathrm{Tb}/\mathrm{in}^2$. This led towards research in alternative technologies for furthering AD gains. There have been three major promising technologies: (a) heat-assisted magnetic recording (HAMR) [17] [20], which is driven through recording physics, (b) bit-patterned media (BPM) [21], driven through media innovations, and (c) two-dimensional magnetic recording (TDMR) [22] with innovations from 2D signal processing and coding i.e., from the channels engineering side. Figure 3 shows different magnetic recording technologies, early versions of some of which are already into production.

In HAMR [20], the recording medium is heated above the Curie temperature for a nano timescale. This makes the medium lose its coercivity. Around the same time, the write head writes on the medium. When the medium cools down, it regains its coercivity, thereby retaining the bit written onto it. The key challenge in HAMR drives is nanoscopic guided heat delivery without burning up the write heads. This is accomplished using nano-scale plasmonic waveguides for directing the energy than a direct laser heating of the medium. Sputtering effects over the neighboring cells must be avoided. Figure 3(a) shows the schematic of HAMR. Laboratory HAMR prototypes developed by Seagate have showed areal densities as high as $2\mathrm{Tb}/\mathrm{in}^2$ [23].

There are also works on other energy-assisted magnetic recording (EAMR) schemes, such as microwave-assisted magnetic recording (MAMR) with an eye towards 3D-magnetic storage. In MAMR, the media is composed of multiple layers, similar to layered media in optical discs. Using high-frequency magnetic fields generated by spin torque oscillators, selective switching of grains at a certain depth in a layered medium is possible [24] [25] [26]. Data is recorded in overlapping layers using selective microwave resonant frequencies attuned to these layers (refer to Figure 3(b)). The success of MAMR depends on the strength of the magnetic field and the head-media spacing (gap length). It is envisaged that MAMR can provide ADs to $10\mathrm{Tb}/\mathrm{in}^2$. In HAMR, lower layers destroy the information in an upper layer. This is circumvented in MAMR. However, MAMR is also prone to similar challenges that HAMR faces.

In BPM, the medium is tessellated into magnetic islands in an orderly manner. Each magnetic island is well-separated with guard bands (shown in Figure 3(d)). Fabrication of such media requires careful lithography, which could be costly. In practice, one could have deleted or fused magnetic islands as part of fabrication defects, requiring significant efforts from channels engineering to work with such media. Write synchronization issues and other aspects make this technology a bit difficult for commercialization.

TDMR is a technology that works with random grains, borrowing from the PMR media technology, unlike BPMR with fixed island sizes. However, instead of writing bits on tracks in the usual 1D manner, tracks are shingled, and data is encoded in 2D. To ensure a sufficiently large magnetization field capable of magnetizing materials with high coercivity, the head is made larger than the track width. However, to achieve a higher areal bit density, the tracks should be very narrow. As a result of narrower tracks, each sweep of the head during writing partially overlaps with the previous track, i.e., writing is noisy.

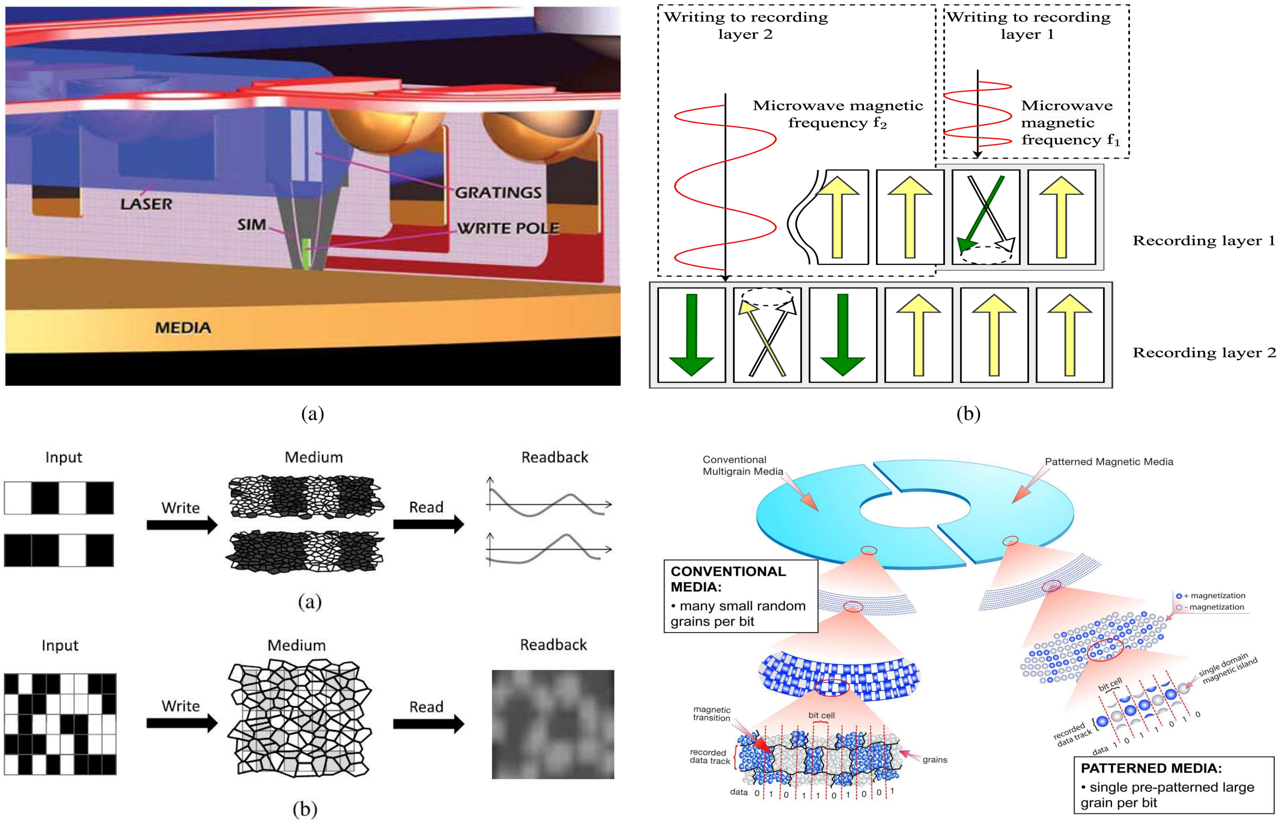
Unlike traditional recording, where data is organized in well-separated tracks (shown in Figure 3(c).a), in TDMR systems, the data bits are arranged in a 2D array (see Figure 3(c).b). In traditional systems the intersymbol-interference (ISI) is small and along the downtrack, which can be controlled by a sequence detector. In a TDMR system, since the head picks up magnetization from adjacent tracks, there is severe ISI both along a track and across the tracks (inter-track interference (ITI) as a wider head reads data larger than the physical dimensions of a stored bit. Narrow read heads can be fabricated (minimizing ITI) with a penalty in the noise during reads.

2D offers many advantages over 1D. First, one can achieve significantly higher ADs by packing more bits per unit area by using clever 2D coding and signal processing techniques[3]. Next, it is possible to have a relaxed 2D synchronization since the read head is not confined to a narrow single track to handle timing artifacts. Timing issues in the down-track and cross-track directions can be handled as a whole through multi-channel processing. Last, the throughput is significantly higher since an entire array of bits can be read and processed. However, all these advantages towards getting higher ADs come at the cost of designing the array of read heads and processing the signals. It is expected that processing these array of signals requires sophisticated signal processing that could be power and area intensive when implemented on a read channel chip. With advancements in low-power VLSI technologies, both at the device and circuit architectural levels, it is possible to overcome these challenges in pursuit of realizing the promise of TDMR gains. We will describe the channels engineering aspects for TDMR in subsequent sections. The reader must note that TDMR provides additive AD gains over both HAMR/MAMR and BPMR technologies. Thus, it is important to develop native 2D coding and signal processing solutions for TDMR channels for increasing the existing AD limits of HDDs.

D. Organization of the article

With this broad technological background in mind, we describe the organization of this article. In Section II, we provide an overview of the channel modeling aspects for various magnetic recording technologies. This will help us

[3] The reader must recall that, unlike 1D traditional recording, we have 2D ISI/crosstalk in the down-track and cross-track directions.


LASER
SIM
GRATINGS
WRITE POLE
MEDIA
Writing to recording layer 2
Writing to recording layer 1
Microwave magnetic frequency f_2
Microwave magnetic frequency f_1
Recording layer 1
Recording layer 2
Input
Medium
Readback
Write
Read
Conventional Multigrain Media
Patterned Magnetic Media
CONVENTIONAL MEDIA:
• many small random grains per bit
+ magnetization
- magnetization
single domain magnetic island
bit cell
recorded data track
data
magnetic transition
grains
PATTERNED MEDIA:
• single pre-patterned large grain per bit

(a) (b)

(a)

(b)

(c) (d)

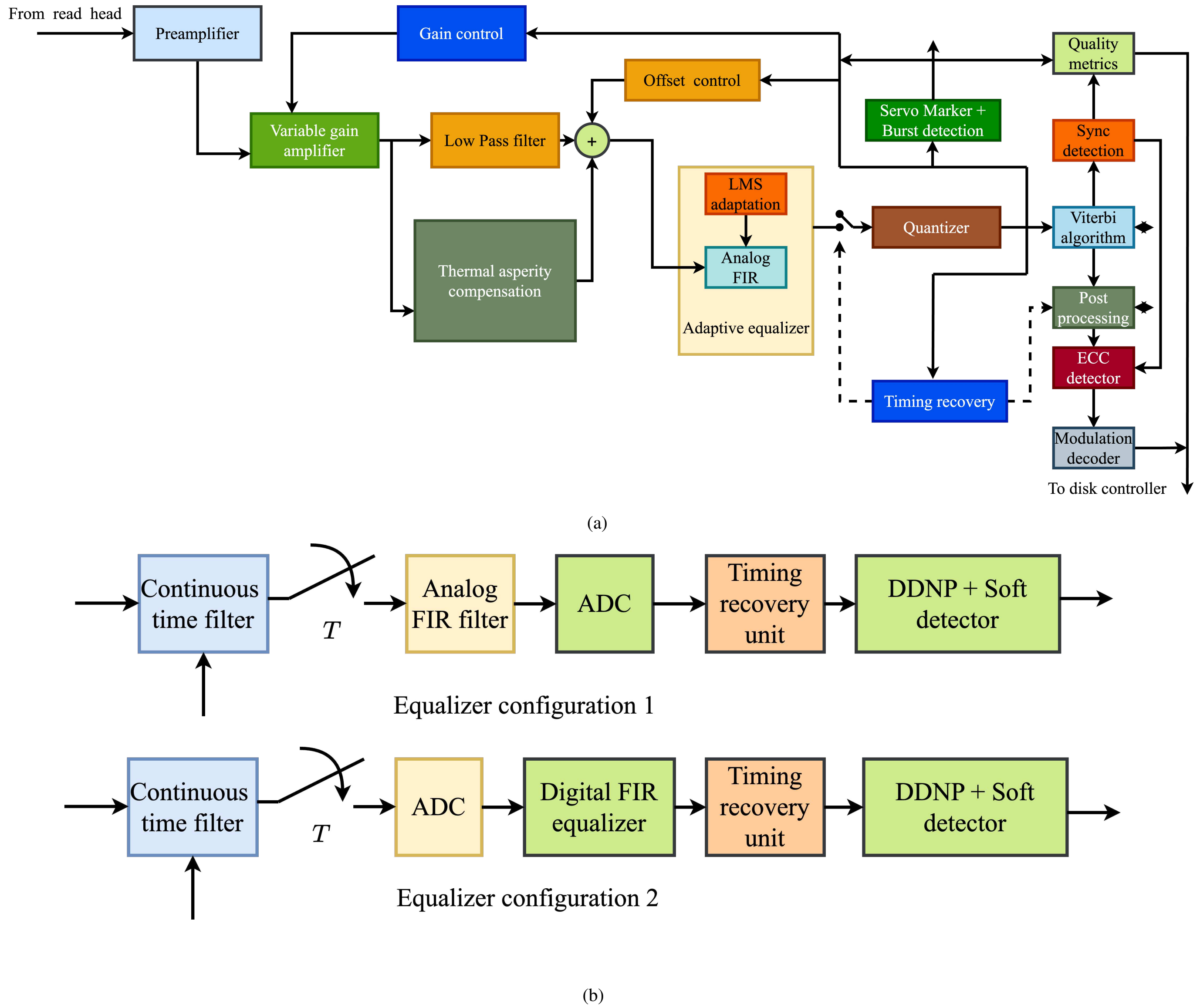
Figure 3. (a) Schematic of a HAMR setup with guided LASER beams impinging on the medium. Courtesy: [17]. (b) Schematic of a MAMR setup. Using spin torque oscillators with microwave resonant frequencies tuned to each layer, selective switching of grains in a layered media is possible (Adapted from [18]). (c) Schematic of a TDMR writing and reading process. a) In conventional setup writing and reading is on a single track. b) In TDMR, the data is organized as a 2D array. The 2D readback signal is due to sensing multiple tracks. Severe crosstalk in the readback signal has much of the information required for decoding the original information. Adapted from [6]. (d) Illustration of a BPMR pattern. The medium is fabricated carefully with equally-spaced islands, i.e., ordered grains. Courtesy: Source [19].

connect the physics to a communication-theoretic setup, useful towards building algorithms in practice. In Section III, we survey information-theoretic tools for assessing the mutual information rates (MIR) of MR channels. These results are important to get a feel for achievable ADs under noisy conditions. In Section IV, we discuss the signal processing innovations that had to be conceived with every generation of the read channel. We will provide a summary of various signal processing techniques such as analog pre-filtering, equalization, timing recovery, signal detection, etc. applicable to 1D and 2D channels. In Section V, we discuss the important coding techniques that were practically applicable to HDDs. Specifically, we will highlight the role of algebraic codes and iterative codes that were implemented in practice, within HDDs. We will also discuss modulation codes used in early versions of practical HDDs. We will also highlight the circuit architectural aspects towards realizing the coding algorithms in practice by explaining the importance of algorithmic/system-level tweaks that have had a significant impact in practice. In Section VI, we conclude the article along with our perspectives on the channels engineering aspects for next-generation HDDs.

II. Modeling for Magnetic Recording Channels

From now onwards, we will focus on the channels engineering aspects of HDDs. Figures 4(a) and 4(b) (Adapted from [27], Chapter 15) show the block diagram schematic of a read channel architecture. Information bits are encoded through an error correction coded and then modulated to satisfy the constraints of the channels. These bits are then written on to the medium through the write head after converting the current pulses to flux changes on the medium. After sensing the readback waveform, the signal is passed through a band-limited filter and an adaptive finite impulse response (FIR) equalizer to maintain a partial response (PR) target. Post sampling, the samples are driven by a timing recovery loop and signal detection. The detected bits are then decoded by the modulation code and through the ECC decoder before retrieving the information bits. The post-processor block in Figure 4(a) indicates a turbo loop for soft-decision decoding within the PRML setup. Thus, coding and signal processing algorithms are part of read channel ICs that interface with magnetic disk drives.

The act of storing and retrieving information reliably from a storage device is an instance of a noisy communications channel. With appropriate channel modeling, one can abstract the physical processes within a communication-theoretic framework so that tools from information theory, coding theory and signal processing can be applied towards channels development. A first step towards this effort is signal modeling.



(a)

(b)

Figure 4. (a) Schematic of a conventional read channel architecture. The signal from the read head goes through a preamp circuit, followed by a continuous-time filter to remove any out of band noise and an analog-to-digital converter (ADC). The signal chain includes an analog FIR equalizer based on the least mean squares (LMS) adaptation engine, adaptive gain and timing loops, and the Viterbi detector with post-processing towards error recovery. (Adapted from [27], Chapter 15). (b) Various configurations of analog and digital FIR equalizers within a PRML detector setup with noise whitening capability, such as using data-dependent noise prediction (DDNP) filters.

A. Signal Modeling for LMR and PMR

The playback signal strength depends on the physical parameters specific to a recording scheme.

By considering the linear superposition of pulse amplitudes of isolated transition magnetic responses that depend on the written bits, we can abstract the signal model for the dibit response $p(t)$ for longitudinal recording over a pair of transitions separated by interval T as

$$y(t) = \sum_k b_k \underbrace{[h(t-kT) - h(t-(k-1)T)]}_{=p(t)} + n(t), \qquad (1)$$

where $h(t)$ is the Lorentzian model response given by

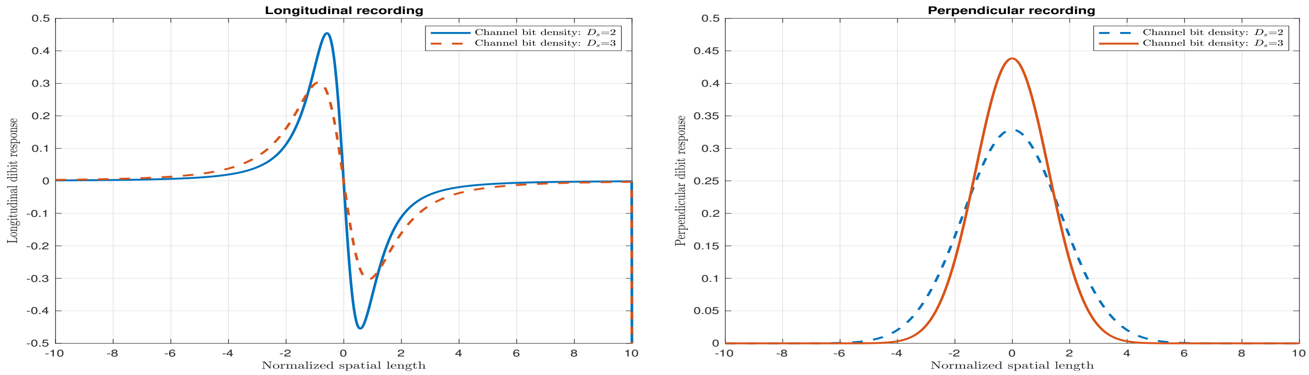
$$h(t) = \sqrt{\frac{4E_t}{\pi w}} \frac{1}{1+\left(\frac{2t}{w}\right)^2}, \qquad (2)$$

with w being the pulse width at half-maximum. The peak amplitude of the pulse depends on the properties of the magnetic medium such as the remanent magnetization, air gap length, width of the free layer etc. For more details, the interested reader can refer to [27] (see equation (2.31), Chapter 2 in [27].). In equation (2) $E_t = \int |h(t)|^2 dt$ represents the normalized energy of the isolated transition response. The noise term $n(t)$ in (1) represents both the media and electronics noise sources.

Similarly, the transition response for the PMR channel is simplified as

$$h(t, w) = V_p \left(\frac{2\sqrt{\ln(2)}}{w} t \right), \qquad (3)$$

where V_p is the peak value of the isolated transition response. The pulse width w at half-maximum for PMR depends on the



(a) (b)

Figure 5. LMR and PMR transition responses.

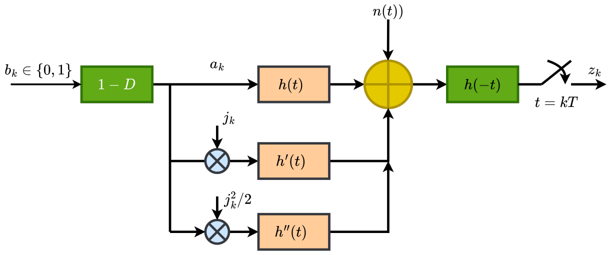


Figure 6. Continuous-time signal model with jitter contributions upto a second order. Adapted from [28].

air gap length, transition width, GMR free layer thickness, etc. To normalize the energy under the impulse response to unity, the peak value V_p is given by

$$V_p = \frac{1}{2}\left(\frac{\pi}{2w^2\ln(2)}\right)^{\frac{1}{4}}. \tag{4}$$

Figure 5 shows the sketch of the transition response for both longitudinal and perpendicular recording schemes. Noise modeling from recorded waveforms is important for deriving channel models. Media noise, which is usually colored, arises due to variations in the media magnetization across the grains. The electronics noise, which is usually white, arises due to sensing and preamplifier circuits. Media noise is a consequence of stationary dc remanent noise due to random in-plane anisotropy dispersion (Chapter 2, [27]), noise due to transitions and modulation noise due to surface roughness. The dc remanent noise is absent is perpendicular recording due to loop squareness and the noise due to transitions is the only noise since magnetic grains can fall across bit cell boundaries.

A simple way of obtaining the noise model towards simulations is to have noise perturbations in the position t and width w parameters within the transition response $h(t,w)$ as done in [28]. At a position k,

$$h_k(t,w) = h(t - kT + j_k, w + w_k), \tag{5}$$

where j_k and w_k are random variables and assumed to be Gaussian distributed with zero mean and variances σ_j^2 and σ_w^2, respectively.

By doing a Taylor series approximation of equation (5) using equations (2) and (3), we get the equivalent model described in Figure 6.

With this in place, we are set to define the SNR. The SNR for recording channels is given by

$$\text{SNR} = \frac{E_t}{N+M}, \tag{6}$$

where N is the electronics noise variance and M is the media noise variance. The media noise variance is given by

$$M = 2\sigma_j^2 I_j + 2\sigma_w^2 I_w, \tag{7}$$

where $I_j = \int_0^\infty \left(\frac{\partial h}{\partial t}\right)^2 dt$ and $I_w = \int_0^\infty \left(\frac{\partial h}{\partial w}\right)^2 dw$.

The linear density or channel bit density (cbd) is given by

$$D_s = \frac{PW_{50}}{T}. \tag{8}$$

Since the data is coded at a rate R, the user bit density (ubd) is given by

$$\text{ubd} = \text{cbd} \times R. \tag{9}$$

In a practical setup, the systems are oversampled with a factor O_s. If the SNR is S dB and the energy in the transition response is E_i, then, for a given cbd and $\frac{M}{N+M} = f$, the electronic noise variance is given by

$$\sigma_e^2 = (1-f)\frac{E_i \times \text{cbd}}{2} O_s 10^{-0.1S}. \tag{10}$$

Equation (10) is helpful for obtaining the required noise parameters for generating the waveforms towards simulations.

With these details in place, we are now ready towards a simulation model for assessing the performance of the signal chain for LMR and PMR channels. Let us now discuss signal models for advanced channels.

B. Signal Models for Advanced Channels: A TDMR Case Study

TDMR achieves high AD gains by reducing the bit size in both the directions within the bounds dictated by the magnetic quadrilemma. With bit-sizes to the order of a magnetic grain size, irregularities in the medium defined by the position and geometry of the grains will influence the read and write processes.

Though there are many simple models for TDMR such as the binary error and erasure model and the discrete grain model [29] [30], we will focus specifically on the Voronoi-based model and discuss the details of this channel model (media model, read and write procedures, and noise characteristics) in depth since it is the accepted model for read channel simulations in the industry, and is used in [31] [32]. Figure 7 shows the Voronoi tiling model for TDMR. Each Voronoi region on the magnetic medium represents a grain. The distribution of Voronoi centers is modeled using a point process, typically a Poisson random process. The magnetic domains are formed using the Voronoi regions whose centers are the grain centers. Physics-based micromagnetic models simulate the sizes, shapes, and distribution of the grains close to an actual magnetic recording medium.

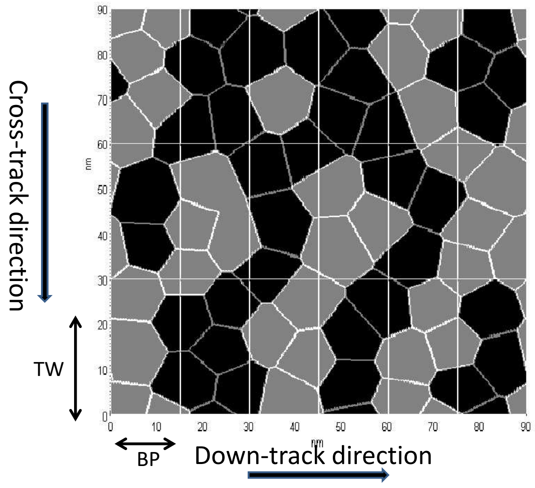


Figure 7. An example of the Voronoi channel model. The bit cell width along the track is BP=15 nm. In the cross-track direction, the track width is TW=30 nm. The center-to-center (CTC) spacing is 10 nm. Adapted from [6].

The micromagnetic recording model [33]–[35] assumes a granular thin film medium in which grains are uniformly magnetized. This model makes no prior assumptions of a grain shape or location. The magnetostatic and exchange interactions between nearest neighbors are calculated with the knowledge of the grain shape, and the magnetostatic interactions between more distant pairs of grains are computed hierarchically. The time evolution of the magnetization is computed by integrating the Landau-Lifshitz-Gilbert (LLG) equation in spherical polar coordinates using a Krylov ordinary differential equation (ODE) solver [36]. Head-field distributions are precomputed for some direct currents (dc), and the recording sequence is defined by the velocity of the head over the medium and a head current waveform represented by the random bit sequence to be recorded. The field at each point of interest in the medium is then computed by spatially interpolating the head-field distribution. TDMR platform simulations were also done in [37] using the micromagnetic model. Also, the micromagnetic model has been used to predict AD limits in perpendicular recording [38]. There are other intermediate models based on quasi-micromagnetic simulations [39]. However, the high complexity of micromagnetic model makes it difficult to run simulations for the performance evaluation of coding and signal processing algorithms [39], [40].

We shall now discuss more details about the Voronoi model.

1) Recording Media Model: The Voronoi tiles are used to simulate the irregularities of magnetic grains. The distribution of grains on the medium can be modeled using a Poisson-disk process with boundary sampling, as proposed in [41]. Each new grain is randomly generated such that it touches at least one of the existing grains to achieve a close random packing under the CTC constraint. Before a new grain is generated, the boundary that is at a CTC distance from the existing grain centers is identified. The position of a new grain is randomly generated with a uniform probability density on the identified boundary.

According to the Voronoi model, the storage medium comprises tiling of the shifted grain-centers with each region representing the grains. With a rectangular grid over bit cells, each cell of size $\mathrm{BL} \times \mathrm{TW}$ represents the size of the bit cell in the downtrack and crosstrack directions. The bit aspect ratio (BAR) which is defined as $\mathrm{BAR} = \frac{\mathrm{TW}}{\mathrm{BL}}$ governs the minimum resolution of magnetization. The act of writing and reading an information bit from a bit cell, i.e., from a rectangular cell with a given bit length and bit aspect ratio, constitutes an instance of a noisy communication process, i.e., a *noisy channel*. The bit cell area is equivalent to the channel bandwidth. The channel bit density is given by $\frac{1}{\mathrm{TW} \times \mathrm{BL}}$ bits/unit-area.

2) Write Process: During the writing process, the read head writes the bipolar coded symbol $x_{i,j}$ by changing the magnetic polarity of all grains whose centers lie within the $(i,j)^{\mathrm{th}}$ bit cell according to the value of bit $x_{i,j}$. Magnetic domains are formed by the continuous regions of Voronoi cells with the same polarity of magnetization. The channel input signal $x(t_1,t_2)$ is given by

$$x(t_1,t_2) = \sum_{i,j} x_{i,j} \Pi_{TW}(t_1 - i \times TW) \Pi_{BL}(t_2 - j \times BL), \tag{11}$$

where $x_{i,j} \in \{-1,+1\}$ is the symbol which will be written on the $(i,j)^{\mathrm{th}}$ bit cell and

$$\Pi_T(t) = \begin{cases} 1 & \text{if } 0 \le t < T, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

The indices t_1 and t_2 refer to the spatial coordinates on the magnetic disk.

3) Read Process: The readback signal depends on the grain magnetization and read head geometry [42] along with crosstalk received from neighboring cells that depends on the grain distribution. Suppose that the read head picks up magnetization only from $m \times n$ neighboring cells. As a result, the read head output sample $y_{i,j}$ at the center of the $(i,j)^{\mathrm{th}}$ cell depends only on the polarity of the grains in the $m \times n$ neighborhood around the $(i,j)^{\mathrm{th}}$ cell, denoted as $C_{i,j}$. We use the 2D Gaussian pulse model for the read head sensitivity function. The 2D Gaussian pulse is characterized by the pulse widths $\mathrm{PW50}_x$ and $\mathrm{PW50}_y$ at half-amplitude in the down-track and cross-track directions.

$$h(x,y)=\frac{\ln 2}{\pi \mathrm{PW50}_x \mathrm{PW50}_y}\exp\left(-\frac{(\ln 2)x^2}{\mathrm{PW50}_x^2}-\frac{(\ln 2)y^2}{\mathrm{PW50}_y^2}\right), \tag{13}$$

with $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x,y)dxdy=1$ for normalization.

The read head sensitivity function is the contribution of each grain towards the generation of a readback signal. Figure 8 shows the distribution of grains on the medium, magnetization of the Voronoi regions on the medium and the continuous time readback signal without electronic noise.

Let $h_{i,j}[p,q]$ be the discrete-time read response of the bit at position (i,j). The indices p,q are integers representing samples on the 2D media after sampling. These response coefficients are random and dependent on the position and shape of grains within the bit area. We can compute the average bit-response as

$$h(p,q)=\mathbf{E}_{I,J}\left[h_{i,j}[p,q]\right], \tag{14}$$

where I and J are random variables indicating the distribution of the positions of grain centers in the down-track and cross-track directions, respectively.

The readback signal sample without considering the electronic noise is given by

$$y_{i,j}=\sum_p\sum_q x_{i-p,j-q}h_{i-p,j-q}[p,q], \tag{15}$$

where $x_{i,j}$ is the symbol written on the $(i,j)^{\mathrm{th}}$ bit-cell. Furthermore, the ideal read head output, $s_{i,j}$, is obtained by considering the average discrete-time output of the $(i,j)^{\mathrm{th}}$ bit area as

$$s_{i,j}=\sum_p\sum_q x_{i-p,j-q}h[p,q]. \tag{16}$$

The mean squared value of the read-back signal is computed as $V_p^2=\sum_p\sum_q|h[p,q]|^2$. The media noise comes from the random perturbations of $h_{i,j}[p,q]$ around the average response $h[p,q]$. Therefore, the energy of media noise σ_m^2 is computed as

$$\sigma_m^2=\mathbf{E}_{I,J}\left[\sum_p\sum_q|h_{i,j}[p,q]-h[p,q]|^2\right]. \tag{17}$$

For TDMR channels, we have three different SNR notions:

$$\mathrm{SNR}=10\log_{10}\left(\frac{V_p^2}{\sigma_m^2+\sigma_e^2}\right), \tag{18}$$

$$\mathrm{SNR}_{\mathrm{Media}}=10\log_{10}\left(\frac{V_p^2}{\sigma_m^2}\right), \tag{19}$$

$$\mathrm{SNR}_{\mathrm{Elec}}=10\log_{10}\left(\frac{V_p^2}{\sigma_e^2}\right), \tag{20}$$

SNR refers to the overall SNR, while $\mathrm{SNR}_{\mathrm{Media}}$ and $\mathrm{SNR}_{\mathrm{Elec}}$ are the SNRs corresponding to the media and electronic noise sources, respectively.

The noise distribution for different input patterns can be obtained using the Voronoi model to study the impact of how neighboring bit transitions lead to increased noise. Study of the noise distribution is helpful in computing the symmetric information rates and optimization of the TDMR system parameters under various channel conditions [43] [44]. The most harmful patterns are those that have consecutive transitions along the on-track and off-track directions, leading to degraded performance during signal detection [44] [45].

C. Signal Models for HAMR and BPMR

The read channel model for BPMR can be described by a 2D ISI channel [46]. The contribution of a magnetic island to the readback signal is determined by the integral of the head potential function over that island. The 2D ISI channel is parameterized by two parameters: (a) the down-track crosstalk parameter that is determined by the distance between the shields and the MR element, and (b) the cross-track crosstalk parameter determined by the width of the read head. By obtaining the 2D ISI channel in the form of a matrix, the discrete model for readback signal is obtained using the 2D filter and amplitude coded bits within a linear systems framework [47].

For the write process, since the head size can be more than the spacing between the magnetic islands, there could be write errors, with subsequent islands overwritten. The positioning of the head requires synchronization during writing without which one can have synchronization mismatch during writing due to write clock offsets. Further, due to fused islands [21], one can have insertions or deletions that need to be handled through special codes. All these contribute to *written-in* errors, which must be overcome through proper coding during the write process [48].

In [49], the authors derived a channel model for HAMR using the thermal Williams-Comstock model [50]. The authors in [51], consider modeling the write and read portions using HAMR over a bit-patterned media. In that model, during writing a portion of the grains P can be flipped according to a flip probability [51] along the reversing field that depends on energy barrier required to flip the magnetization w.r.t the switching field and the temperature.

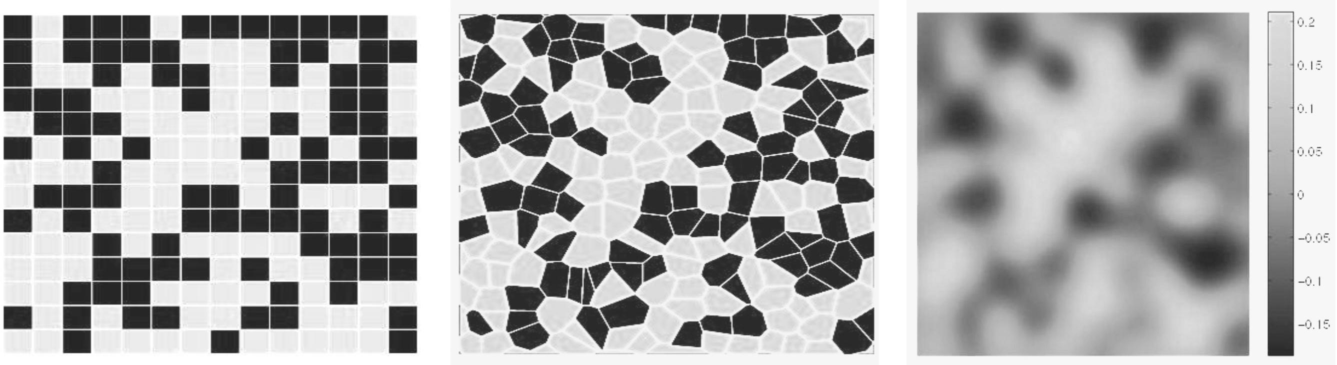
$$\frac{dP}{dt}=\alpha(1-P)\exp\left(-\frac{\Delta_\epsilon}{k_BT}\right), \tag{21}$$

where α is a constant. The energy barrier Δ_ϵ depends on the magnetic energy $M(T)$ per unit volume that in turn depends on temperature, anisotropy constant $K(T)$, applied field H and magnetization as a function of temperature T given by [51]

$$\Delta_\epsilon=(K(T)-M(T))VH. \tag{22}$$

III. Capacity Estimation for Magnetic Recording Channels from Information-Theoretic Tools

The capacity of a data storage system under noisy conditions is the upper limit on the number of bits per unit area that one can store on the magnetic medium with an arbitrarily small



(a) (b) (c)

Figure 8. Write/read model for the Voronoi medium: (a) Desired magnetization of an ideal medium. The grains with magnetization +1/-1 are shaded white and black in each of the squares. (b) Magnetization of a non-ideal medium as per the Voronoi model. (c) Readback signal (before sampling). We assume the readback impulse response is a truncated 2D Gaussian pulse of unit energy with half-maximum of 1 bit-period and a span of 3 bit-periods in both dimensions. Adapted from [6].

probability of decoding error. From Shannon's channel coding theorem [52], if we choose a long enough code length n, one can code the data bits at a rate $R < C$, where C is the channel capacity with an arbitrarily small probability of error $p_e^{(n)} < \epsilon_n$ and $\epsilon_n > 0$. The evaluation of the channel capacity C is difficult. In general, it is not possible to exactly obtain the capacity for magnetic recording channels since the channel is non-stationary as the error rates can vary across the tracks and with device aging. Also, the stationarity/ergodicity conditions in Shannon's original theorem do not hold in practice for magnetic recording channels. Thus, it is important to come up with reasonable estimates for computing the channel capacity under certain assumptions.

The capacity estimate of the recording channel can serve as a performance benchmark for designing error correction codes and for optimizing the physical parameters of data storage systems [53]–[55]. The reader must also note that a practical way of distilling the tightest coding rate for a practical magnetic recording channel is to increase the coding rate to such an extent that there are errors beyond a threshold[4]. Using bit error rate as the comparison criterion encapsulates the channel and the signal detector using a simple model, such as a binary symmetric channel, which provides a loose lower bound on the information rate. Based on physically abstracted channel models, one could directly compute the mutual information rate with different input distributions under different bit aspect ratios to obtain a lower bound on the achievable storage density as done in [55].

Chan et al. [56] use the grain-flipping probability model to optimize the areal density using a signal chain that includes the partial response equalizer, soft-output Viterbi detector, and low-density parity check decoder. By varying the physical parameters, such as the bit length, track pitch and code rate in the simulations, areal density is evaluated empirically for different head/media configurations. However, the detector is still 1D, which could be still limiting.

The capacity of discrete channels is defined as the maximum MIR over all discrete-input distributions. Various bounds on the capacity of certain 1D discrete input channels with ISI have been proposed [57]–[60]. The MIR or i.i.d. capacity can be computed with reasonable accuracy using Monte Carlo methods.

A. Computing Mutual Information Rates: Trellis-based Approach

Tne information rate for 1D additive white Gaussian noise (AWGN) channels with memory can be computed using the forward recursion of the sum-product (Bahl-Cocke-Jelinek-Raviv, BCJR) algorithm [61].

1) Capacity Estimate for the 1D Framework: Post partial response equalization with a pre-target and whitening of the noise, the equivalent magnetic recording channel is approximately a linear ISI channel. We shall describe how to estimate the MIR for an ISI channel. Consider the MIR for n-uses of the channel, taking inputs $\mathbf{x}^{(n)}$ and producing outputs $\mathbf{y}^{(n)}$ given by:

$$I(X;Y) = \lim_{n\to\infty} \frac{1}{n} I\left(\mathbf{x}^{(n)};\mathbf{y}^{(n)}\right). \tag{23}$$

Now, equation (23) can be computed as

$$I(X;Y) = h(Y) - \underbrace{h(Y|X)}_{=h(Z)}, \tag{24}$$

where $h(Y) = \lim_{n\to\infty} \frac{1}{n} h\left(y^{(n)}\right)$ and $h(Z) = \lim_{n\to\infty} \frac{1}{n} h\left(z^{(n)}\right)$.

After whitening through NPML [13], the noise statistics are approximately Gaussian distributed $\mathcal{N}(0,\sigma_x^2)$; hence, $h(Z) = \frac{1}{2}\log_2(2\pi e\sigma_z^2)$. We are now left with the computation of $h(Y)$.

This is where the ideas behind the BCJR algorithm [62] help us. For this, we need to bring in some parameters related to the trellis formalism. Suppose the ISI memory is N and the input alphabet size is $|\mathcal{X}|$. There are $|\mathcal{X}|^N$ states in the trellis.

[4] This is one of the practical ways engineers fine-tune the coding parameters after an initial estimate of the capacity since the stationary/ergodicity assumptions do not hold in a practical setup. Also, a wide range of coding rates are distilled depending on the device aging properties and various stress tests done in practice.

The forward probability of ending in a state j at time k can be computed as

$$\alpha_k(j) = P(y^k, S_k = j). \tag{25}$$

The term $\alpha_k(j)$ in can be obtained through the recursion

$$\alpha_k(j) = \sum_i \alpha_{k-1}(j)\gamma_k(i,j), \tag{26}$$

where $\gamma_k(i,j) = P(S_k = j, y_k|S_{k-1} = i)$ is the branch probability i.e., transitioning from state i at time instant $k-1$ to state j at time instant k.

The overall probability $P\left(y^{(k)}\right)$ can be obtained as

$$P\left(y^{(k)}\right) = \sum_j \alpha_k(j). \tag{27}$$

By initializing $\alpha_0(j)$ for all states j with the stationary distribution [5] of the inputs, one can efficiently compute $P\left(y^{(k)}\right)$ for a certain noise realization. By averaging the computation $p\left(y^{(n)}\right)$ over several runs, we can obtain

$$h(Y) = -\lim_{n\to\infty} \frac{1}{n}\mathbf{E}\left[\log_2\left(p\left(y^{(n)}\right)\right)\right]. \tag{28}$$

From equations (24)-(28), one can estimate the MIR for ISI channels post noise whitening. The reader must note that when the noise is not fully whitened, one must account for correlations in the noise while computing $h(Z)$. Further, in the case of data-dependent noise prediction (DDNP) [63] that comprises a bank of noise whitening filters that depend on each input pattern, the overall entropy rate conditioned to each input pattern can be evaluated based on the local noise statistics from DDNP filters using the MIR estimation ideas we discussed in this subsection.

2) Capacity Estimate for the 2D Framework: Similar to the 1D case, the overall 2D channel for TDMR systems can be approximated to a 2D finite-state ISI channel with AWGN after noise whitening, described by

$$y_{i,j} = \sum_{k=1}^{M}\sum_{l=1}^{N} h_{k,l}x_{i-k,j-l} + n_{i,j}, \tag{29}$$

where $x_{i,j} \in \{-1,+1\}$ indicates the magnetization of $(i,j)^{\text{th}}$ channel's bit cell, $y_{i,j}$ is the $(i,j)^{\text{th}}$ read-back sample, and $n_{i,j}$ is the realization of noise under Gaussian statistics, i.e., $\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})$. The MIR of the TDMR channel with the probability distribution function $P(\mathbf{y}|\mathbf{x})$ is defined as the mutual information between channel's input $\mathbf{x} = [x_{i,j}]$ and the output $\mathbf{y} = [y_{i,j}]$. We now compute MIR as follows:

$$\text{MIR} = \frac{1}{NM}I(\mathbf{X};\mathbf{Y}) = \frac{1}{NM}H(\mathbf{Y}) - \frac{1}{NM}H(\mathbf{Y}|\mathbf{X}), \tag{30}$$

where $H(\cdot)$, $H(\cdot|\cdot)$, and $I(\cdot;\cdot)$ in equation (30) are the entropy, conditional entropy, and mutual information terms, respectively. Knowing the channel $P(\mathbf{y}|\mathbf{x})$, $H(\mathbf{Y}|\mathbf{X})$ can be computed. The problem of obtaining the MIR reduces to computing the entropy rate of the channel's output $H(\mathbf{Y})$. From Shannon-McMillan-Breimann theorem [64], assuming stationarity and ergodicity, the entropy rate is computed as

$$-\frac{1}{n}\log p(\mathbf{y}) \to H(Y), \tag{31}$$

as $n \to \infty$ with probability 1. By adopting the trellis-based strategy over multiple rows over the 2D array of inputs and outputs, similar to how we outlined the procedure for the 1D case, one could calculate the marginal output distribution $p(y)$ for large n for 2D arrays as well. For more details, the reader is referred to [6].

In general, for 2D channels with memory, it is not known whether a stationary ergodic random field will achieve the capacity [65]. Recently, for a special class of 2D channels, Li and Siegel [66] showed that the operational capacity and information capacity (Shannon capacity) are equal and can be achieved by a stationary ergodic random field with input constraints.

B. GBP based Capacity Estimation Method

Probabilistic graphical models are important in a wide variety of applications from solving combinatorial problems in statistical physics to inference problems in signal processing [67]. These problems can be reformulated equivalently as the computation of marginal probabilities on factor graphs [68] using message passing algorithms, such as the belief propagation (BP) algorithm used in coding theory [69] for decoding low-density parity check (LDPC) codes and in artificial intelligence [70]. Computing marginals of functions on a graphical model has its roots in the broad class of Bayesian inference problems [71].

It is well-known that the BP algorithm gives exact inference only on acyclic graphs, i.e., trees. Further, it is also well-known that BP works poorly on graphs containing a large number of short cycles (subgraphs with girth=4). The problem of TDMR channel capacity estimation can considered as one of the problems corresponding to a factor graph with many short cycles. There are many cycles in a TDMR channel factor graph, referring to Figure 9, which invalidates the tree-like assumption used in BP, leading to poor performance of the BP algorithm. A new class of message passing algorithms called *generalized belief propagation* (GBP) was introduced by Yedidia, Freeman and Weiss [72] to solve this problem[6]. The GBP estimates are approximately close to the true estimate.

Since GBP benefits from region-to-region message passing instead of the node-to-node message passing algorithm of BP, GBP algorithm can often dramatically outperform the BP algorithm in either accuracy or convergence properties. The output probabilities from a 2D channel actually correspond to a Boltzmann distribution of an Ising Hamiltonian, with pairwise interactions and external random fields [73] [74]. The difficulty in estimating *a posteriori* probabilities lies in estimating the partition function of factor graphs, or similarly, the free energy in statistical physics. For the case of capacity

[5] The reader must note that when input sequences are constrained (see for example, RLL constraints) and represented through finite state transition diagrams or equivalently constrained graphs, the stationary distribution of the source can be used as the initial condition.

[6] The reader must note that the estimate obtained through the GBP algorithm is not *exact* in general.

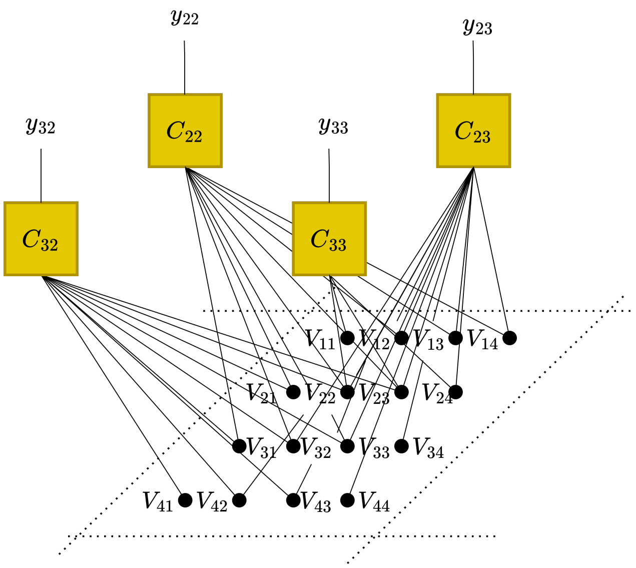

y_{22}
y_{23}
y_{32}
y_{33}
C_{22}
C_{23}
C_{32}
C_{33}
V_{11} V_{12} V_{13} V_{14}
V_{21} V_{22} V_{23} V_{24}
V_{31} V_{32} V_{33} V_{34}
V_{41} V_{42} V_{43} V_{44}

Figure 9. Every 3×3 square within the lattice of 4×4 comprising variables $v_{i,j}$ is controlled by a local function $c_{i,j}$. The bi-indices (i, j) indicate the coordinates of the variable node on the lattice. Adapted from [6].

estimation of TDMR channels, the GBP algorithm can be utilized to estimate the marginal distribution from the channel outputs and consequently the channel capacity. GBP algorithm can be used to estimate the MIR for TDMR channels [44].

The GBP algorithm as a message passing algorithm can operate on the region graph of the TDMR channel to compute the marginal probabilities. The belief of each region as an output of the GBP algorithm is an approximation of the marginal probability. As the GBP is a message passing algorithm, we first introduce the graphical representation for the procedure.

The factor graph is a bipartite graph representing the factorization of a function, comprising a set of random variables $\mathbf{V}$ and a set of local functions (local constraints) $\mathbf{F}$. In the factor graph, random variables $V_i \in \mathbf{V}$ are represented by circles (variable node) and local functions $f_j \in \mathbf{F}$ are illustrated by squares (factor node). A variable node V_i is connected to a factor node f_j if and only if V_i is an argument of f_j. Figure 9 depicts the factor graph corresponding to a 4×4 grid where each 3×3 square region is locally constrained by a factor node.

For a given graphical model, the region graph is generated according to the cluster variation method [72] [75]–[77]. A parent region R is specified by a set of variable nodes and factor nodes such that if $f_j \in R$, then all the variable nodes connected to f_j must be in R. Figure 10 shows the region graph for the factor graph of 2D ISI constraint shown in Figure 9. In this example, we choose each factor node to be in a separate parent region for simplicity. The variable nodes connected to the factor node also reside in that region. The child regions of a region graph are then constructed by taking the intersection of the parent regions, the intersections of the intersections, and so on through the tree.

In the case of capacity estimation setup, the factor function $f(\mathbf{x}_{C_{i,j}}) = p(y_{i,j}|\mathbf{x}_{C_{i,j}})$, and the local constraint is the same for all the parent regions. The partition function Z and the Helmholtz free energy F_H are related as $F_H = -\ln Z$. For the purpose of estimating the information rate, we define the partition function as

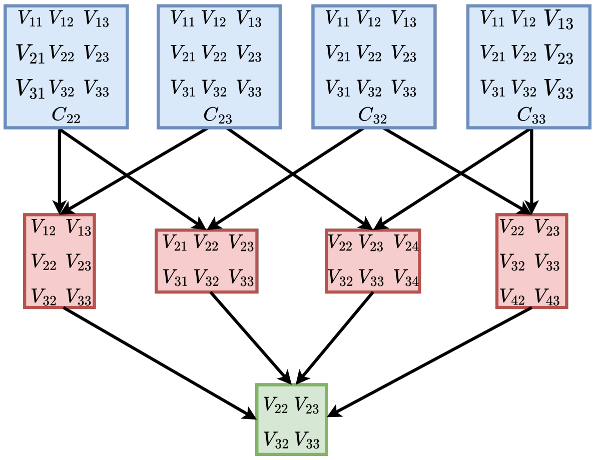

V_{11} V_{12} V_{13}
V_{21} V_{22} V_{23}
V_{31} V_{32} V_{33}
C_{22}
V_{11} V_{12} V_{13}
V_{21} V_{22} V_{23}
V_{31} V_{32} V_{33}
C_{23}
V_{11} V_{12} V_{13}
V_{21} V_{22} V_{23}
V_{31} V_{32} V_{33}
C_{32}
V_{11} V_{12} V_{13}
V_{21} V_{22} V_{23}
V_{31} V_{32} V_{33}
C_{33}
V_{12} V_{13}
V_{22} V_{23}
V_{32} V_{33}
V_{21} V_{22} V_{23}
V_{31} V_{32} V_{33}
V_{22} V_{23} V_{24}
V_{32} V_{33} V_{34}
V_{22} V_{23}
V_{32} V_{33}
V_{42} V_{43}
V_{22} V_{23}
V_{32} V_{33}

Figure 10. Region graph representation of the factor graph in Figure 9. Adapted from [6].

$$Z(\mathbf{y}) = \sum_{\mathbf{x}} \prod_{i,j} p(y_{i,j}|\mathbf{x}_{C_{i,j}}) = \sum_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}), \tag{32}$$

where $f(\mathbf{x}_{C_{i,j}})$ is the factor function explained above.

As previously discussed, the main problem of estimating the MIR reduces to estimating the entropy of the channel output $\mathbf{y}$. For this purpose, we use empirical averaging

$$H(\mathbf{Y}) = -\mathbf{E}_{\mathbf{y}} \log p(\mathbf{y}) \approx -\frac{1}{L} \sum_{l=1}^{L} \log p(\mathbf{y}^{(l)}), \tag{33}$$

where L is the number of samples $\mathbf{y}$ drawn according to $p(\mathbf{y})$. Applying Bayes' law and using the channel model distribution, $p(\mathbf{y})$ can be written as

$$p(\mathbf{y}^{(l)}) = \sum_{\mathbf{x}} p(\mathbf{x}) p(\mathbf{y}^{(l)}|\mathbf{x}), \tag{34}$$

where $\sum_{\mathbf{x}}$ corresponds to a sum over all possible $\mathbf{x} \in \mathcal{X}$.

The output entropy term reduces to

$$\begin{aligned} H(Y) &= -\frac{1}{L} \sum_{l=1}^{L} \log \left(\frac{1}{\mid \mathcal{X} \mid} Z(y^{(l)}) \right) \\ &= \log(\mid \mathcal{X} \mid) - \frac{1}{L} \sum_{l=1}^{L} \log(Z(y^{(l)})), \end{aligned} \tag{35}$$

with uniform input distribution i.e., $p(\mathbf{x}) = \frac{1}{|\mathcal{X}|}$. Therefore, the problem of estimating the mutual information rate of a TDMR channel reduces to the problem of estimating $\sum_{\mathbf{x}} p(\mathbf{y}^{(l)}|\mathbf{x}) = Z(\mathbf{y}^{(l)})$ as in (32). The indicator function can be written as the product of local kernels, each having some subset of $\mathbf{x}$ as an argument i.e., $f(\mathbf{x}) = \prod_a f_a(\mathbf{x}_a)$, where the indices a of the local kernels correspond, for example, to the set of all the three adjacent bits in the horizontal and vertical directions.

Computing Z can be done by the finding the region-based free energy estimate. More precisely, the Helmholtz free energy F_H can be estimated using the region-based free energy approximation technique, giving the partition function Z. If the GBP is used to compute the beliefs of each region $b_R(\mathbf{x}_R)$, using the estimated beliefs, an estimate of the free energy $\hat{F}_H$ can be computed using

$$\hat{F}_H = \sum_{R\in\mathcal{R}} c_R \sum_{\mathbf{x}_R} b_R(\mathbf{x}_R)\left(\ln\, b_R(\mathbf{x}_R) - \ln \prod_{a\in A_R} f_a(\mathbf{x}_a)\right), \tag{36}$$

where $\mathcal{R}$ is the set of all regions, c_R is the counting number, $\mathbf{x}_R$ is the set of variables in R and A_R is the set of local kernels in region R. We use parent-to-child messaging with one kind of message passed between regions. The belief of any region is the product of all the local factors in the region, multiplied by the messages coming into that region and to its descendants from outside. Each region R has a belief $b_R(\mathbf{x}_R)$ given by [72]

$$\begin{aligned} b_R(\mathbf{x}_R) &= \prod_{a\in A_R} f_a(\mathbf{x}_a)\left(\prod_{P\in\mathcal{P}(R)} m_{P\to R}(\mathbf{x}_R)\right) \\ &= \left(\prod_{D\in\mathcal{D}(R)}\ \prod_{P'\in\mathcal{P}(D)\backslash\mathcal{E}(R)} m_{P'\to D}(\mathbf{x}_D)\right), \end{aligned} \tag{37}$$

where A_R is the set of elements in region R and the $f_a(\mathbf{x}_a)$ are the local factors of region R. $\mathcal{P}(R)$ and $\mathcal{D}(R)$ are, respectively, the parent and descendant regions of R. $\mathcal{E}(R) = R \cup \mathcal{D}(R)$ and $\mathcal{P}(D)\backslash\mathcal{E}(R)$ is the set of all regions that are parents of region D except for R and descendants of R.

With the terms, $T_{P\backslash R} = \prod_{a\in F_{P\backslash R}} f_a(\mathbf{x}_a)$ and $T_{N(P,R)} = \prod_{(I,J)\in N(P,R)} m_{I\to J}(\mathbf{x}_J)$, the message-update rule in the parent-to-child algorithm is

$$m_{P\to R}(\mathbf{x}_R) = \frac{\sum_{\mathbf{x}_{P\backslash R}} T_{P\backslash R} T_{N(P,R)}}{\prod_{(I,J)\in D(P,R)} m_{I\to J}(\mathbf{x}_J)},$$

where the set $N(P,R)$ indicates all connected pairs of regions (I,J) such that $J \in \mathcal{E}(P)\backslash\mathcal{E}(R)$, while $I \notin \mathcal{E}(P)$. $D(P,R)$ is the set of all connected pairs of regions (I,J) such that $J \in \mathcal{E}(R)$, while $I \in \mathcal{E}(P)\backslash\mathcal{E}(R)$. $F_{P\backslash R}$ is a set of factor nodes in the region $P\backslash R$.

Example 1: We estimate the MIR by using the GBP algorithm for an $M \times N$ array over the Voronoi channel model. We obtain lower and upper bounds on the GBP-based MIR estimation for a Voronoi channel. The lower and upper bounds merge to the actual value for the MIR estimation of the Voronoi channel with increasing dimensions of the 2D array.

Obtaining the lower bound: In order to compute the beliefs of the boundary regions, we assume that all the states of the boundary regions are equiprobable. Under this assumption and using the GBP algorithm as described before, we establish a lower bound on the MIR of a TDMR system.

Obtaining the upper bound: The boundary information of the medium is known to the MIR estimator. For boundary regions, the values of the boundary variable nodes are given and treated as deterministic in the GBP algorithm. For this case, we compute an upper bound on the MIR of the Voronoi channel.

Table I
ALL THE PARAMETERS IN THE TABLE ARE IN NANOMETERS. WE DENOTE $n_1{:}n_2{:}n_3=\{n_1, n_1+n_2, n_1+2n_2, \dots, n_3\}$. CTC=10NM

	TW	BL	$PW50_x$	$PW50_y$
$TDMR_1$	10:1:20	7	20	14
$TDMR_2$	10:1:20	5:0.5:10	20	10

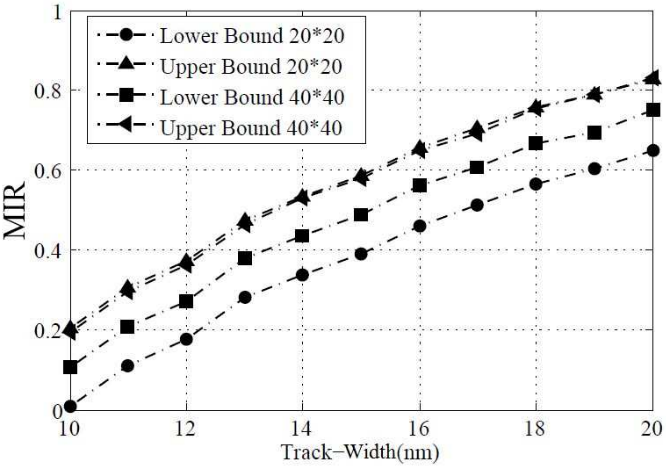


Figure 11. Lower and upper bounds on the MIR for the TDMR_1 system. Increased track width reduces media noise. Adapted from [6].

Figure 11 shows the empirically computed lower and upper bounds for the MIR estimation of the Voronoi channel with random 20×20 and 40×40 bit arrays generated according to the uniform distribution. The parameters of the TDMR_1 system simulated are given in Table I. It is worth noting that the upper bound converges much faster than the lower bound and both the bounds converge for larger array sizes. In other words, for a large array size, the boundary bits can be considered to be known.

IV. Signal Processing for Recording Channels

The read channel is an interfacing circuit between the read head and the HDD controller. Encoded data from the computer or a network is converted to a bipolar current that passes through electromagnet coils and written as flux changes over the storage medium through the write head. Now, when a read head senses the signal from the disk, the readback signal comprising several artifacts due to timing offsets, ISI, thermal asperities, noise etc. must be processed before the data is decoded back.

Variations in the head-media spacing, variations in the magnetic, electrical and mechanical properties during the sensing process all contribute to gains and offsets in the readback signal, affecting the SNR from the read side. The readback signal is first compensated w.r.t gains and offsets via analog control loops. Also, when the read head hits dust particles on the medium, the readback signal appears to be in the shadow of a low-frequency signal of higher energy. This is called *thermal asperity*, which can be detected and compensated (refer to Figure 4).

High-frequency noise is removed using a continuous-time (CT) filter, typically a Butterworth or an equiripple linear-phase filter so that the readback signal without aliasing of the high-frequency noise spectrum can be sampled and processed further. The CT filter has programmable cutoff frequencies. Also, to process servo information towards handling timing, the CT filter would have to be switched to allow low cutoff frequencies. All these aspects are part of the analog front end.

Post-sampling, the readback signal is passed through a timing recovery circuit. Post GPR equalization, the data is processed through a sequence detector. In modern read channels, the sequence detector and the error correction decoder are coupled via a turbo loop within the framework of iterative detection and decoding [78]. We shall now discuss all the details of the signal processing chain, covering both 1D and 2D techniques.

A. Channel Equalization

Early version of HDDs used analog equalization. With the introduction of PRML channels, equalization is done digitally. The sampled readback signal is first equalized using a linear equalizer before the signal is detected using a maximum-likelihood (ML) detector. The linear equalizer reduces the extent of ISI and achieves a desired overall response called the partial response that controls both the complexity and performance of the ML sequence detector. Additional regularization constraints can be forced on the equalizer so that the pre-target is matched to the channel spectrum.

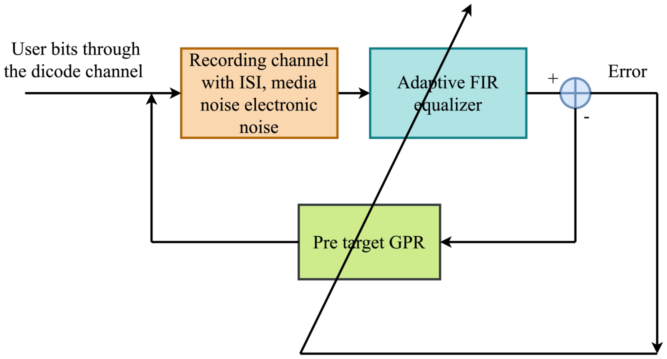

User bits through the dicode channel
Recording channel with ISI, media noise electronic noise
Adaptive FIR equalizer
+
-
Error
Pre target GPR

Figure 12. GPR equalization in 1D. The combined effects of the recording channel and the equalizer are equivalent to a PR channel.

PMR channels have used pre-targets [79] based on polynomials of the form $(1-D)^m(1+D)^n$. Since the parameters m and n influence the choice of the polynomial[7] in terms of performance as well as the complexity of the sequence detector, one could balance the extent of partial response ISI and the overall system performance. One could also adapt the FIR equalizer and the target as shown in Figure 12.

B. Timing Recovery for 1D channels

It is important to synchronize the discrete readback samples so that signal detection can be accomplished post PR equalization. The timing recovery module accomplishes this goal by providing the samples at the desired time instants. Timing errors in HDDs are due to accumulated phase errors, frequency errors and jitter. Timing jitters can be modeled as a discrete random process using random walks [80].

Timing recovery algorithms are of two classes: (a) PLL-based [81] and (b) interpolative timing recovery (ITR)-based [82] (Chapter 27 from [27]). We will first review both these techniques for the 1D case before getting to the shingled case and 2D.

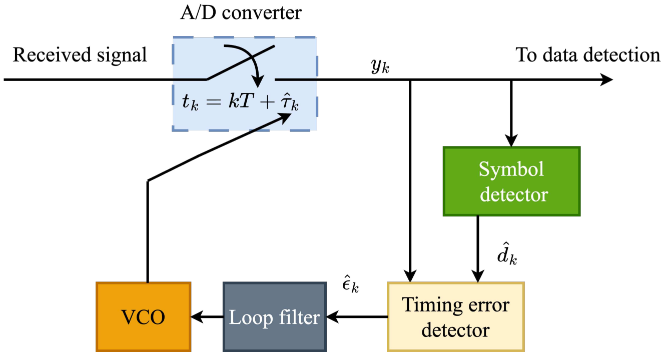


Figure 13. A conventional VCO-based timing recovery. This is part of the loop-Viterbi engine. The timing error estimates from the timing error detector (TED) are passed through the loop filter and a VCO to feed the timing estimates for resampling.

1) PLL-based Timing Error Detector: The noisy equalized sample $y(k)$ is fed to a PLL unit having a phase detector, typically a second order loop filter and a voltage-controlled oscillator (VCO) as shown in Figure 13. The other signal inputs to the phase detector are the ideal/desired values $\hat{y}(k)$ obtained by filtering known data $d(k)$ through the PR target. The phase detector obtains the misalignment between the ideal samples and the actual samples from the sampler output. Using the Mueller and Muller estimate [83], the timing gradient is computed as

$$\hat{\epsilon}(k) = y(k)\hat{y}(k-1) - y(k-1)\hat{y}(k). \tag{38}$$

The estimated timing error $\hat{\epsilon}(k)$ is now filtered through a second order loop filter with additional delays z^{-L} to handle any timing loop latencies. The sampling offsets are updated according to the following equations:

$$\hat{\theta}_k \ = \hat{\theta}_{k-1} + \beta\hat{\epsilon}_k, \tag{39}$$

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha\hat{\epsilon}_k + \hat{\theta}_k, \tag{40}$$

where α and β are the PLL parameters for gain adjustment and for controlling the loop bandwidth and convergence rate, $\hat{\theta}_k$ corresponds to the frequency error and $\hat{\tau}_k$ is the adjusted

[7] Historically, $(1+D)$ is referred to as PR1, $(1-D)(1+D)$ is referred to as PR4, $(1-D)(1+D)^2$ is referred to as EPR4 etc. in the data storage community.

timing offset. The reader must note that a *preamble* is used for training the PLL in the acquisition mode.

2) Interpolative Timing Recovery: The VCO unit of the PLL-based timing recovery can be made digital using the ITR scheme. In this method, received samples are resampled at a higher rate $\sim 2-5\%$ of the baud rate and interpolated using a digital filter for obtaining the correct timing sample. The digital filter is obtained using the MMSE criterion that minimizes the MSE between ideal and the interpolated samples.

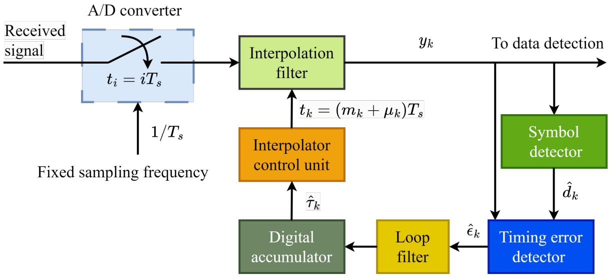


Figure 14. The interpolative timing recovery unit comprises an interpolation filter which adjusts the timing instants in the oversampled domain based on the sampling phase offsets.

Figure 14 shows the ITR architecture. The sampling time instant $t_k = kT + \hat{\tau}_k = (m_k + \mu_k)T_s$. The integer m_k is a multiple of T_s and μ_k is a fraction of the oversampled time period. With 5% oversampling, the oversampling rate $f_s = \frac{1}{T_s}$ is typically related as $E(T) = 1.05T_s$ since T is non-ideal and is a random variable due to clock jitters.

Without any timing error, binary sequence $a_k \in \{-1, 1\}$ is filtered through the PR equalizer with PR target $p(t)$ to yield the samples $y(kT)$ given by

$$\hat{y}(kT) = \sum_{m=-\infty}^{\infty} a_m p(kT - mT). \tag{41}$$

Using the interpolating filter f_{μ_k} for the phase μ_k over a span $n_1 + n_2 + 1$ samples, we have

$$y(kT) = \sum_{l=-n_2}^{n_1} f_{\mu_k}(l) y_{\mathrm{ip}}((m_k - l)T_s), \tag{42}$$

where the interpolated signal $y_{\mathrm{ip}}(kT - \mu_k T_s - lT_s)$ is given by

$$\begin{aligned} y_{\mathrm{ip}}(kT - \mu_k T_s - lT_s) = \sum_{i=-\infty}^{\infty} a_i p\left(kT - \mu_k T_s - lT_s - iT\right) \\ + n\left(kT - \mu_k T_s - iT_s\right). \end{aligned} \tag{43}$$

The optimal filter $\mathbf{f}_{\mu_k}$ for the sampling phase μ_k is solved using

$$\mu_k^* = \min_{\mu_k} \mathbf{E}\left[\left(\hat{y}(kT) - y(kT)\right)^2\right]. \tag{44}$$

The estimation of the initial sampling phase can be done using a preamble and this process is called the digital zero phase start. Further, the reader must note that the solution of equation (44) can be obtained using adaptive algorithms such as the LMS algorithm by estimating the coefficients of the adaptive filter when the channel conditions change. In practice, one could you simple interpolative methods, such as a linear interpolator at the cost of performance degradation. Since solving for optimal filters is not practical for high speed circuits, one can design a bank of such filters under quantized phase offsets and use an appropriate filter based on the timing phase estimation.

C. Signal Detection for MR channels

Early versions of PRML channels used hard decision Viterbi detectors [84] [85]. With the advent of turbo codes [86] and turbo equalization [78], almost all HDDs use the soft-decision Viterbi algorithm (SOVA) [87] [88] for signal detection. Post equalization and timing, the Viterbi algorithm obtains the ML-optimal sequence $\mathbf{b}$ of length N from the noisy version of the sequence $\mathbf{y}$ by finding

$$\mathbf{b}_{ML} = \arg\max_{(\mathbf{b})} P(\mathbf{y}|\mathbf{b}). \tag{45}$$

Since the signal detection is over a PR target, using the memory of the equivalent ISI channel i.e., corresponding to the PR target, a trellis structure amenable to a desired level of the computational complexity can be chosen. If I is the memory of the channel ISI, we define the states $S_k := b_{k-I}^{k-1}$ and $S_{k-1} := b_{k-I+1}^{k}$ corresponding to the memory of the Markov process. Over the trellis stages, the conditional probability of the received sequence given the input bit sequence is given by

$$P(\mathbf{y}|\mathbf{b}) = \prod_{k=0}^{N-1} P(y_k|S_k, S_{k-1}). \tag{46}$$

The quantity $P(y_k|S_k, S_{k-1}) = P(n_k)$ is the probability of the noise sample at time instant k. Assuming that the noise statistics are Gaussian $\mathcal{N}(0, \sigma^2)$, given the received value y_k and the ideal value o_k, one could easily compute $n_k = y_k - o_k$ by plugging it in equation (46). For numerical stability, the computations can be done in the logarithmic domain.

$$\log P(\mathbf{y}|\mathbf{b}) \propto -\sum_{k=0}^{N-1} \left(y_k - o_k\right)^2. \tag{47}$$

The state metric $\mathrm{SM}_k^{(i)}$ for a trellis state i at time k is related by the recursion

$$\mathrm{SM}_k^{(i)} = \mathrm{SM}_{k-1}^{(i)} + \mathrm{BM}_k, \tag{48}$$

where BM is the least among the branch metrics that connects state i at time k from any other state j at time $k-1$. This is commonly referred to as the add-compare-select (ACS) logic. By doing the recursion over the entire length of the trellis after picking up the state metric that is the least among all the states, one could back trace the path, reading the labels of the bits in the reverse order corresponding to the path with the least state metric at time N. This is the essence of the Viterbi sequence detection.

The SOVA provides soft information by computing the probability of a wrong/complementary decision through the

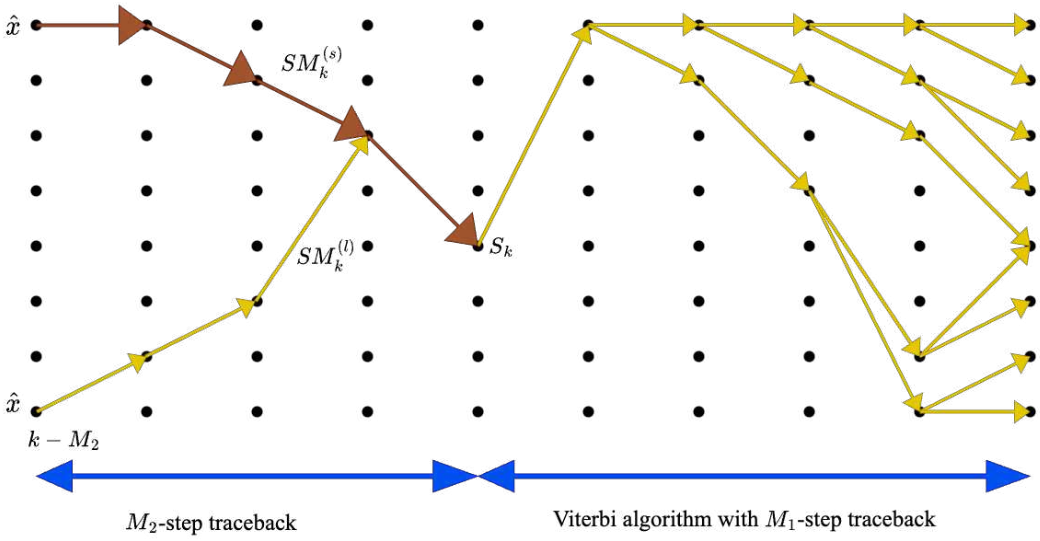


Figure 15. Using a M_1 step traceback, through the Viterbi algorithm ML path is determined. Using a M_2 step traceback, the second best path is found using the SOVA.

survivor paths on the trellis. Specifically, at any time instant k, following a M_1-step traceback from the hard decision Viterbi algorithm, the state metric of the survivor is obtained as $\mathrm{SM}_k^{(s)}$. Now, with a M_2-step traceback further from the time instant k, the loser path or the second best ML path yielding a complementary decision has a state metric $\mathrm{SM}_k^{(l)} < \mathrm{SM}_k^{(s)}$.

From the SOVA algorithm, the probability of choosing the second best path over the ML path is $P_{\text{err}} = \frac{1}{1+e^{\Delta_k}}$, where $\Delta_k = \mathrm{SM}_k^s - \mathrm{SM}_k^l$, corresponding to the log-likelihood ratio or odds for the survivor decision to be correct. The soft decisions that SOVA allows helps in the turbo-equalization process while dealing with an iterative ECC decoder. At this stage, we note that one could obtain the MAP decisions for signal detection based on the elegant BCJR algorithm [61]. However, latencies on the order of the sequence length and area/power complexities limit use of the BCJR in hardware. For the aforementioned reasons, though the SOVA is sub-optimal in performance, it is the state-of-the-art algorithm [89] implemented in read channel ICs. Recently, the authors in [90] have proposed an asynchronous version of the SOVA with an eye towards low-power design.

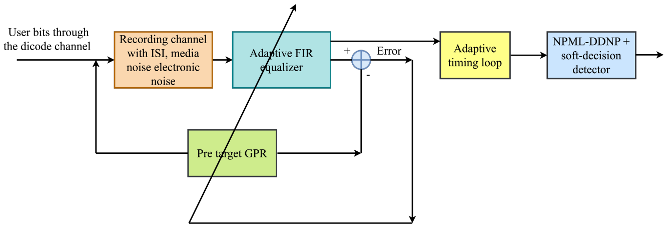

User bits through the dicode channel
Recording channel with ISI, media noise electronic noise
Adaptive FIR equalizer
+
-
Error
Adaptive timing loop
NPML-DDNP + soft-decision detector
Pre target GPR

Figure 16. Post equalization and timing, noisy samples obtained as as difference from the ideal samples and the equalized samples are whitened using the NPML/DDNP. The noise statistics extracted is fed to the soft-decision detector within a turbo detector/decoder setup.

To further improve channel reliabilities, subsequent versions of soft-decision-based signal detectors used NPML detection [13]. Figure 16 shows the noise path after adaptive equalization and timing recovery. Colored noise must be whitened and the noise statistics are fed to the detector for branch metric computation. With experimental evidence into pattern-dependent noise arising due to magnetic transitions, the DDNP-detection algorithm was conceived [63]. This is now the standard algorithm residing in HDDs that we shall describe as follows:

For each data pattern $\mathbf{b}$ of length L, the noise observed at the output of the detector can be modeled using an autoregressive (AR) process. In other words, we group the noise samples from the input data sequence specific to each data pattern $\mathbf{b}$ and predict this using a linear predictor.

$$\hat{n}_k^{(\mathbf{b})} = \sum_{l=1}^{L} a_l n_{k-l}^{(\mathbf{b})} + e_k^{(\mathbf{b})}. \tag{49}$$

The filter coefficients $a_l^{(\mathbf{b})}$ for $1 \leq l \leq L$ are solved using the MMSE criterion:

$$\mathbf{a}^{*(\mathbf{b})} = \min_{\mathbf{a}^{(\mathbf{b})}} \mathbf{E}\left[(n_k^{(\mathbf{b})} - \hat{n}_k^{(\mathbf{b})})^2\right]. \tag{50}$$

The noise variance for each pattern $\mathbf{b}$ is its corresponding prediction error $E(e_k^2)$. The noise is now whitened using this data-dependent prediction filter referred to as DDNP while computing the branch metric within the SOVA algorithm. The reader must note that in practice, one can account for bipolar symmetry in the data patterns to reduce the filterbank size by half, amenable for hardware. For example, for a 4-bit pattern, 0101 and 1010 are bipolarly similar. These patterns can share the same DDNP filter. DDNP filters can also be adapted [91] using the LMS algorithm towards ease of hardware implementation.

With this background, we are now set to discuss the signal processing algorithms and architectures for shingled recording and native 2-D signal processing algorithms for TDMR channels.

D. *Signal processing for shingled systems*

As described earlier, shingled systems work with overlapped writes. With multiple read heads positioned over two or more tracks, one can embrace handling crosstalk along and across the tracks as well as synchronization issues in 2D by processing the tracks jointly. These techniques are part of shingled systems. We will discuss some signal processing architectures for single-track as well as the multi-track cases.

1) Single-track case: In single-track detection, the read heads are positioned over a single track of interest [92] [93] to extract the timing information and to detect and recover the bits.

In the signal processing architecture for single track detection with multiple readers, each readback waveform is separately equalized before being added together. The architecture is equivalent to a multiple-input single-output (MISO) equalizer [6] within the GPR equalization framework [94] with monic constraints that we discussed earlier. The equalizer handles any crosstalk from the neighboring tracks during the reads. Using 1D strategies, such as the interpolated timing recovery (ITR), the timing offsets can be handled prior to detection. The rest of the signal chain proceeds with a 1D detector with DDNP capability post equalization and timing recovery using the ideas that we discussed earlier. In this architecture since the crosstalk is suppressed asynchronously prior to synchronization, a 1D timing loop is sufficient post equalizer, prior to detection.

2) Multi-track case: Additional SNR gains can be obtained by doing multi-track detection [95] [96]. In [97], [98], the authors proposed a detector that uses a different trellis structure whose output labels are independent of the inter track interference (ITI) level, with ITI-dependence appearing only in a scale factor for weighing the computed path metrics towards retaining ML optimality. The detector formulation can track the time-varying ITI and provide ITI estimates to adaptively adjust the weights in the path metric evaluation. However, these techniques assume that the tracks are synchronous.

Multiple tracks can be jointly processed for doing equalization using the multiple-input multiple-output (MIMO) framework[8] since 2D ISI contains significant energy that must be processed to provide improved reliability. In the signal processing architecture for multiple readers and multitrack detection, the monic constraint for the single-track case is extended to a 2×2 target with certain constraints [80]. By minimizing the norm of the error vector, the equalizer and the target filters are solved using the GPR framework. In [99] [100], the authors proposed a *remix* strategy to handle synchronization. In their approach, the cascade of the MIMO channel and the equalizer is forced equivalent to a pair of MISO equalizers i.e., in diagonal form, termed as 'unmixing' stage. Post the unmixing, the equalized streams are separately processed using a timing recovery module, such as the ITR algorithm. After this, a MIMO filter is used to 'remix' the signals to restore the non-diagonal nature of the original MIMO channel using a whitening filter. The remixed signal is used subsequently for detection.

For jointly detecting asynchronous multiple tracks, the rotating-target (ROTAR) algorithm in which a time-varying target based on the per-survivor processing timing recovery algorithm can jointly handle timing recovery and detection [101]. This work is also recently extended towards a MIMO-based PR equalization and multi-track detection, where detected data is written asynchronously [102].

Except for equalization and pre-target selection, the modules for rest of the signal chain are still 1D. Perfect equalization and timing may not possible. There can be residual 2D crosstalk which must be removed by the detector for improved decisions. This necessitates the design of native 2D algorithms for TDMR channels. Within a 2D framework, we could generalize 1D signal processing techniques to work with crosstalk and timing errors in 2D.

E. *Signal processing for 2D channels*

With large rectangular array sizes, TDMR channels are entirely 2D, requiring native 2D signal processing techniques. We first discuss the GPR equalization strategy that can work for both separable and non-separable 2D filters. This approach does not impose any specific constraint on the PR target to be in lower-triangular form as discussed in the shingled case [6]. This work generalizes the GPR equalization strategy done for the 1D case directly to 2D.

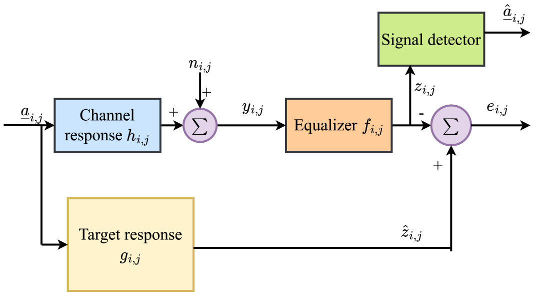


Figure 17. Schematic of the 2D GPR equalization. The 2D GPR target could be separable or non-separable, influencing the performance and complexity of signal detection later in the signal chain.

1) GPR Equalization: Figure 17 shows the PR equalization scheme for the 2D channel. The schematic for the 1D case follows similarly. Let $a_{i,j} \in \{-1, 1\}$ be an array of 2D bits. Let $y_{i,j} \in \mathbf{R}$ be the discrete readback samples. Let $\mathbf{F} := [f_{i,j}]$ and $\mathbf{G} := [g_{i,j}]$ be the coefficients of the PR equalizer and the target. Using a vector notation, we can raster F and G as $\underline{f} = \mathrm{vec}(\mathbf{F})$ and $\underline{g} = \mathrm{vec}(\mathbf{G})$. The samples at the output of the equalizer corresponding to the local span of the input samples are given by

$$z_{i,j} = \underline{f}^T \mathbf{y}^{(i,j)}. \tag{51}$$

Similarly, following the signal path in Figure 17, the output of the pre-target is given by

$$\hat{z}_{i,j} = \underline{g}^T \mathbf{a}^{(i,j)}. \tag{52}$$

With the error $e_{i,j} = z_{i,j} - \hat{z}_{i,j}$, we setup the MMSE criterion as follows:

$$\mathcal{E} = E\left(e_{i,j}^2\right). \tag{53}$$

[8] The MIMO framework is suitable when the number of readers is small.

Using equations (51) and (52) and expanding equation (53), we have

$$\mathcal{E} = \underline{\mathbf{f}}^T \mathbf{R}_{yy} \underline{\mathbf{f}} + \underline{\mathbf{g}}^T \mathbf{R}_{aa} \underline{\mathbf{g}} - 2\underline{\mathbf{g}}^T \mathbf{R}_{ay} \underline{\mathbf{f}}, \tag{54}$$

where

$$\begin{aligned} \mathbf{R}_{aa} &= \mathbf{E}\left(\mathbf{a}^{(i,j)} \left(\mathbf{a}^{(i,j)}\right)^T\right), \\ \mathbf{R}_{yy} &= \mathbf{E}\left(\mathbf{y}^{(i,j)} \left(\mathbf{y}^{(i,j)}\right)^T\right), \\ \mathbf{R}_{ay} &= \mathbf{E}\left(\mathbf{a}^{(i,j)} \left(\mathbf{y}^{(i,j)}\right)^T\right). \end{aligned}$$

Rewriting equation (54), we have the quadratic form:

$$\begin{aligned} \mathcal{E} = &\left(\underline{\mathbf{f}} - R_{yy}^{-1} R_{ay} \underline{g}\right)^T \mathbf{R}_{yy} \left(\underline{\mathbf{f}} - R_{yy}^{-1} R_{ay} \underline{g}\right) \\ &+ \underline{\mathbf{g}}^T \left(R_{aa} - R_{ay} R_{yy}^{-1} R_{ay}^T\right) \underline{\mathbf{g}}. \end{aligned} \tag{55}$$

The equalizer $\underline{\mathbf{f}} = R_{yy}^{-1} R_{ay} \underline{g}$ is the solution for a chosen pre-target. One can however enforce certain regulatory constraints to solve for the target as well.

We have the following two cases:

1) Unit energy constraint: With $\underline{\mathbf{g}}^T \underline{\mathbf{g}} = 1$, we can solve $\underline{\mathbf{g}}^*$ to the eigenvector corresponding to the smallest eigenvalue of $R_{aa} - R_{ay} R_{yy}^{-1} R_{ay}^T$.
2) Monic constraint: With the constraint $\underline{\mathbf{u}}^T \underline{\mathbf{g}} = 1$, $\underline{\mathbf{g}}^* = \frac{R_{aa} - R_{ay} R_{yy}^{-1} R_{ay}^T \underline{\mathbf{u}}}{\underline{\mathbf{u}}^T R_{aa} - R_{ay} R_{yy}^{-1} R_{ay}^T \underline{\mathbf{u}}}$.

For 2D targets, one can also consider separable 1D filters. For more details, the reader is referred to [103]. It is important to note that under time-varying channel conditions, we could adapt the targets and equalizers as well. This can be easily handled within this GPR framework through adaptation based on the LMS algorithm.

2) 2D Timing Recovery Techniques: Timing recovery for 2D is done using the (a) 2D PLLs, (b) 2D ITR recovery schemes and (c) 2D joint timing detection schemes.

The servo mechanisms while reading are far from ideal conditions due to mechanical vibrations and shocks on the read heads. These are nanoscale events[9]. 2D timing recovery has advantages since asynchronicity across multiple tracks can be taken to our advantage to obtain synchronous tracks by jointly processing these tracks using suitable algorithms as discussed earlier. Within a linear approximation setup, readback signal is modeled as filtering the written data $d_{\mathbf{k}}$ through the 2D head response $h(\mathbf{t})$ along with timing offsets $\boldsymbol{\tau}$ as [104]:

$$r(\boldsymbol{t}) = \sum_{\boldsymbol{k} \in \mathbb{Z}^2} d_{\boldsymbol{k}} h(\boldsymbol{t} - \boldsymbol{k}^T \boldsymbol{T} - \boldsymbol{\tau}(\boldsymbol{k})) + n(\boldsymbol{t}), \tag{56}$$

where $\boldsymbol{t} = [x, y]^T$, $\boldsymbol{k} = [m, n]^T$, $\boldsymbol{T} = \mathrm{diag}(T_x, T_y)$, $\boldsymbol{\tau} = [\tau_x, \tau_y]^T$, all in 2D. The terms T_x and T_y represent the spatial bit periods along the x and y directions. Similarly, τ_x and τ_y represent the timing errors along the x and y directions. NRZ coded binary data stored on the media is represented by $d_{\boldsymbol{k}}$, where $\boldsymbol{k} \in \mathbb{Z}^2$. The term $n(\boldsymbol{t})$ represents the electronic noise associated with the readback process and can be assumed to be normally distributed in two-dimensions. Whitening filters can remove any noise coloration due to filtering or jitter.

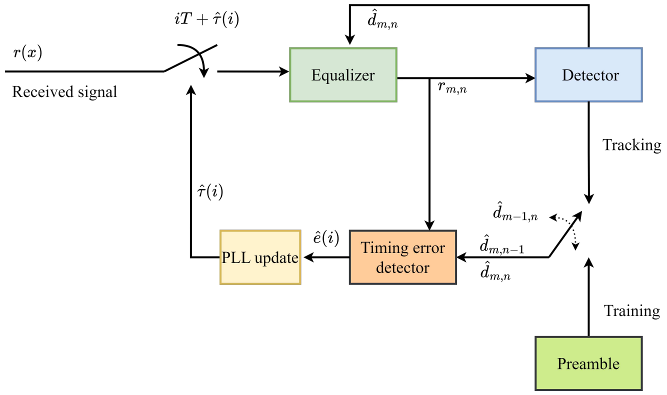


Figure 18. Schematic architecture of the 2D PLL scheme. The timing estimates from the PLL are used to adjust the sampling instants. Adapted from [6].

Timing errors in TDMR can be a combination of both phase and frequency errors on a 2D surface. Figure 18 shows the 2D PLL architecture for correcting these timing errors. Let the phase errors along the x and y directions be ϕ_x and ϕ_y respectively. Similarly, let $\delta T_x^{(x)}$ and $\delta T_y^{(y)}$ be the period offsets along the x and y directions. The overall timing error for separable frequency offsets due to a direction dependent timing error can be modeled as

$$\boldsymbol{\tau}(\boldsymbol{k}) = \boldsymbol{\Phi} + \boldsymbol{m}^T \boldsymbol{B} + n(\boldsymbol{k}), \tag{57}$$

where $\boldsymbol{\Phi} = [\phi_x, \phi_y]^T$, $\boldsymbol{B} = \mathrm{diag}(\delta T_x^{(x)}, \delta T_y^{(y)})$.

Frequency drifts in 2D can result in non-separable timing offsets that could be modeled by modifying $\boldsymbol{B}$ to allow projections of the timing errors in the x and y directions as

$$\boldsymbol{B} = \begin{bmatrix} \delta T_x^{(x)} & \delta T_y^{(x)} \\ \delta T_x^{(y)} & \delta T_y^{(y)} \end{bmatrix}, \tag{58}$$

where $\delta T_x^{(y)}$ and $\delta T_y^{(x)}$ represent the projections on the y and x directions due to period offsets. Non-separable errors can occur due to both direction and position dependent physical errors in the servo system. We discuss several 2D timing recovery schemes relevant to TDMR systems.

Upon sampling the readback signal with timing errors, using the matrix $\boldsymbol{T}$, we obtain

$$\begin{aligned} r(\boldsymbol{i}^T \boldsymbol{T}) = d_{\boldsymbol{i}} h(-\boldsymbol{\tau}(\boldsymbol{i})) + &\sum_{\boldsymbol{k} \neq \boldsymbol{i} \in \mathbb{Z}^2} d_{\boldsymbol{k}} h(\boldsymbol{i}^T \boldsymbol{T} - \boldsymbol{k}^T \boldsymbol{T} - \boldsymbol{\tau}(\boldsymbol{k})) \\ &+ n(\boldsymbol{i}^T \boldsymbol{T}), \end{aligned} \tag{59}$$

where $\boldsymbol{i} = [m, n]^T$ represents the 2D coordinates of the samples. The first term in (59) represents the encoded bits written on the medium, evidently distorted by the presence of timing errors. The second term represents the 2D ISI that needs to be mitigated by the equalizer. The third term is the electronic noise component with Gaussian statistics.

[9] It is often remarked by practitioners in the industry that a flying read head over a medium at nanoscales is equivalent to Boeing 747 flying several meters above the ground. Alignment of the head on the right track is equivalent to a flight landing carefully on a narrow air strip.

In a PLL driven timing recovery architecture, the timing errors are corrected by changing the sampling instants $\boldsymbol{i}^{\boldsymbol{T}}\boldsymbol{T}$ to $\boldsymbol{i}^{\boldsymbol{T}}\boldsymbol{T}+\hat{\boldsymbol{\tau}}(\boldsymbol{i})$, where, $\hat{\boldsymbol{\tau}}(\boldsymbol{i})$ are the estimated timing errors. These timing estimates are generated prior to the sampling instant $\boldsymbol{i}^{\boldsymbol{T}}\boldsymbol{T}$ based on the past samples available from previously sampled data of the readback signal. The corrected sampling process is given by

$$\begin{aligned} r(\boldsymbol{i}^T\boldsymbol{T}+\hat{\boldsymbol{\tau}}(\boldsymbol{i})) &= d_{\boldsymbol{i}}h(\hat{\boldsymbol{\tau}}(\boldsymbol{i})-\boldsymbol{\tau}(\boldsymbol{i}))+ \\ \sum_{\boldsymbol{k}\neq\boldsymbol{i}\in\mathbb{Z}^2} & d_{\boldsymbol{k}}h(\boldsymbol{i}^T\boldsymbol{T}-\boldsymbol{k}^T\boldsymbol{T}+\hat{\boldsymbol{\tau}}(\boldsymbol{k})-\boldsymbol{\tau}(\boldsymbol{k}))+n(\boldsymbol{i}^T\boldsymbol{T}+\hat{\boldsymbol{\tau}}(\boldsymbol{i})). \end{aligned} \tag{60}$$

The PLL-based timing architecture shown in Figure 18 is a decision directed scheme. A loop Viterbi detector is included to provide decision estimates on the individual bits. The reader must note that this loop Viterbi is a data-aided detector, and different from the signal detector before the ECC decoder in typical turbo-equalization setup. The PLL operates in two modes: a) the acquisition mode using data from the preamble, and b) the tracking mode which uses estimated decision information from the detector. The estimated error components, $\hat{e}_x$ and $\hat{e}_y$ are filtered using a loop filter. The sampling at the $\boldsymbol{i}^{\text{th}}$ instant is done using the estimated timing offsets along x and y directions i.e., using the components of $\hat{\boldsymbol{\tau}}(\boldsymbol{i})$. This timing module is entirely digital.

Timing Error Detector (TED):

Based on the current and past sampled values, $r_{m,n}$, $r_{m-1,n}$ and $r_{m,n-1}$ and the corresponding decisions on these samples, the TED can generate the phase error estimates. For this, we bring in a bit of signal geometry into the update equations, extending the ideas of [83] naturally to a 2D setting. Consider a vector of received samples $\vec{R}_{\boldsymbol{i}} = r_{m,n}\vec{i}+r_{m-1,n}\vec{j}+r_{m,n-1}\vec{k}$, where, $\vec{i}$, $\vec{j}$ and $\vec{k}$ are unit orthonormal vectors in a 3D space. Similarly, a corresponding decision vector/ideal vector $\vec{D}_{\boldsymbol{i}} = \hat{d}_{m,n}\vec{i}+\hat{d}_{m-1,n}\vec{j}+\hat{d}_{m,n-1}\vec{k}$ is formulated. To achieve synchronization on a two dimensional grid, the angle between the two vectors, $\vec{R}$ and $\vec{D}$ must be minimized. The angle between the two vectors at the $\boldsymbol{i}^{\text{th}}$ instant, $\theta_{\boldsymbol{i}}$ is given by

$$\sin(\theta_{\boldsymbol{i}}) = \frac{\vec{R}_{\boldsymbol{i}}\times\vec{D}_{\boldsymbol{i}}}{|\vec{R}_{\boldsymbol{i}}||\vec{D}_{\boldsymbol{i}}|}\vec{n}. \tag{61}$$

For small angles, $\sin(\theta_{\boldsymbol{i}}) \approx \theta_{\boldsymbol{i}}$. This assumption is valid since the timing drifts are a small fraction of the sampling times. Ignoring the denominator term in (61), we can minimize the square of the numerator. The angle $\theta_{\boldsymbol{i}}$ can be written as

$$\begin{aligned} \theta_{\boldsymbol{i}}^2 \approx &(r_{m,n}\hat{d}_{m-1,n}-r_{m-1,n}\hat{d}_{m,n})^2+ \\ &(r_{m,n}\hat{d}_{m,n-1}-r_{m,n-1}\hat{d}_{m,n})^2+ \\ &(r_{m-1,n}\hat{d}_{m,n-1}-r_{m,n-1}\hat{d}_{m-1,n})^2. \end{aligned} \tag{62}$$

The minimization of θ would involve minimizing each individual term in (62). We now define the terms $\hat{e}_x(\boldsymbol{i})$, $\hat{e}_y(\boldsymbol{i})$ and $\hat{e}_{xy}(\boldsymbol{i})$ as

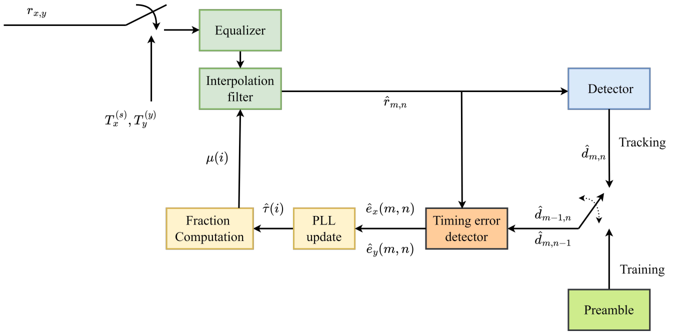


Figure 19. Schematic architecture of the 2D ITR scheme. A 2D interpolating filter provides the timing estimates in the oversampled domain.

$$\hat{e}_x(\boldsymbol{i}) = r_{m,n}\hat{d}_{m-1,n}-r_{m-1,n}\hat{d}_{m,n}, \tag{63}$$

$$\hat{e}_y(\boldsymbol{i}) = r_{m,n}\hat{d}_{m,n-1}-r_{m,n-1}\hat{d}_{m,n}, \tag{64}$$

$$\hat{e}_{xy}(\boldsymbol{i}) = r_{m-1,n}\hat{d}_{m,n-1}-r_{m,n-1}\hat{d}_{m-1,n}. \tag{65}$$

The term $\hat{e}_{xy}(\boldsymbol{i})$ can be expressed as a linear combination of $\hat{e}_x(\boldsymbol{i})$ and $\hat{e}_y(\boldsymbol{i})$, simplifying the minimization of θ^2, since it is now sufficient to minimize $\hat{e}_x$ and $\hat{e}_y$.

PLL Update Equations:

We consider a second order 2D PLL for tracking. The update equations are given by

$$\begin{aligned} \hat{\tau}_x(m+1,n+1) &= \hat{\tau}_x(m,n)+K_x^{(p)}\hat{e}_x(m,n) \\ +K_x^{(ix)}\sum_{l=-\infty}^{m-1}&\hat{e}_x(l,n)+K_y^{(ix)}\sum_{l=-\infty}^{n-1}\hat{e}_x(m,l), \end{aligned} \tag{66}$$

and

$$\begin{aligned} \hat{\tau}_y(m+1,n+1) &= \hat{\tau}_y(m,n)+K_y^{(p)}\hat{e}_y(m,n) \\ +K_x^{(iy)}\sum_{l=-\infty}^{m-1}&\hat{e}_y(l,n)+K_y^{(iy)}\sum_{l=-\infty}^{n-1}\hat{e}_y(m,l). \end{aligned} \tag{67}$$

Here, $K_x^{(p)}, K_y^{(p)}$ are the proportional constants used to scale the error estimates $\hat{e}_x$ and $\hat{e}_y$ respectively. $K_x^{(ix)}$ and $K_y^{(ix)}$ are the integral scaling factors associated with $\hat{e}_x$ along the x and y directions respectively. Similarly, $K_x^{(iy)}$ and $K_y^{(iy)}$ are the integral scaling factors associated with $\hat{e}_y$ along the x and y directions respectively. Real-time control of oscillators in the timing loop is difficult to realize in practice. To overcome these issues, a fully digital 2D interpolative timing recovery architecture (ref. to Figure 19) is proposed in [105]. The 2D readback signal is oversampled by a small amount along both the directions. The oversampling requirement in 1D magnetic storage systems is $\sim 5-10\%$. The specifications in 2D are similar. The interpolative timing recovery architecture [104], [105], shown in Figure 19 is applicable for TDMR systems. These ideas generalize the timing recovery techniques going beyond those for shingled systems.

The interpolation scheme provides more refined estimates of the desired sampling point instead of requiring an ADC or a

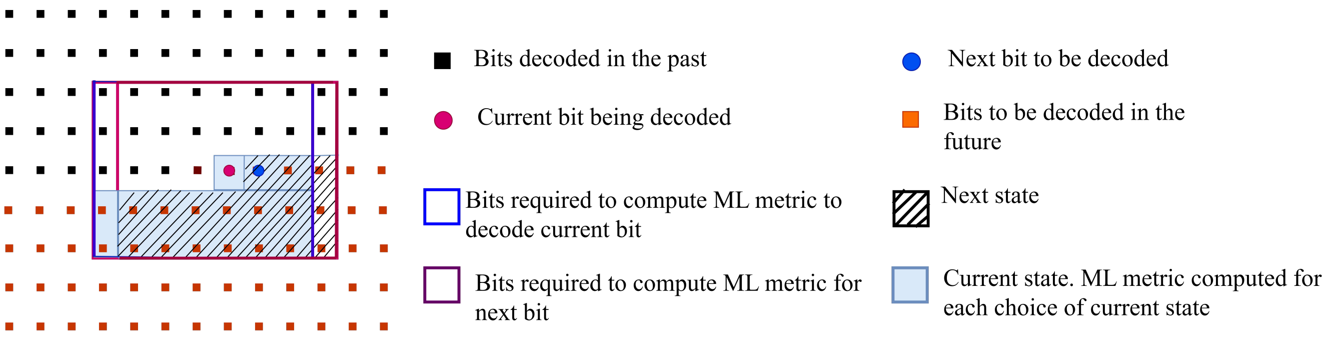

Bits decoded in the past
Next bit to be decoded
Current bit being decoded
Bits to be decoded in the future
Bits required to compute ML metric to decode current bit
Next state
Bits required to compute ML metric for next bit
Current state. ML metric computed for each choice of current state

Figure 20. Viterbi detector extended in 2D for soft-decision detection. The decisions are based on the scanning order of the 2D array confined to a local region $\mathbb{M}$.

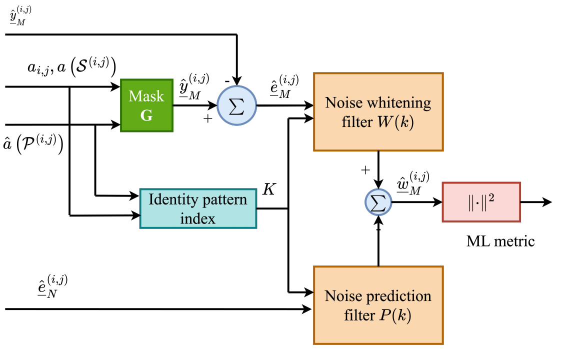


Figure 21. 2D pattern-dependent noise prediction and whitening filters are used within the signal detector to improve the quality of soft-decisions.

servo to latch on to a sampling point [6]. The derivation of an optimal interpolation filter towards timing recovery based on the MMSE criterion is given in [105]. Adaptation algorithms such as LMS can also be done to overcome the limitations of the direct solution obtained through the MMSE criterion while dealing with hardware.

The 2D ITR approach proposed in [104] shown in Figure 19 is demonstrated to provide superior gains in both timing estimates and implementation complexity for a fixed filter order compared to the sinc-based interpolation approach, paving the way for possible circuit realizations of this architecture.

Now, we are set to discuss 2D signal detection techniques. Though this topic is a detailed magazine article in itself, we try to provide a gist of the ideas behind the signal detection techniques. The detailed mathematical derivations can be referred to in the cited papers.

3) 2D Signal Detection and Joint Signal Processing Engines: The 2D SOVA algorithm is a generalization of the 1D SOVA detector developed in [31]. Among all possible 2D arrays, the one that maximizes the likelihood probability is to be chosen. Since computing the likelihood over an entire array is computationally infeasible [10], we can restrict the search to a local region to make decisions for the individual bits. Unlike the 1D case, decisions depend on the scanning order of the 2D array when confined to a local search. Over a local region $\mathbb{M}$ of the 2D array with the received samples $\underline{y}_{\mathbf{M}}^{(i,j)}$,

$$P\left(\underline{y}_{\mathbf{M}}^{(i,j)}|\mathbf{a}\right) \propto \frac{1}{2\sigma^2}\underbrace{||\underline{y}_{\mathbf{M}}^{(i,j)} - \underline{\hat{y}}_{\mathbf{M}}^{(i,j)}||^2}_{=\Gamma},$$

where $\underline{\hat{y}}_{\mathbf{M}}^{(i,j)}$ denotes the ideal samples $\hat{y}^{(i,j)} = \mathbf{g}^T\mathbf{a}_{\mathbf{G}}^{(i,j)}$ obtained by filtering the 2D data through the pre-target.

By minimizing the ML metric Γ, one can obtain the hard decisions. These decisions are made by decomposing the neighborhood around the point (i,j) of interest into 3 parts, comprising (a) regions where decisions are already made $\mathcal{P}^{(i,j)}$, (b) the point (i,j) and (c) the region where decisions are to be made $\mathcal{S}^{(i,j)}$. The decision at point (i,j) is made by minimizing the ML metric over all possible choices of bits in the region $\mathcal{S}^{(i,j)}$ as illustrated in the Figure 20. Compactly put,

$$\hat{a}_{i,j} = \arg\min_{a_{i,j}} \underbrace{\left[\min_{a\left(\mathcal{S}^{(i,j)}\right)} \Gamma\left(\hat{a}\left(\mathcal{P}^{(i,j)}\right), a_{i,j}, a\left(\mathcal{S}^{(i,j)}\right)\right)\right]}_{\mathrm{ML}_1(i,j)}, \tag{68}$$

where $\mathrm{ML}_1(i,j)$ the ML metric corresponding to the 2D Viterbi hard decisions i.e., the least among all the Γ metrics.

In 2D, it is non-trivial to identify the competing surfaces that merge at a state given by (i,j) and $a\left(\mathcal{S}^{(i,j)}\right)$. For each decision, we consider a single competing surface corresponding to a wrong decision at the current position (i,j). The corresponding ML metric for this second best path is given by:

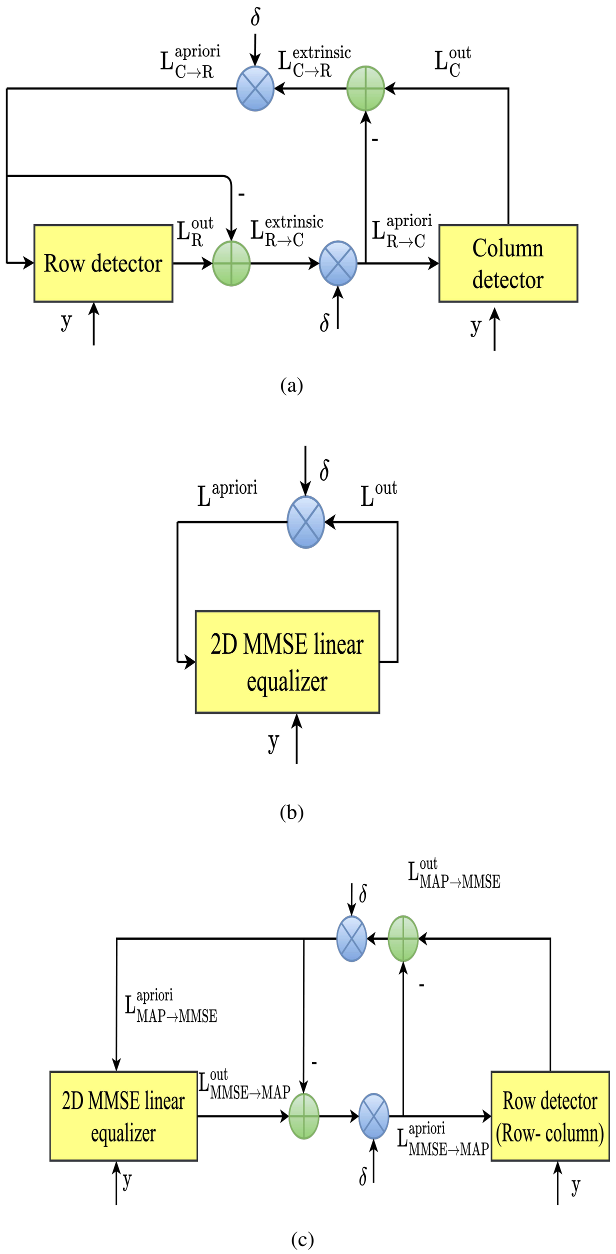
$$\mathrm{ML}_2(i,j) = \min_{a_{i,j} \neq \hat{a}_{i,j}} \left[\min_{a\left(\mathcal{S}^{(i,j)}\right)} \Gamma\left(\hat{a}\left(\mathcal{P}^{(i,j)}\right), a_{i,j}, a\left(\mathcal{S}^{(i,j)}\right)\right)\right]. \tag{69}$$

Using equations (68) and (69), we can obtain the soft decisions as

$$\mathrm{LLR}_{i,j} = \left(\mathrm{ML}_2(i,j) - \mathrm{ML}_1(i,j)\right)\hat{a}_{i,j}. \tag{70}$$

Figure 21 shows the signal chain for the bank of 2D DDNP filters along with the 2D signal detector. For a choice of the current state specific to each data pattern, ideal samples are computed, the noise is predicted and whitened using

[10] ML detection for 2D is NP-hard


δ
$L^{\text{apriori}}_{C\to R}$
$L^{\text{extrinsic}}_{C\to R}$
L^{out}_{C}
-
-
L^{out}_{R}
$L^{\text{extrinsic}}_{R\to C}$
$L^{\text{apriori}}_{R\to C}$
Row detector
Column detector
y
δ
y
δ
L^{apriori}
L^{out}
2D MMSE linear equalizer
y
$L^{\text{out}}_{\text{MAP}\to\text{MMSE}}$
δ
$L^{\text{apriori}}_{\text{MAP}\to\text{MMSE}}$
-
-
$L^{\text{out}}_{\text{MMSE}\to\text{MAP}}$
2D MMSE linear equalizer
$L^{\text{apriori}}_{\text{MMSE}\to\text{MAP}}$
Row detector (Row- column)
y
δ
y

(a)

(b)

(c)

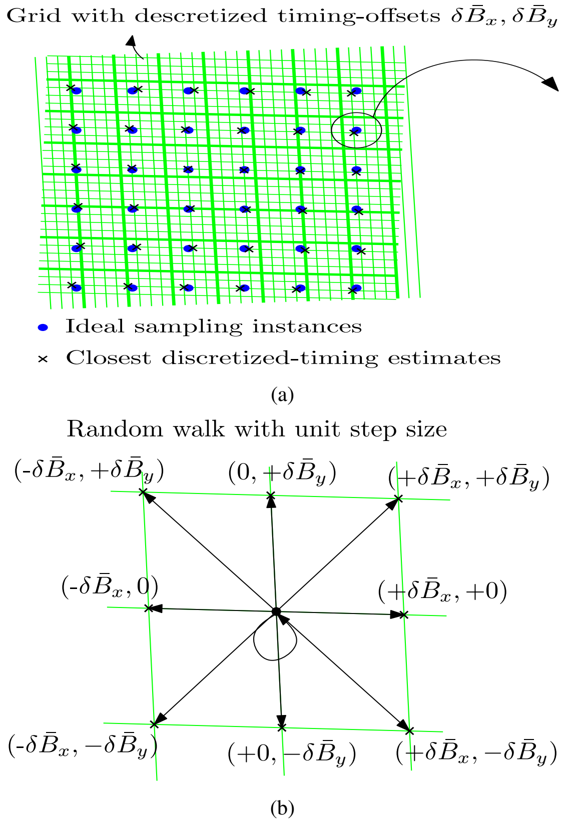
Figure 22. (a) Schematic of the multi-row/multi-column detector configured in a turbo setup. The detectors are based on the 1D BCJR extension. (b) Schematic of a 2D self-iterating soft-equalizer. (c) Schematic of the overall joint 2D self-iterating equalizer and the 2D detector coupled in a turbo setup.

the 2D prediction and whitening filters. The whitened noise along with its error variance forms the branch metric to be used with the 2D detector, mirroring the 1D DDNP-detection engine. However, the reader must note that the 2D engine uses feedback along a neighborhood of 2D samples since handling the entire array for inference on a bit is computationally prohibitive. For more details on the TDMR models and signal processing algorithms, the reader is referred to [106].

In [107], the authors developed an iterative multi-row/multi-column detector. The multi-row detector acts row-wise, while the multi-column detector acts column-wise. Both these detectors are trellis-based and use feedback information from neighboring pixels for making decisions, as shown in Figure 22(a). These coupled detectors exchange soft-information within a turbo setup towards obtaining *near-MAP* performance.

Further, in the same work, the authors developed a 2D self-iterating equalizer, whose schematic is shown in Figure 22(b). The 2D self-iterating soft equalizer brings in additional SNR gains. This engine is further coupled to the 2D row/column detector in a turbo setup in a fully iterative equalizer-detection setup. The architecture of the JTED engine is illustrated in Figure 22(c). The combined engine provided $\sim$ 8dB *significant* coded SNR gain compared to the uncoded 2D equalizer-detector system over 64×64 coded LDPC arrays. The reader must note that the ideas behind JTED can be explored for other combinations, such as using a 2D SOVA detector etc.

The GBP detector that we discussed for estimating MIR has also been used for 2D detection [108]. Though the GBP algorithm provides near ML performance, it is computationally intensive and is not scalable for handling large arrays that are practically relevant to 2D data storage. Recently, deep neural networks based architectures are being explored for the equalizers and detectors [109] [110] towards TDMR. Complexity and performance trade-offs can be realized using various JTED detector/equalizer configurations [107]. Further, the JTED engine is scalable for large arrays and could be used in practice. The turbo setup has also been used to obtain additional SNR gains when the 2D detector is coupled with a timing recovery algorithm. In the joint 2D timing recovery and signal detection scheme, [111], the timing errors are first


Grid with descretized timing-offsets $\delta\bar{B}_x, \delta\bar{B}_y$
● Ideal sampling instances
× Closest discretized-timing estimates
Random walk with unit step size
$(-\delta\bar{B}_x, +\delta\bar{B}_y)$
$(0, +\delta\bar{B}_y)$
$(+\delta\bar{B}_x, +\delta\bar{B}_y)$
$(-\delta\bar{B}_x, 0)$
$(+\delta\bar{B}_x, +0)$
$(-\delta\bar{B}_x, -\delta\bar{B}_y)$
$(+0, -\delta\bar{B}_y)$
$(+\delta\bar{B}_x, -\delta\bar{B}_y)$

(a)

(b)

Figure 23. (a) Timing offsets are discretized according to a desired level of timing error resolution. (b) The discrete set of timing offsets can be modeled as a 2D random walk process to be folded within the joint state space of the timing-recovery and detector algorithm.

discretized as shown in Figure 23, and the frequency offsets are estimated using a preamble. For each possible timing error, the ideal sample is obtained using an interpolation filter. The timing offsets are included within the definition of a trellis that operates over the joint state space of timing errors and the 2D channel ISI. This approach has the advantage to naturally handle correlated timing errors along with signal detection since it is within a Markov framework. The likelihood probability is computed over a local span of readback samples. The timing errors are estimated along with the bit decisions by maximizing the likelihood probability as derived in [111]. Due to the rastering, the timing error estimates of future samples may not available to estimate the timing error at a desired location for the first pass. However, these estimates are available and get refined over the subsequent iterations during the turbo iterative process.

It was reported that nearly 10% areal density gains can be realized using the iterative joint timing detector engine around the 1 Tb/in^2 regime with grain sizes $\sim$ 10 nm and bit sizes of 25x25 nm using the 2D SOVA compared to a standalone timing loop coupled to a 2D detector in an open-loop configuration for TDMR systems [111] comprising a 2D generalized partial response (GPR) equalization along the 2D SOVA with DDNP capability over the Voronoi media model.

FPGA implementations of a high-throughput 2D separable iterative soft-output Viterbi detector are also done in [112] by building over the algorithm proposed in [113]. These efforts are a step towards hardware realizations of the signal processing algorithms. During TMRC conferences, TDMR has been often dubbed as ‘terribly difficult magnetic recording’ due to several challenges in building advanced signal processing and coding techniques. By innovating native 2D algorithms meeting the challenges for handling equalization, timing recovery and signal detection along with ECCs, ADs in TDMR systems can be more than doubled.

We will now discuss coding techniques relevant to HDDs.

V. Coding Techniques for Recording Channels

A. *Modulation Coding for HDDs*

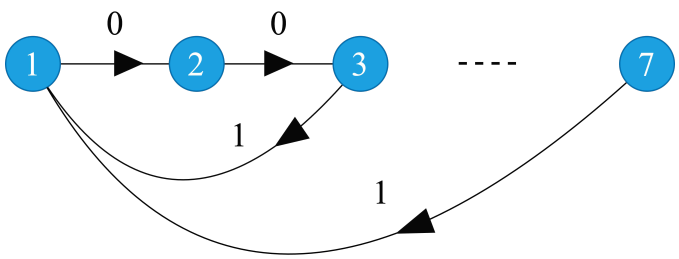


Figure 24. Constrained graph of RLL $(2,7)$ code used in early version of HDDs with peak detection circuits for signal detection.

HDDs using peak detection circuits for signal detection used binary (d,k) constrained codes. In early developments of channels efforts for HDDs, these codes played a *critical* role in dealing with channel ISI and timing issues. In the conventional setup, post error correction encoding and modulation encoding, such constrained codes were written on the magnetic medium. With the introduction of reverse order coding (ROC) by Bliss [114], the data is first modulation encoded and then error correction encoded using a systematic error correction code before being written onto the disk. This scheme avoids channel errors propagating through the modulation decoder in the conventional setup, particularly while dealing with high-rate codes with long block lengths. Since systematic encoders are used in practice, parity portion of the data payload may weakly satisfy the modulation constraints. This may not be much of a concern dealing with high-rate large block length codes typically used in magnetic recording. Binary (d,k) constrained codes have a minimum of d zeros and a maximum of k zeros between any two ones. These constrained sequences can be represented as digraphs. The combinatorial entropy or equivalently the noiseless capacity of a channel admitting such constrained sequences is given by $C = \log_2(\lambda_{\max})$, where $\lambda_{\max}$ is the largest eigenvalue of the adjacency matrix of the graph representing the (d,k) constraints. For a thorough treatment on constrained coding, the reader is referred to the book by Lind and Marcus [115]. For a comprehensive discussion on code construction methods, the reader is referred to the online lecture notes by Marcus, Roth and Siegel [116].

Figure 24 shows the example of a $(2,7)$ constrained code used in IBM drives that we discussed in the Introduction. A lot of research effort was done in the construction of efficient encoders and decoders for such (d,k) constrained codes. The state splitting algorithm, also called the ACH algorithm named after its inventors Adler, Coppersmith and Hassner [117] was one of the key techniques used for constructing these codes. The state splitting algorithm yields fixed-length encoders. Table II shows the encoder states for the fixed-length RLL $(1,7)$ code. There are totally 5 states in the state machine. Each state takes in 2 input bits of information and outputs 3 coded bits. The output bits and the corresponding outgoing states are described in each entry corresponding to each set of input bits and the input state.

Table II
RATE $\frac{2}{3}$ FIXED-LENGTH CODES FOR RLL $(1,7)$ [118] (SHANNON CAPACITY IS 0.6793.).

Input\State	1	2	3	4	5
00	101/4	100/4	001/4	010/4	000/4
01	101/3	100/3	001/3	010/3	000/3
10	101/5	100/2	001/5	010/2	000/2
11	100/5	100/1	010/5	010/1	000/1

There has also been work on variable-length encoders using ideas such as bit-stuffing etc. [119] [120]. Table III shows the encoding lookup-table for variable-length RLL $(2,7)$ code. For more details on modulation codes for recording channels, the reader is referred to the comprehensive survey papers by Marcus et al. [121] and Immink et al. [118].

Investigation into the minimum distance error events over PR channels [122] led to error event characterization for a variety of PR targets. Modulation codes were designed based on error event analysis to avoid bad sequences that led to errors at the output of the Viterbi detector within a PRML setup. The maximum transition run (MTR) code developed by Moon and Bricker [123] is one such example of a modulation

Table III
Rate $\frac{1}{2}$ variable-length encoder for RLL $(2, 7$ code (Shannon capacity is 0.5172) from [8].

Input	Output
10	0100
11	1000
000	000100
010	100100
011	001000
0010	00100100
0011	00001000

code that forbids/limits the length of occurrence of pairs of consecutive bit transitions. SNR gains were observed using high-rate MTR codes over PMRL channels. The trellis state of the PRML detector was equipped to handle both the modulation constraints as well as channel ISI.

Analysis of the power spectrum of (d, k) modulation codes was important [124] to quantify the bandwidth compression arising from written data on the medium post RLL encoding. Also, the power spectrum analysis was helpful to determine the amount of interference due to embedded servo and other timing information within the data signal, as well determine the crosstalk from adjacent tracks. Further, there were also notable works in the area of spectral-null codes [125] for applications in MR channels. Generalizations have been done for higher-order spectral-null codes in [126]. Techniques for the spectral analysis of modulation codes attuned to MR channels is useful for assessing the power allocation across such coded sequences.

(d, k)-constrained codes came with a code rate penalty, especially with $d \geq 1$. With powerful signal detectors for mitigating ISI, the d constraint that yielded low coding rates was no longer relevant for HDDs post peak detection schemes. Instead, high-rate $(0, k)$-constrained codes [127] were more relevant to PMR channels for timing recovery.

In the same spirit as 1D, there has been extensive research in the area of 2D constrained arrays for 2D data storage applications. Unlike the 1D case, computing the noiseless capacity for 2D constrained channels is a notoriously difficult problem. We do not yet have a generic formula for the exact analysis of the capacity of 2D constrained channels. Tight upper and lower analytical bounds are available for a few cases, such as the hard-square constraint [128], no-isolated-bit (n.i.b) constraint [129], checkerboard constraints [130]. Analytical bounds were also derived for the capacity of some 2D RLL M-ary constraints [131]. There has also been some works towards the construction of encoders and decoders for such 2D constrained arrays [132] [133].

Based on the empirical evidence from error events collected post signal detection using the Voronoi based channel model, it is found that the n.i.b. constraint is the dominant error event [32] [134]. To achieve the same storage density for a constrained coded system and an uncoded system, the rate loss due to the input constrained arrays must be compensated by scaling the bit size of the coded system by a factor of R_c, which is the rate of the constrained code. This reduction in bit size is justifiable if the gain in the performance due to 2D constrained coding is high enough to compensate for the effects of increased 2D ISI. Recently, the authors in [135] have developed high-rate two-dimensional lexicographically ordered constrained codes (TD-LOCO) for avoiding the square isolation pattern, useful for TDMR. As discussed earlier, modulation codes come handy along with powerful 2D signal detection algorithms when SNR performance and complexity trade-offs have to be assessed for TDMR system optimization towards high ADs. The choice of having a 2D constrained code eventually depends on the TDMR system constraints and choice of signal detectors.

With this, we now discuss error correction coding relevant to HDDs.

B. Error Correction Coding Specifications for HDDs

ECCs are critical for the successful working of HDDs. With powerful signal processing algorithms and optimization of the parameters attuned to the readback signals, under nominal SNR conditions, one can hope to achieve bit error rates $\sim 10^{-3}$ from the output of the signal detector so that the desired level of code failure rate can be achieved using ECCs. Unlike wireless channels, where the acknowledgment signals can be used for retransmission of failed packets, re-reads from the disk are costly. Also, unlike wireless channels that require frame error rates $\sim 10^{-6}$, HDDs require sector failure rates below 10^{-12}. Over the years, sector sizes have evolved from 512 bytes to 1KB and 4KB. With high coding rates, one can imagine the stringent requirements on the error rates to maintain data integrity in HDDs. Typically, PMR channels operate at coding rates ~ 0.9, while coding rates ~ 0.6 are relevant for TDMR channels. If the ECC decoder passes the syndrome test leading to a wrong codeword, it is undesirable. This error metric called the *miscorrection rate* must be below 10^{-22}. Miscorrection rate analysis often overlooked in the coding community is very important from a practical perspective. Furthermore, in the context of iterative decoders, it is desirable to not see any error floors above 10^{-12}.

In addition to all these requirements, ECCs for HDDs must handle a mixture of both random errors and burst errors since burst errors can occur due to thermal asperities [136], media defects [137] etc. The imposition of all these practical requirements from a code design perspective makes it challenging to design ECCs from a coding-theoretic perspective. Along with the coding requirements, one must be mindful of hardware implementations that imposes further constraints, such as high throughput, low decoding latencies, power and area-wise efficient coding architectures if the algorithm has to be translated to a working piece of Silicon.

While dealing with ECCs, we can think of two classes: (a) algebraic codes and (b) iterative codes. Algebraic codes have a rich mathematical structure with firm roots based on Galois theory over finite fields and rings. In general, one can compute bounds for guaranteed error correction ability for algebraic codes based on the code parameters. This makes it tractable for predicting the performance of ECCs for MR channels if we use algebraic codes. Examples of algebraic codes popularly used in data storage devices are Bose Chaudhuri and Hocquenghem (BCH) codes, Reed Solomon (RS) codes etc. On the other hand, iterative codes are based on soft-decision decoders. Iterative codes provide higher SNR gains (> 3 dB) in the waterfall region. However, they are prone to error floors [138] that could be mitigated by careful design of codes and control over the quantization parameters.

In this section, we will discuss ECCs that are an integral part of HDDs.

C. Algebraic Codes: The Reed Solomon Case

Before the advent of soft-information driven LDPC-based read channels, RS codes [139] were mainly used in HDDs since they could correct burst errors more efficiently than interleaved binary codewords. RS decoders were mainly based on hard decisions. Soft-decision based RS codes with list-decoding was tried. However, several practical issues made it difficult for incorporating these codes within HDDs. First, identifying the flip list was one of the major problems for large block lengths. Second, the SNR gain obtained using soft-decision- based RS codes was just a fraction of dB compared to the huge implementation cost in terms of hardware complexity. Last, coupling a soft-decision RS decoder with an LDPC code was catastrophic since the number of errors from failure of LDPC codes was beyond the ECC ability of soft-decision RS codes, leading to diminishing SNR returns from the combined system. Hence, soft-decision-based RS codes were not feasible for incorporation into HDDs. However, with carefully designed LDPC codes with excellent ECC performance (no errors below 10^{-12}), one can design RS codes as an outer error detecting code to provide guaranteed miscorrection rates. With this background, we will discuss RS codes, useful for HDDs.

RS codes are a class of non-binary symbol-error correcting cyclic codes [139]. An (n, k, t) with code length n, message length k and error correcting capability t attains the Singleton bound with equality since the distance is $d = n - k + 1$. Grobner basis spans a Reed Solomon code space. The parity check matrix of a $t-$error correcting Reed Solomon code is a $(n - k) \times n$ matrix $H := [h_{ij}] = [\alpha^{ij}]$ $1 \leq i \leq 2t$, $0 \leq j \leq n - 1$. Over $GF(2)$, $n = 2^m - 1$. In general, one could define RS code over $GF(q)$. However, from a hardware perspective, the binary field is preferred for obvious reasons. Unlike binary error correction, where it is sufficient to determine just the error locations and flip the bits, for the non-binary case, we need to identify both the error locations and evaluate the error values at these locations. Though there are other techniques such as the modified Euclidean algorithm for finding the error location polynomial, the Berlekamp-Massey (BM) algorithm is preferred since it is efficient from an implementation perspective in terms of complexity. The error evaluation is based on Forney's algorithm. We will briefly describe the steps in the error correction procedure for the sake of completeness and comment on the hardware implementation.

1) Error correction procedure: We will follow the notations introduced in [139] and [141] for describing the error correction procedure. A codeword $c(x)$ corrupted by an additive error $e(x)$ results in $r(x)$. Let υ denotes the number of errors in $e(x)$ having the form

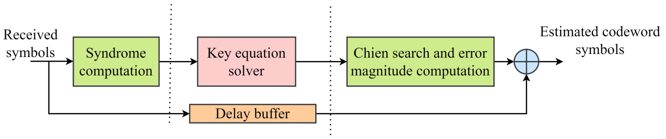

Received symbols
Syndrome computation
Key equation solver
Chien search and error magnitude computation
Estimated codeword symbols
Delay buffer

Figure 25. Conventional 3-stage RS decoder. Adapted from [140].

$$e(x) = \sum_{i=1}^{\upsilon} y_i x^i, \ \ 0 \leq i \leq n - 1, \tag{71}$$

where y_i is the error magnitude at error location i.
The syndromes are calculated as

$$S_j = r(\alpha^j) = e(\alpha^j) = \sum_{i=1}^{\upsilon} Y_i X_i^j, \ \ 1 \leq j \leq 2t, \tag{72}$$

where $Y_i = y_i$ and $X_i = \alpha^i$. The aim is to solve the above $2t$ equations to get the pairs (X_i, Y_i). Defining the error locator polynomial $\Lambda(x)$ given by

$$\prod_{i=1}^{\upsilon}(1 + X_i x) = \Lambda_0 + \Lambda_1 x + \cdots + \Lambda_{\upsilon-1}x^{\upsilon-1} + \Lambda_\upsilon x^\upsilon. \tag{73}$$

The X_i values are evaluated using inverse roots of the above equation. Given the values X_i, the linear system of equations (72) in Y_i can be solved. The error correction process involves a 4-step procedure outlined as follows:

Table IV
BERLEKAMP'S ITERATIVE PROCEDURE FOR FINDING THE ERROR LOCATOR POLYNOMIAL $\Lambda(x)$ OF A RS CODE [139].

μ	$\Lambda^{(\mu)}(x)$	d_μ	l_μ	$\mu - l_\mu$
-1	1	1	0	-1
0	1	S_1	0	0
1	$1 - S_1 x$			
⋮				
2t				

Step 1: Calculation of syndromes S_j:

Syndromes can be evaluated according to equation (72) from $r(x)$.

Step 2: Calculation of Λ_i from S_j: (Berlekamp) [139]

We can compute $\Lambda(x)$ iteratively in $2t$ steps. Let $\Lambda^{(\mu)}(x)$ denote the error locator polynomial at μ^{th} step of the iteration. To find $\Lambda(x)$ iteratively, we start with the initialized Table IV shown and proceed to fill the rest of the table entries. Let l_μ be the degree of $\Lambda^{(\mu)}(x)$. Assuming that we have filled out the μ^{th} row, we find $(\mu + 1)^{\text{st}}$ row using the procedure shown below

1) If the discrepancy $d_\mu = \sum_{i=0}^{l_\mu} S_{\mu+1-i}\Lambda_i^{(\mu)} = 0$, then $\Lambda^{(\mu+1)}(x) = \Lambda^{(\mu)}(x)$ and $l_{\mu+1} = l_\mu$.

2) If $d_\mu \neq 0$, then we search another row ρ prior to the μ^{th} row where $d_\rho \neq 0$ and the number $\rho - l_\rho$ in the last column of Table I having the largest value. $\Lambda^{(\mu+1)}(x)$ and $l_{\mu+1}$ are updated as

$$\Lambda^{(\mu+1)}(x) = \Lambda^{(\mu)}(x) - d_\mu d_\rho^{-1} x^{\mu-\rho} \Lambda^{\rho}(x), \qquad (74)$$

$$l_{\mu+1} = \max[l_\mu, l_\rho + \mu - \rho]. \qquad (75)$$

Step 3: Calculation of X_i from Λ_i: (Chien's search) [139]

If α^{-i} is a root of $\Lambda(x)$, the error is present at location i.

Step 4: Calculation of Y_i: (Forney's formula) [139]

To evaluate the error magnitudes, we use Forney's formula

$$Y_i = -\frac{\Omega({X_i}^{-1})}{\Lambda'({X_i}^{-1})}, \qquad (76)$$

where $\Omega(x) = S_1 + (S_2 + \Lambda_1 S_1)x + (S_3 + \Lambda_1 S_2 + \Lambda_2 S_1)x^2 + \cdots + (S_\upsilon + \Lambda_1 S_{\upsilon-1} + \cdots + \Lambda_{\upsilon-1} S_1)x^{\upsilon-1}$. $e(x)$ obtained from X_i and Y_i is added to $r(x)$ to get the decoded codeword polynomial $c(x)$.

The implementation of RS decoding algorithm is equally important. To achieve higher throughput, one needs to carefully pipeline the architecture. The popular three-stage pipelined RS decoder [142] is shown in Figure 25. Since the overall throughput is decided by the slowest pipelined stage, to increase the throughput with efficient area utilization, each pipelined stage should complete its computations in about the same amount of time. Further, parallelism can be employed to adjust the number of clock cycles required for the syndrome computation (SC) and the Chien search, error magnitude computation (CSEMC) stages. The size of the delay buffer used for buffering received symbols depends on the latency of the decoder.

Figures 26(a) and 26(b) show the detailed architectures for the SC and key equation solver (KES) stage, and the error evaluation unit is shown in Figure 27. Use of parallelism ensures that the SC unit takes exactly $2t$ cycles, which is same as the KES stage in order to maximize the throughput. With this parallelism, one can merge the SC and KES stages into a single stage without affecting the throughput.

The *two-stage* pipelined RS decoder shown in Figure 28 has significantly less pipeline registers and delay buffers. To maximize the throughput and minimize the latency, one can make use of the efficient architecture for error locator and magnitude computation [142] coupled with the J-parallel Chien search architecture [143] to find the error locations and the corresponding error magnitudes simultaneously. This can be accomplished in $t + \frac{n}{J}$ cycles. All these ideas are important details for building a practical RS decoder.

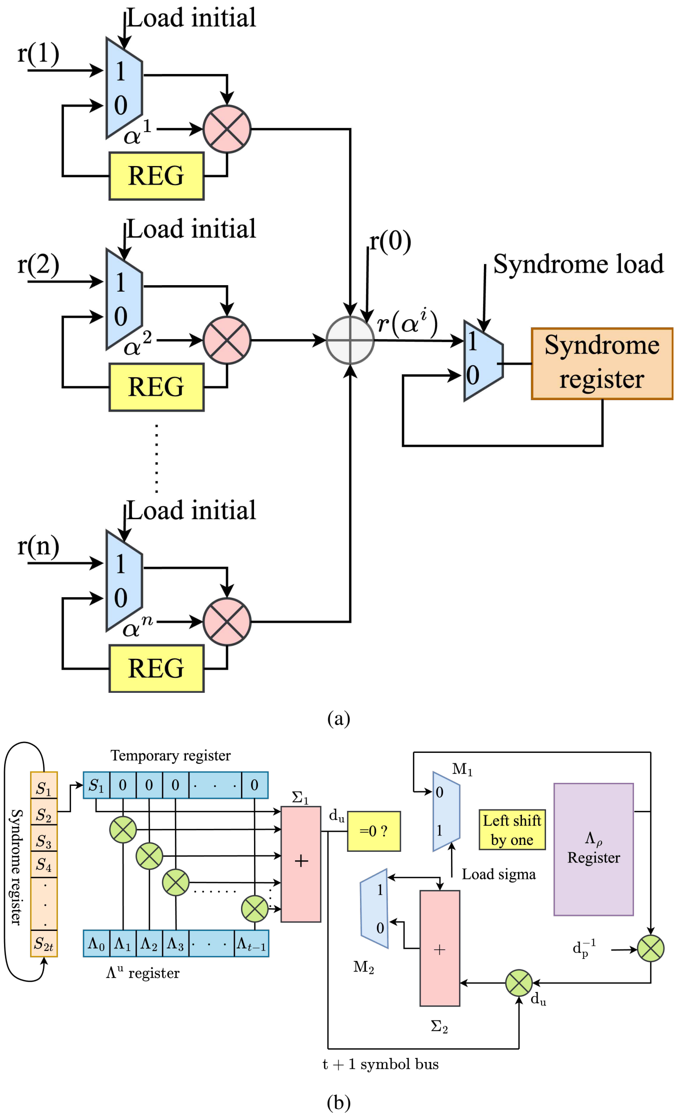


Figure 26. (a) Parallel architecture for syndrome computation (SC) that computes $2t$ syndromes in exactly $2t$ cycles. (b) Low complexity key equation solver (KES) architecture to compute the error locator polynomial $\Lambda(x)$ based on Berlekamp's iterative procedure in exactly $2t$ cycles. Adapted from [140].

With the two-stage pipelined approach, the KES stage needs to wait for only two extra cycles with respect to the SC stage to initialize the BM iterative procedure. Hence, the throughput achieved by the architecture in Figure 28 is almost the same as the three-stage pipelined decoder with significant improvement in latency. The latencies with the two-stage pipelined design is $2t$ cycles less compared to the three-stage design and is efficient compared to the design in [144]. For the RS $(255, 239)$ decoder over $GF\left(2^8\right)$, with $J = 30$, we could get a throughput of 24 Gbps. There is a trade-off between area and throughput. For example, one can save the Silicon area by choosing $J = 10$ at the expense of throughput that reduces to 12 Gbps. For more technical details, the reader is referred to [140].

We now discuss the design of LDPC codes and decoder architectures for HDDs.

D. Iterative Codes: 1D LDPC Codes

Initial efforts in trying to use turbo codes based on constituent convolution codes for HDD read channels during the mid 1990s met with limited success since error floors were observed, despite impressive SNR gains in the waterfall region. HDD channels required the design of efficient linear block codes for large block lengths along with scalable decoding

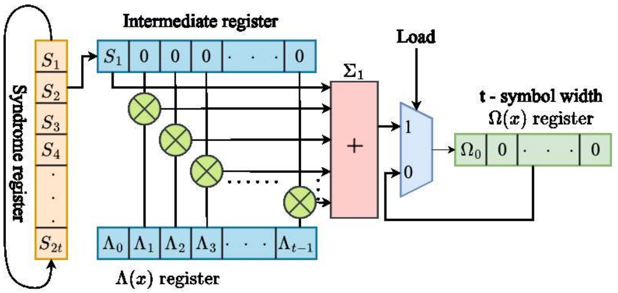


Figure 27. Error evaluator architecture to compute $\Omega(x)$ in exactly t cycles. Adapted from [140].

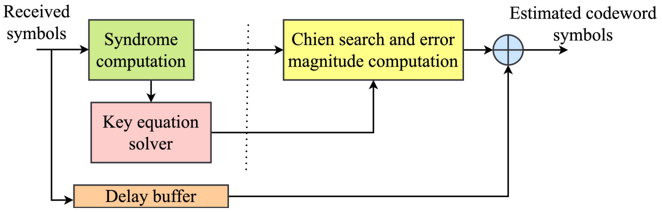
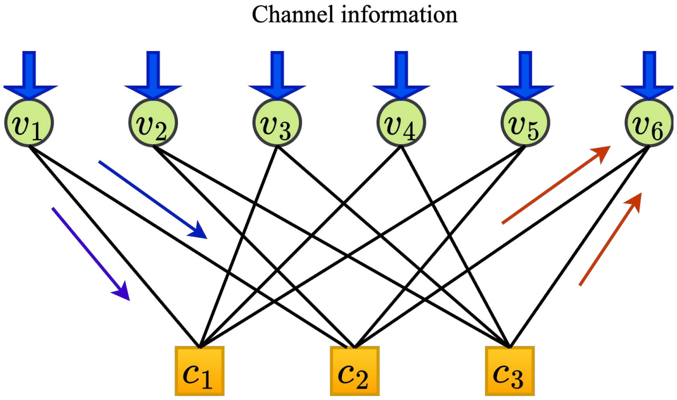

Received symbols
Syndrome computation
Chien search and error magnitude computation
Estimated codeword symbols
Key equation solver
Delay buffer

Figure 28. 2-stage pipelined RS decoder. The syndrome computing engine and the KES are merged into a single stage. Adapted from [140].

algorithms with an inherently parallel structure amenable to hardware implementation. With pioneering research efforts on LDPC codes by Tanner [145], Mackay [146], Urbanke and Richardson [147], [148] and many others, LDPC codes originally conceived by Gallager [149] in his seminal PhD work in 1960 were resurrected almost 3 decades later. However, significant additional efforts were required for deploying these codes in HDDs. We shall briefly describe LDPC codes.

$$\mathrm{H} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

(a)



(b)

Figure 29. Example of a sparse parity check matrix and its equivalent Tanner graph.

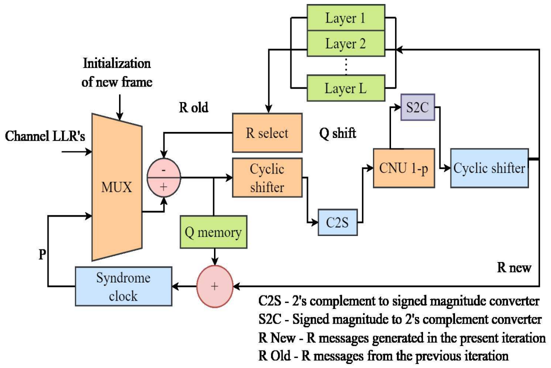
The parity check matrix of an LDPC code can be bijectively mapped to a Tanner graph, which is a bipartite graph consisting of a set of variable nodes and check nodes representing the columns and rows of a parity check matrix respectively. A ‘1’ in a parity check matrix implies a connection between the corresponding check node and variable node as illustrated in Figure 29. This representation helps with message passing from the variable nodes to check nodes towards improving the quality of soft decisions during the decoding process.

Through density evolution techniques [147], the degree distribution of LDPC codes can be optimized, mindful of the hardware constraints and performance specifications required for HDDs. Regular LDPC codes, i.e., with the same row weight and column weight, are preferred from hardware perspective since irregular codes require additional control logic, leading to lower latencies.

With an estimate on the MIR for MR channels, we require the construction of LDPC codes with excellent performance in the waterfall region and without noticeable error floors, meeting the code design specifications. Since error exponents scale with large lengths and MR channels require data payloads with larger lengths that are scalable over a range of coding rates, we need a family of LDPC codes that work well in practice. The quasi-cyclic family of LDPC codes [150] [151] naturally fits this requirement. QC LDPC codes, which are special cases of proto-graph-based LDPC codes [152] can be constructed by tiling permutation matrices $p \times p$. For a regular LDPC code of block length n and rate R, the column weight and row weights are respectively $\frac{n(1-R)}{p}$ and $\frac{n}{p}$. Good upper bounds on the minimum distance of these codes can be obtained through semi-analytical means [153].

QC LDPC codes are most suitable for HDD channels [154] [155] due to the following reasons: (a) QC codes provide flexibility in terms of adjustable code lengths and code rates suitable for a wide range of data payloads and format sizes under various channel conditions without requiring to store several parity check matrices within on-chip memory. (b) Since QC codes are based on tiling permutation matrices, it provides the needed parallelism while decoding blocks of rows and columns. (c) Efficient encoding structures are possible since the generator matrices obtained from QC codes can also be realized as a tiling of circulant matrices [156]. Points (a)-(c) are very important from a hardware perspective due to storage requirements. (d) It is possible to construct QC codes devoid of short cycles and harmful structures such as trapping sets [157]–[160] meeting the error floor specifications. (e) The code geometry is flexible for optimizing the degree distribution in the design of the parity check matrix.

Though the original sum product algorithm (SPA) for decoding LDPC codoes is not hardware efficient. The min-sum algorithm [161] is the accepted standard for hardware since it avoids the use of bulky lookup tables for handling complex arithmetic. As in any hardware design, there exists design architectural trade-offs, compromising a slight degradation in SNR performance if other criteria such as speed, area, throughput and latencies must be optimized for hardware. Since area is one of the important considerations, the layered min-sum algorithm [162] is the preferred choice for hardware implementation [163]. Unlike the parallel update in the non-layered case, the layered engine processes block rows serially, thus achieving a low area. Also, the layered MSA provides better signal to noise ratio (SNR) performance than its non-layered counterpart since more recent updates over a prior block of processed data are made available to the next block row for decoding. This inherent non-linearity brings additional gains. Some of these ideas are what it takes to bring LDPC decoders to practice.



C2S - 2's complement to signed magnitude converter
S2C - Signed magnitude to 2's complement converter
R New - R messages generated in the present iteration
R Old - R messages from the previous iteration

Figure 30. Architecture of the layered decoder. Adapted from [140].

Figure 30 shows the schematic of a layered architecture. The architecture consists of check node units (CNU) driven by barrel shifters[11] towards realizing the intended parallelism. There are adders, subtractors, and block random access memories (BRAM) for storing the intermediate overall reliability information P, check node messages Q and variable node messages R. The CNU array comprises p parallel CNU units that compute the partial state for each row producing the R messages in block serial form. The MUX is required to supply new LLRs to the decoder when the decoder has corrected the previous frame or the maximum iteration limit is reached. Normally, signed to 2's complement and 2's complement to signed converters are required before and after the CNU in case of uniform quantization. One could work with non-uniform quantizers based on channel conditions [140] to realize area-efficient designs. The initialization for decoding a sector of data is handled by a MUX in front of the cyclic shifter. In the beginning, the output of the R select unit is set to a zero vector. The P messages are computed by adding the delayed version of the Q messages stored in a BRAM to the R messages. The R messages are then stored in a R message BRAM, which would be used in subsequent iterations. The next block row is now ready to be processed as the P messages are directed by the MUX to the subtractor. Subsequent next rows are processed as explained before. Syndromes are computed efficiently in hardware. The process goes on till the sector has been corrected, or a maximum iteration limit has been reached [12].

The check node unit [163] in Figure 31 emulates the operations at the check node on a Tanner graph. It sends back the minimum of the values received from a certain variable node, without accounting for the variable node. The check node unit consists of a minimum value N_1 and a second minimum value N_2 finder, a partial state that stores N_1 and N_2 temporarily and updates them on each clock cycle, a final state which stores the final N_1 and N_2 value, and a sign processing unit which takes care of the sign of the LLR to be sent. Incoming variable messages are compared to two up-to-date least minimum numbers to generate new partial state. In this state, we have N_1 (first minimum value), N_2 (second minimum value) and the index of N_1. The final state is achieved after all the messages have been received. The R selector then assigns one of these 2 values (N_1 and N_2) based on the index of N_1 and sign of all the R messages generated by the XOR logic.

It has been more than a decade and half where such LDPC decoders based on layered architectures achieving high bit throughput rates in excess of 4Gbps and consuming $\sim 1W$ of power over $512-$byte format sizes were part of read channel chips in HDDs. Today's format sizes are on the order of 4KB and almost near-capacity. However, the QC LDPC framework is still valid for such large format sizes. The reader might wonder how such low error rates are simulated. It must be noted that computer simulations can only reach around 10^{-7}. If error rates beyond this have to be reached, we need high speed FPGA circuits to process the sectors before getting them to ASICs. Using arrays of FPGAs, it is possible to ascertain if the codes provide error rates below 10^{-12}. The lack of theory, rather the theoretical difficulty to predict the *exact* performance of individual codes is circumvented through simulations using FPGAs in practical systems.

The reader might ponder about using non-binary codes for HDDs, motivated by RS constructions that can correct burst errors. There has been an extensive study of non-binary LDPC codes and decoding algorithms [164]–[168]. Also, there are some VLSI implementations of such non-binary LDPC codes [169]–[171]. There is also the layered min-sum version for non-binary LDPC codes [172]. In general, decoders for non-binary codes are computationally far more intensive than the binary case. One can always realize equivalent binary decoders using multistage decoding, re-using the hardware for binary decoders. Also, one will need soft-decision detectors that are compatible with such non-binary decoders in the turbo-equalization setup. These practical difficulties make non-binary LDPCs less attractive for HDD read channels from a hardware perspective.

The concept of spatially-coupled (SC) codes is based on periodic time-varying LDPC convolutional codes, originally proposed by Felstrom and Zigangirov [173]. These codes provide resilience to burst errors while having less overhead than individual block codes with interleavers since different copies of the base code are partitioned into component matrices and connected together. In some sense, interleaving happens naturally within the code construction. Windowed decoding for SC-LDPC codes [174] brings improved latencies, useful for streaming applications. Multidimensional-SC LDPCs based on quasicyclic designs were proposed in [175] by coupling 1D SC-LDPC codes through rewiring the connections across the SC blocks without adding any extra variable nodes/check nodes. By optimizing the number of small cycles BER im-

[11] The barrel shifters are specific to QC code implementations.

[12] The maximum number of iterations is decided based on the hardware constraints and decoding performance

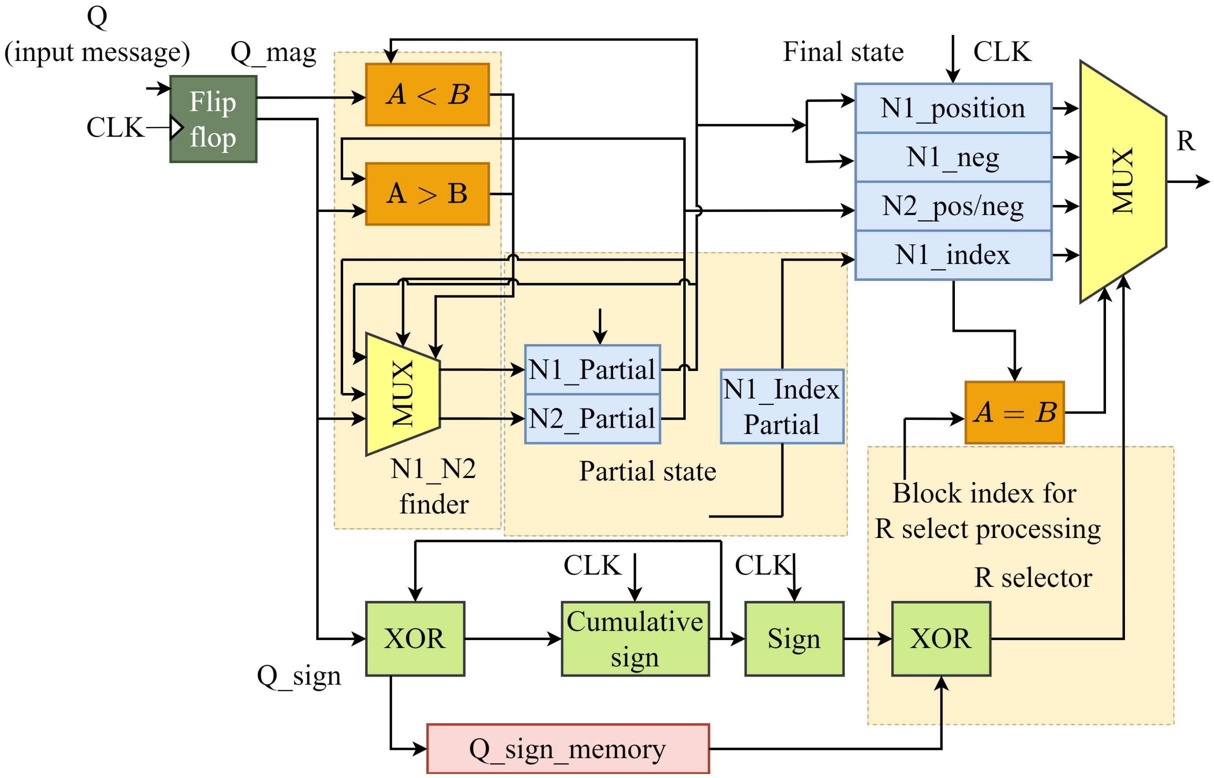


Figure 31. Architecture of the CNU. Q and R represent the incoming and outgoing messages respectively. Adapted from [140].

provements are seen with the modified code designs [176] [175]. These codes were also shown to have better resilience to burst errors than their 1D counterparts. More recently, the authors in [177] have proposed a probabilistic framework for designing *near-optimal* SC codes with large memory, useful to practice. In [178], the authors considered interleaved SC codes to provide resilience to SNR variations in the MR channel.

Having interleavers and deinterleavers within a conventional turbo-equalization setup can incur significant latencies. The reader must note that carefully constructed interleavers can be embedded within such LDPC codes [179], amenable to the layered decoding architecture with some modifications. These designs add zero latencies i.e., without interleavers/deinterleavers in the turbo loop, which can be significant when block lengths are large. We remark that along with the core algorithms, several system-level innovations are an integral part of engineering practice towards a working prototype/product. This subsection summarizes 1D LDPCs for HDDs.

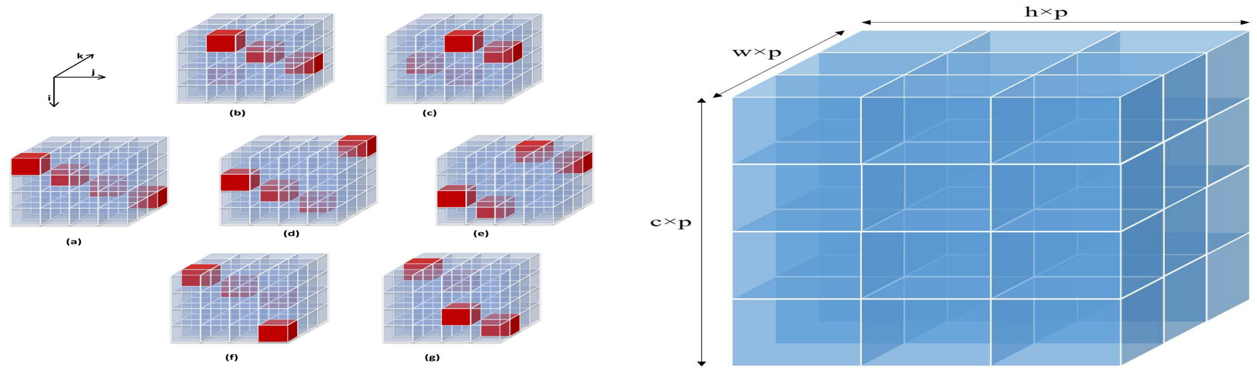
We now discuss the design of 2D codes suitable for TDMR systems.

E. Native 2D Codes for TDMR

Iterative error correction codes like the 2D LDPC [180] provide ability to correct large 2D cluster errors, circumventing the need for 1D LDPC with interleavers for handling 2D arrays. This motivates the design of codes and circuit architectures for decoding native 2D LDPC codes applicable to TDMR channels.

Though Cassuto and Shokrollahi [181] proved existential results of 2D LDPC codes, they did not explicitly provide the construction for correcting 2D burst erasures. Matcha et al. [180] proposed constructions of native 2D LDPC codes capable of correcting large 2D burst erasures. Their construction involved stacking $c \times h \times w$ permutation tensors of size $p \times p \times p$ along the i, j, k axes. These permutation tensors were constructed by applying a combination of shifts along j and k axes on an identity tensor. The shifts were chosen in such a way that the code had a burst erasure correction capability of at least $p \times p$. A variant of the code used in [180] was further developed by Kamabe and Lu [182] with improved burst erasure correction capability. Recently, 2D LDPC codes were designed using a particular choice of the shifts within the tensors leading to graphs devoid of short cycles i.e., of length 4 for carefully chosen code parameters, bringing significant coding gains [183] over [180]. Similar to the code in [183], the proposed code construction has provably burst erasure correction capability of at least $p \times p$. Hardware architectures for a 2D LDPC decoder based on the non-layered min-sum algorithm (NL-MSA) were also been proposed in [183].

Each codeword in a 2D LDPC code is a 2D array of bits and the parity check tensor for a 2D LDPC code has 2D arrays stacked on top of each other, maintaining orthogonality with the code space. Each horizontal layer in the parity check tensor represents a single parity-check equation, analogous to the parity check matrix for a 1D LDPC code. Parity check tensors can be obtained by stacking 3D permutation tensors in a 3D fashion [180]. The position (i, j, k) represents the bit (j, k) in the i^{th} horizontal layer of the parity-check tensor. $I_{3\text{-}D}$ is chosen to be a identity tensor of size $p \times p \times p$, as



(a) (b)

Figure 32. (a) Identity tensor $I_{3\text{-}D}$ and its various shifts. (b) Stacking of $c \times h \times w$ permutation tensors along i, j, k directions to form the parity check tensor $H_{2\text{-}D}$. Each small cube represents a permutation tensor obtained from cyclic shifts I-shift and J-shift on identity tensor $I_{3\text{-}D}$. Adapted from [183].

shown below:

$$I_{3\text{-}D} = [I_{i,j,k}]_{i,j,k=1}^{p},\ I_{i,j,k} = \begin{cases} 1 & i = j = k \\ 0 & \text{otherwise.} \end{cases} \tag{77}$$

The tensor $I_{3\text{-}D}$ can be permuted in three directions of the co-ordinate axes i, j, k. Let $P : \{1, 2, \cdots, p\} \to \{1, 2, \cdots, p\}$, $Q : \{1, 2, \cdots, p\} \to \{1, 2, \cdots, p\}$ and $R : \{1, 2, \cdots, p\} \to \{1, 2, \cdots, p\}$ be the three permutation operations defined on a tensor $T = [T_{i,j,k}]_{i,j,k=1}^{p}$ as shown below:

$$P(T) = [T_{i,P(j),k}]_{i,j,k=1}^{p} \tag{78a}$$
$$Q(T) = [T_{Q(i),j,k}]_{i,j,k=1}^{p} \tag{78b}$$
$$R(T) = [T_{i,j,R(k)}]_{i,j,k=1}^{p}. \tag{78c}$$

Similar to the permutation matrices in 1D, we can choose P, Q and R to be circular shifts given by

$$P(i) = Q(i) = R(i) = \begin{cases} p & i = 1 \\ i - 1 & \text{otherwise.} \end{cases} \tag{79}$$

We denote the circular shifts P, Q and R as J-shift, I-shift and K-shift respectively. These shifts have been illustrated for an identity tensor $I_{3\text{-}D}$ of size $4 \times 4 \times 4$ in Figure 32(a).

H_{2D} can be obtained by stacking $c \times h \times w$ cubes, each of size $p \times p \times p$ along i, j and k directions respectively as shown in Figure 32(b). A possible way to choose the $(i, j, k)^{\text{th}}$ cube is to allow shifts of the form $P^{a(i,j,k)} \circ Q^{b(i,j,k)}(I_{3\text{-}D})$ with $a(i, j, k)$ and $b(i, j, k)$ chosen as follows:

$$\begin{aligned} a(i,j,k) &= \operatorname{mod}\left((i-1) + \left\lfloor \frac{i-1}{p} \right\rfloor ((j-1)w + k), p \right) \\ b(i,j,k) &= \operatorname{mod}\left(\left\lfloor \frac{i-1}{p} \right\rfloor (j-1), p \right). \end{aligned} \tag{80}$$

The code size is $hp \times wp$ with a parity check tensor H_{2D} of dimensions $cp \times hp \times wp$. The rate of the code is $(hwp - c)/hwp$. If c is a multiple of p, the choice of shifts in equation (80) produces a parity check tensor having uniform column weight c/p. Each permutation tensor in H_{2D} contributes to a row weight w_r of exactly 1. Thus the row weight of the constructed code is hw. The above construction has the following properties [183]:

1) If p is sufficiently large, the girth of the code is greater than 4. For $(h-1)(c/p-1) < p$ and $(w-1)(c/p-1) < p$, the girth of the code is greater than 4. For example, for $p = 16$ and $c = 64$, $h = 3$, $w = 3$, we obtain a girth greater than 4. Due to the absence of short cycles of length 4 in the constructed code, we observe good error correction performance.
2) The construction is able to correct 2D burst erasures of size at least $p \times p$.

Under the standard AWGN channel[13], the performance of 2D codes vs. 1D codes should not matter. However, when bursts and erasures are introduced, the performance of native 2D is superior to 1D codes since 1D codes are tailored for it. Further, as we discussed earlier, 1D codes with interleavers are not a preferred choice due to decoding latencies etc. With this in mind, we now discuss the performance of 2D LDPC codes.

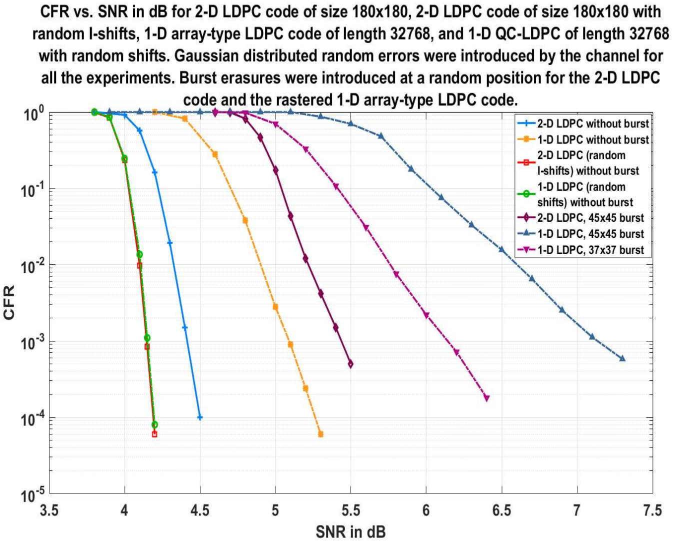


Figure 33. Performance of 2D and 1D LDPC codes over random errors and burst erasures. Adapted from [184].

Figure 33 shows the performance of 2D LDPC codes constructed with parameters $p = 45$, $c = 315$, $h = 4$, and

[13] The performance evaluation over AWGN is reasonable since post equalization and DDNP detection, the equivalent channel is close to the AWGN case.

$w = 4$ over arrays of size $180 \times 180 \sim 4\text{KB}$. The parity check tensor with dimensions $14175 \times 180 \times 180$ had a column weight $w_c = 7$, and row weight $w_r = 16$, and code rate = 0.5625. With maximum iteration limit set to 5, SNR performance of the 2D LDPC code is ~ 0.7 dB better than its counterpart 1D array-type LDPC code at a code failure rate (CFR) of 10^{-3}. The reader must note that the drop per dB in the waterfall region for the 2D LDPC code is also significantly steeper. However, a 1D QC-LDPC code with random shifts for permutation matrices performs ~ 0.25 dB better than the proposed 2D LDPC code since it was difficult to incorporate random shifts through the sampling and rejection process for the 2D case. From Fig. 33, we also observe that a 2D LDPC code constructed using random I-shifts devoid of short cycles has almost the same performance as the 1D QC-LDPC code with random shifts.

With a 45×45 burst erasure at a random position along with Gaussian distributed random errors, the 1D LDPC is ~ 1 dB inferior in SNR performance compared to the 2D case, seen at a CFR of 10^{-3}. The drop per dB in the waterfall region is also significantly worse in presence of the 45×45 burst erasure. This is because the 1D LDPC code is not suitable for handling a 2D burst erasure, since the 2D burst erasure on the rastered 1D LDPC codes is equivalent to numerous non-contiguous smaller 1D burst erasures, which the 1D array-type LDPC code construction is not built to handle. In the presence of a 37×37 burst erasure, there was a ~ 1 dB SNR performance degradation at a CFR of 10^{-3}. These discussions clearly point out the importance of constructing native 2D codes for TDMR channels. These designs can be further improved.

One can also build decoders for 2D LDPCs. The interested reader is referred to the recent paper [183].

Before we end this subsection, we would like to comment on 2D algebraic codes for TDMR, similar to the RS codes that we discussed earlier. Algebraic code constructions are helpful for guaranteed ECC ability for 2D burst errors of small, pre-defined error shapes [185] [186]. However, the SNR gains are expected to be inferior compared to soft-decision 2D LDPC codes. In the same spirit of 1D hard-decision RS codes, such 2D algebraic codes could be used for correcting cluster errors with guaranteed error correction over pre-defined errors shapes based on error events collected from 2D detectors, or as an outer error detecting code post 2D LDPC decoding. Correction of arbitrary $t \times t$ bursts using $n \times n$ 2D BCH code were studied by Madhusudhana and Siddiqi [187] using improved Blahut's algorithm (IBA-I) [188]. Recently, modified IBA-I decoding algorithms were proposed towards efficient hardware architectures for 2D BCH codes in [189]. For more details on the hardware architectures for 2D codes, the reader is referred to [184]. In general, construction of algebraic codes for handling multiple cluster errors over 2D arrays is a difficult problem. For comprehensive details on the tools for constructing codes over curves and planes, the reader is referred to the book by Blahut [190].

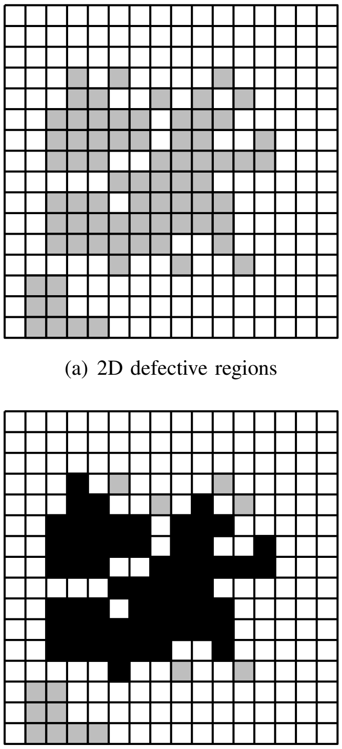
F. Handling Media Defects and Other Channel Architecture Considerations

The design of ECCs for storage channels must include the ability to resolve burst erasures due to media defects [136] that could be deep or shallow. Usually, deep defects are spread over a smaller number of bits, while shallow defects are more wide spread [137]. The signal energy over defective regions can be one of the cues to identify erasure locations. Thus, the amount of ECC power required for handling burst erasures is specific to head/media combinations. In earlier versions of the track-based magnetic recording, burst erasures were overcome by using RS codes in conjunction with inner iterative codes. Initial works on post-ECC modeling techniques to decide the t-level error correction power for an RS code was based on the block multinomial model [191]. Since the block multinomial model cannot handle correlated errors that fall outside the code boundaries, subsequent modeling improvements were based on hidden Markov models (HMMs) [192] [193] for deciding the amount of ECC required for HDDs. Using data collected from critically failing drives, the t-level error correction power for an RS code was decided for the given media conditions [193].

Structured LPDC codes like the QC codes that exhibit excellent performance in the waterfall and error floor regions can also provide good erasure correction ability since erasures over two consecutive tiles of permutation matrices can resolve the burst. Carefully designed interleavers can also enhance the burst erasure capability of the LDPC code. Fossorier [194] provided construction of a (n, k) LDPC code that can correct bursts up to a length of $n - k - 1$, achieving the Roger bound. Construction of LDPC codes for iteratively correcting burst erasures using the belief propagation algorithm by identifying trapping sets has also been investigated in [195], [196].

The identification of burst erasures is an important step towards error correction. In fact, this is one of the specifications while designing read channels for HDDs. Traditional approaches for defect identification include using RLL codes, or a full response reequalization [197] for defect identification. There are many signal processing cues one could garner by observing defective regions from empirical data towards flagging the erasure locations. These include (a) onset of low signal energy over a defective region, (b) low LLR values observed at the output of a signal detector, and (c) signature analysis from frequently occurring state transitions within the trellis states over the defective region [198].

In the context of TDMR, Figure 34(a) shows how cluster errors appear as 2D media defects. Defect detection for TDMR channels poses significant challenges since defective regions having arbitrary shapes and sizes must be identified accurately, unlike the 1D case. By identifying 3×3 squares that are defective and growing these squares over the defective region to accommodate all edge connected bit cells, the authors in [32] were able to map most of the defective cells to form a largest edge connected region as shown in Figure 34(b). Those cells that were not mapped as part of the region growing procedure were treated as random errors. The reader can now appreciate the role of modeling to provide ECC specifications for both random and burst error protection.



(a) 2D defective regions

(b) Identified rectangular bursts

Figure 34. 2D defective regions on a medium are identified within a largest edge connected region. These are later flagged as erasures and corrected using an LDPC code and a channel detector. Adapted from [32].

Figure 35 shows the schematic of a defect detector architecture. The LLR values within the detector are fixed to zero for defective cells to mark erasures. For other bits, the LLR values are populated using the extrinsic information from the decoder. Using the defect detection algorithm [32], the authors were able to correct 38×38 burst erasures, yielding more than 2 dB gain in electrical SNR. By using inter- and intra-codeword interleaving schemes, up to 76×76 burst erasures were corrected [32]. The design of good interleavers [199] is also important for enhancing the burst erasure capability of the channel.

Kurkoski et al. [200] considered the idea of fusing the partial response channel with parity check constraints to form joint factor graphs and obtained message passing decoders that showed better performance than individually optimized detectors and decoders over the perpendicular magnetic recording channel. Matcha et al. [180] developed a 2D joint detection decoding engine based on the GBP algorithm for TDMR channels. In the same spirit, one could explore joint architectures by fusing 2D LDPC decoders [183] with suitable 2D ISI detectors we discussed in this article if hardware constraints permit these architectures in practice.

VI. Perspectives and Conclusions

The push for higher areal densities and the need to stay competitive in the market has steadily advanced magnetic storage from longitudinal recording to perpendicular recording moving towards novel technologies like HAMR, MAMR, BPMR and TDMR technologies. During this journey of technology development, every aspect of the recording subsystem from heads and media to tribology and channels engineering is pushed towards achieving the objectives. With combinations from HAMR or MAMR and TDMR, one can expect combined higher areal densities. Today we already have 3 TB platters, and there is a drive to achieve significantly higher areal densities beyond $5\mathrm{Tb}/\mathrm{in}^2$ in the coming years.

The development of read channels for HDDs is a research odyssey. From analog equalizers and (d, k) constrained codes tailored for peak detection in early versions of HDDs, the development of PRML-based LDPC-coded channels made a significant milestone in the history of read channels evolution. The need for path-breaking solutions has steadily pushed advanced channels development. For example, though LDPC codes and message passing techniques were well-known by early 1990s, it took significant efforts to tailor LDPC codes with no noticeable floors at code failure rates around 10^{-10} over $4\mathrm{KB}$ sectors, develop improved decoding strategies for better SNR gains, and yet be amenable for hardware. In addition, practical constraints such as low-latencies, high throughputs, area and power-efficient designs have pushed innovations into all aspects of the read channels architecture. With 2 readers and joint equalization, SMR systems have shown $\sim 10\%$ AD gains over the PMR case. There is still more room to get additional gains by going to native 2D sector formats.

The role of information theory, signal processing algorithms and coding techniques is *central* towards building such high-performance circuits and systems in practice. As long as there is enough SNR in the magnetic channel to resolve the bits, shifting to 2D from a 1D paradigm will only bring additional gains. Tools from decades of research in the field of information theory, coding theory and signal processing have taught us many valuable techniques to solve complicated problems.

Along this thread of thought, even though TDMR poses several challenges from a technology perspective, such as the need for multiple heads/multiple readers etc., one can expect significant SNR gains by embracing native 2D coding and signal processing solutions that deal with crosstalk and noise along and across the tracks. A paradigm change in dealing with 2D coded sectors instead of 1D sectors will bring in significant changes in the design of hardware, requiring carefully engineered parallel and distributed circuits and system architectures to reduce latencies during detection and decoding. We hope that with low-power VLSI technology nodes, some of these sophisticated algorithms will make it into practice.

The future of magnetic recording rests on how recording physics, media, and channels engineering will converge towards realizing the ultimate aim of storing 1 bit/grain reliably over layered media. With such an extreme push for ADs, one has to overcome noise and crosstalk, possibly in 3D over layered media. Having an array of low-cost and efficient nano read heads to sense the signals from these grains and process them in a parallel/distributed way is the key towards getting significantly higher throughputs. All these requirements come

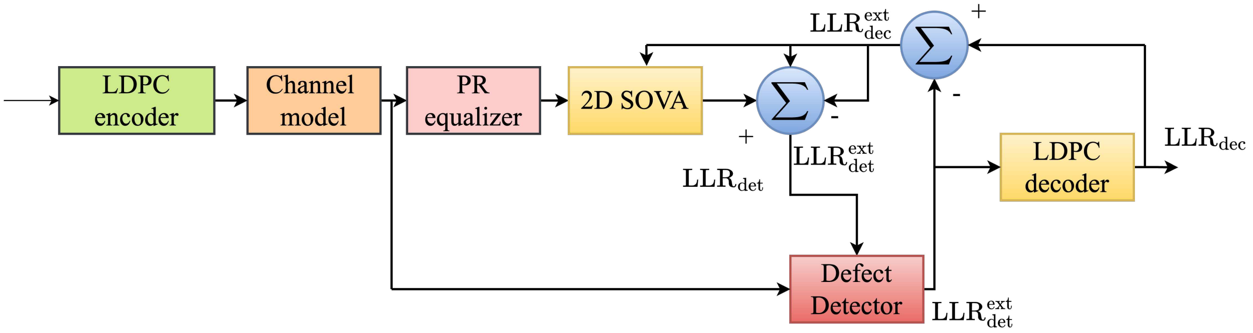

LDPC encoder
Channel model
PR equalizer
2D SOVA
$\sum$
+
-
$\mathrm{LLR}_{\mathrm{det}}$
$\mathrm{LLR}_{\mathrm{det}}^{\mathrm{ext}}$
$\mathrm{LLR}_{\mathrm{dec}}^{\mathrm{ext}}$
$\sum$
+
-
LDPC decoder
$\mathrm{LLR}_{\mathrm{dec}}$
Defect Detector
$\mathrm{LLR}_{\mathrm{det}}^{\mathrm{ext}}$

Figure 35. Schematic of a defect detector and erasure decoder architecture for TDMR. The LLR values for bits identified as defects are set to zero in the channel detector. The LDPC decoder provides extrinsic information for these erasures. The detector and the decoder are iteratively configured to resolve the burst erasures. Adapted from [32].

at the price of area and power needed for realizing practical VLSI circuits for read channels, calling for innovations in the process and device technologies to work at extreme scales and speeds and still be cost-effective.

We would like to also point out that, the solutions to 2D channels relevant to HDDs we presented in this article are applicable to physical layer wireless channels as well. The 2D coding techniques we presented are also applicable to 3D NAND flash memories and multimedia imaging systems. Last, the authors hope that, with further advancements in the physics and media of holographic recording, this archival storage technology could be resurrected, and many useful solutions developed for TDMR channels could be adapted for holographic channels.

Acknowledgment

First, the authors would like to thank the guest editors for inviting us to write this article. We also thank the editor and reviewers for many helpful comments.

S. S. Garani would like to acknowledge all his former industry and academic collaborators. He would like to acknowledge his former PhD student that significantly contributed to TDMR channel efforts, particularly, Dr. Chaitanya Kumar Matcha, Dr. Shounak Roy and Dr. Arijit Mondal. He would also like to acknowledge many of his Masters students who contributed to VLSI circuit architectures for coding and signal processing solutions, relevant to data storage channels.

S. S. Garani would like to acknowledge the funding support from the Department of Science and technology (DST), and Ministry of Electronics and Information Technology (MeitY), and the Indo-US Science and Technology Forum (IUSSTF) for supporting data storage research at PNSI laboratory. Last, S. S. Garani would like to thank his students Sudhir Kumar Sahoo and Abhi Kumar Sharma for their help in preparing some of the figures and compiling some references.

B. Vasić acknowledges the support of the NSF under grants CCF-1855879, CCF-2100013, CIF-2106189, CCSS-2027844 and CCSS-2052751. B. Vasić has disclosed an outside interest in Codelucida to the University of Arizona. Conflicts of interest resulting from this interest are being managed by The University of Arizona in accordance with its policies. The work of Xin Xiao was done while she was at the University of Arizona.

References

[1] IDC Forecasts. *Blog: https://blogs.idc.com/2021/01/06/future-of-industry-ecosystems-shared-data-and-insights/*, 2021.

[2] D. Reinsel, J. Gantz, and J. Rydning. The digitization of the world: From edge to core. *Data age 2025: https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf*, 2018.

[3] IBM Cloud Education. *Blog: https://www.ibm.com/cloud/blog/hard-disk-drive-vs-solid-state-drive*, 2022.

[4] R. F. L. Evans, R. W. Chantrell, U. Nowak, A. Lyberatos, and H.-J. Richter. Thermally induced error: Density limit for magnetic data storage. *Appl. Phys. Lett.*, 100(10):1–3, March 2012.

[5] R. Wood. Shingled magnetic recording (SMR) and two-dimensional magnetic recording (TDMR). *Journal of Magnetism and Magnetic Materials*, 561, Nov. 2022.

[6] S. S. Garani, L. Dolecek, J. R. Barry, F. Sala, and B. Vasić. Signal processing and coding techniques for 2-D magnetic recording: An overview. *Proc. of the IEEE*, 106(2):286–318, Feb. 2018.

[7] E. D. Daniel, C. D. Mee, and M. H. Clark. *Magnetic Recording: The First 100 Years*. IEEE Press, Piscataway, 1999.

[8] P. H. Siegel. Recording codes for digital magnetic storage. *IEEE. Trans. Magn.*, 21(5):1344–1349, Sep 1985.

[9] F. B. Shelledy and J. Mix. Magnetoresistive heads for magnetic tape and disk recording. *IEEE Transactions on Magnetics*, 28(5):283–288.

[10] H. N. Bertram. *The Theory of Magnetic Recording*. Cambridge Univ. Press, London, 1994.

[11] H. Kobayashi and D. T. Tang. Application of partial-response channel coding to magnetic recording systems. *IBM Jour. of Res. and Dev.*, 14(4):368–375, July 1970.

[12] APS News:. 2007 Nobel Prize honors GMR discovery. *American Physical Society*, 2007.

[13] E. Eleftheriou and W. Hirt. Noise-predictive maximum likelihood for PRML channels. In *Intl. Conference on Communciations*, Dallas, June 1996.

[14] H. J. Richter, A. Lyberatos, U. Nowak, R. F. L. Evans, and R. W. Chantrell. The thermodynamic limits of magnetic recording. *Jour. Appl. Phys.*, 111:1–9, Feb. 2012.

[15] S. N. Piramanayagam. Perpendicular recording media for hard disk drives. *Journal of Applied Physics*, 102(011301), Jul. 2007.

[16] Y. Tanaka, S. Iwasaki, and H. Muraoka. Perpendicular magnetic recording. In *IEEE History of Electrotechnology Conf.*, Kobe, Japan, Aug. 2017.

[17] R. E. Rottmayer et al. Heat-assisted magnetic recording. *IEEE. Trans. Magn.*, 42(10):2417–2421, Oct. 2006.

[18] T. Coughlin. Toshiba's 3-D magnetic recording may increase hard disk drive capacity. July 2015. [Online; posted 9-July-2015].

[19] Russak et al. Patterned media processing: A manufacturing approach. In *IDEMA Technical Conf.*, San Jose, Sep. 2010.

[20] Mark H. Kryder, Edward C. Gage, Terry W. McDaniel, William A. Challener, Robert E. Rottmayer, Ganping Ju, Yiao-Tee Hsia, and M. Fatih Erden. Heat assisted magnetic recording. *Proceedings of the IEEE*, 96(11):1810–1835, 2008.

[21] T. R. Albrecht et al. Bit-patterned magnetic recording: Theory, media fabrication, and recording performance. *IEEE Transactions on Magnetics*, 51(5):1–42, May 2015.

[22] R. Wood, M. Williams, A. Kavçić, and J. Miles. The feasibility of magnetic recording at 10 terabits per square inch on conventional media. *IEEE Trans. Magnetics*, 45(2):917–923, Feb. 2009.

[23] Seagate:. HAMR technology. *Technology Paper*, 2017.

[24] J-G Zhu, X. Zhu, and Y. Tang. Microwave assisted magnetic recording. *IEEE Trans. Magnetics*, 44(1), Jan. 2008.

[25] S. Greaves, K. S. Chan, and Y. Kanai. Optimization of dual-structure recording media for microwave-assisted magnetic recording. *IEEE Trans. Magnetics*, 55(07), Jul. 2019.

[26] Chen W. et al. Spin-torque-oscillator designs in microwave-assisted magnetic recording. In *IEEE 33rd TMRC*, pages 1–2, 2022.

[27] B. Vasić and E. Kurtas eds. *Coding and Signal Processing for Magnetic Recording Systems*. CRC Press, New York, 2005.

[28] T. Oenning and J. Moon. Modeling the Lorentzian magnetic recording channel with transition noise. *IEEE Transactions on Magnetics*, 37(1):583–591, 2001.

[29] A. R. Krishnan, R. Radhakrishnan, and B. Vasić. Read channel modeling for detection in two-dimensional magnetic recording systems. *IEEE Trans. Magnetics*, 45(10):3679–3682, Oct. 2009.

[30] R. Radhakrishnan, A. R. Krishnan, and B. Vasić. Error and erasure rates for two-dimensional magnetic recording systems. In *2009 9th International Conference on Telecommunication in Modern Satellite, Cable, and Broadcasting Services*, pages 406–409, Oct 2009.

[31] C. K. Matcha and S. S. Garani. Generalized partial response equalization and data-dependent noise predictive signal detection over media models for TDMR. *IEEE Trans. Magnetics*, 51(10):1–15, Oct. 2015.

[32] C. K. Matcha and S. S. Garani. Defect detection and burst error correction for TDMR. *IEEE Trans. Magnetics*, 52(11), Nov. 2016.

[33] J. J. Miles. Effect of grain size distribution on the performance of perpendicular recording media. *IEEE Trans. Magnetics*, 43(3):955–967, Mar. 2007.

[34] J. J. Miles, D. M. A. McKirdy, R. W. Chantrell, and R. Wood. Parametric optimization for terabit perpendicular recording. *IEEE Trans. Magnetics*, 39(4):1876–1890, July 2003.

[35] D. T. Wilton, D. M. A. McKirdy, H. A. Shute, J. J. Miles, and D. J. Mapps. Approximate three-dimensional head fields for perpendicular magnetic recording. *IEEE Trans. Magnetics*, 40(1):148–156, Jan. 2004.

[36] D. Suess, V. Tsiantos, T. Schrefl, J. Fidler, W. Scholx, H. Forster, R. Dittrich, and J. Miles. Time resolved micromagnetics using a preconditioned time integration method. *Journal of Magnetics Material*, 248:298–311, Mar. 2002.

[37] 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 Transactions on Magnetics*, 45(10):3837–3843, 2009.

[38] R. H. Victora, Jianhua Xue, and Mohammed Patwari. Areal density limits for perpendicular magnetic recording. *IEEE Trans. Magnetics*, 38(5):1886–1891, Sept. 2002.

[39] B. Vasić, M. Khatami, Y. Nakamura, Y. Okamoto, Y. Kanai, J. R. Barry, S. W. McLaughlin, and E. B. Sadeghian. A study of TDMR signal processing opportunities based on quasi-micromagnetic simulations. *IEEE Trans. Magnetics*, 51(4):1–7, April 2015.

[40] J. R. Barry, B. Vasić, M. Khatami, M. Bahrami, Y. Nakamura, Y. Okamoto, and Y. Kanai. Optimization of bit geometry and multi-reader geometry for two-dimensional magnetic recording. *IEEE Trans. Magnetics*, 52(2):1–7, Feb 2016.

[41] D. Dunbar and G. Humphreys. A spatial data structure for fast Poisson-disk sample generation. *ACM Trans. Graph.*, 25(3):503–508, July 2006.

[42] A.R. Krishnan, R. Radhakrishnan, B. Vasić, A. Kavčić, W. Ryan, and F. Erden. 2-D magnetic recording: Read channel modeling and detection. *IEEE Trans. Magnetics*, 45(10):3830 –3836, Oct. 2009.

[43] M. Khatami, M. Bahrami, and B. Vasić. Symmetric information rate estimation and bit aspect ratio optimization for TDMR using generalized belief propagation. In *Proc. IEEE Int. Symp. Information Theory (ISIT)*, pages 1620–1624, June 2015.

[44] C. K. Matcha, M. Bahrami, S. Roy, S. S. Garani, and B. Vasić. Generalized belief propagation based TDMR detector and decoder. In *Proc. IEEE Int. Symp. Information Theory (ISIT)*, pages 210–214, July 2016.

[45] M. Khatami, M. Bahrami, and B. Vasić. Information rates of constrained TDMR channels using generalized belief propagation. In *Proc. IEEE Global Telecommunications Conf. (GLOBECOM)*, pages 1–6, San Diego, CA, Dec. 2015.

[46] S. Karakulak, P. H. Siegel, J. K. Wolf, and H. N. Bertram. A new read channel model for patterned media storage. *IEEE Transactions on Magnetics*, 44(1):193–197, 2008.

[47] S. Karakulak, P. H. Siegel, and J. K. Wolf. A parametric study of inter-track interference in bit patterned media recording. *IEEE Transactions on Magnetics*, 46(3):819–824, 2010.

[48] A. R. Iyengar, P. H. Siegel, and J. K. Wolf. Data-dependent write channel model for magnetic recording. In *2010 IEEE International Symposium on Information Theory*, pages 958–962, 2010.

[49] Y. Ng, B. V. K. Vijaya Kumar, C. Kui, R. Radhakrishnan, and T. C. Chong. Bi-directional pattern-dependent noise prediction for heat-assisted magnetic recording with high jitter noise. *IEEE Transactions on Magnetics*, 48(5):1819–1825, 2012.

[50] T. Rausch, J. A. Bain, D. D. Stancil, and T. E. Schlesinger. Thermal Williams-Comstock model for predicting transition lengths in a heat-assisted magnetic recording system. *IEEE Transactions on Magnetics*, 40(1):137–147, 2004.

[51] Y. Wang and B. V. K. Vijaya Kumar. Write modeling and read signal processing for heat-assisted bit-patterned media recording. *IEEE Transactions on Magnetics*, 54(2):1–10, 2018.

[52] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.

[53] P. H. Siegel, E. Soljanin, A. J. Van Wijngaarden, and B. Vasić. *Advances in information recording*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 2008.

[54] A. Kavcić, X. Huang, B. Vasić, W. Ryan, and M. F. Erden. Channel modeling and capacity bounds for two-dimensional magnetic recording. *IEEE Trans. Magnetics*, 46(3):812–818, Mar. 2010.

[55] S. Yang, A. Kavčić, and W. Ryan. Optimizing the bit aspect ratio of a recording system using an information-theoretic criterion. In *IEEE Int. Magnetics Conf. (INTERMAG 2003)*, Mar. 2003.

[56] K. S. Chan et al. User areal density optimization for conventional and 2-d detectors/decoders. *IEEE Trans. Magn.*, Aug. 2017.

[57] S. Shamai, L. H. Ozarow, and A. D. Wyner. Information rates for a discrete-time Gaussian channel with intersymbol interference and stationary inputs. *IEEE Trans. Information Theory*, 37(6):1527–1539, Nov 1991.

[58] S. Shamai and R. Laroia. The intersymbol interference channel: lower bounds on capacity and channel precoding loss. *IEEE Trans. Information Theory*, 42(5):1388–1404, Sep. 1996.

[59] A. Kavčić. On the capacity of Markov sources over noisy channels. In *Proc. IEEE Global Telecommunications Conf. (GLOBECOM)*, 2001.

[60] S. Yang, A. Kavčić, and S. Tatikonda. Feedback capacity of finite-state machine channels. *IEEE Trans. Information Theory*, 51(3):799–810, Mar. 2005.

[61] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Information Theory*, 20(2):284–287, Mar 1974.

[62] D. M. Arnold, H. A. Loeliger, P. O. Vontobel, A. Kavčić, and W. Zeng. Simulation-based computation of information rates for channels with memory. *IEEE Transactions on Information Theory*, 52(8):3498–3508, 2006.

[63] A. Kavčić and J. M. F. Moura. The Viterbi algorithm and Markov noise memory. *IEEE Transactions on Information Theory*, 46(1):291–301, 2000.

[64] T. Cover and J. Thomas. *Elemenents of Information Theory*. 1st ed. John Wiley and Sons, 1991.

[65] J. Chen and P. H. Siegel. On the symmetric information rate of two-dimensional finite-state ISI channels. *IEEE Trans. Information Theory*, 52(1):227–236, Jan 2006.

[66] Y. Li and P. H. Siegel. On the capacity of 2-dimensional channels. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 1251–1255, 2018.

[67] M. Jordan. *Learning in Graphical Models*. MIT Press, 1999.

[68] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Information Theory*, 47(2):498 –519, Feb. 2001.

[69] R. G. Gallager. *Low Density Parity Check Codes*. PhD thesis, Cambridge, MA, 1963.

[70] J. Pearl. *Probablisitic Reasoning in Intelligent Systems*. Kaufmann, San Francisco, CA, 1988.

[71] B. J. Frey. *Graphical models for machine learning and digital communication*. MIT Press, Cambridge, MA, USA, 1998.

[72] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Trans. Inf. Theory*, 51(7):2282 – 2312, Jul. 2005.

[73] M. Mézard, G. Parisi, and M. Virasoro. *Spin Glass Theory and Beyond*. World Scientific, Singapore, 1987.

[74] H. Nishimori. *Statistical Physics of Spin Glasses and Information Processing*. Oxford Univ., Oxford, U.K., 2001.

[75] R. Kikuchi. A theory of cooperative phenomena. *Physical Review Online Archive (Prola)*, 81(6):988+, March 1951.

[76] A. Pelizzola. Cluster variation method in statistical physics and probabilistic graphical models. *Journal of Physics A: Mathematical and General*, 38(33):309–339, Aug. 2005.

[77] T. Morita. *Foundations and applications of cluster variation method and path probability method*. Publication Office, Progress of Theoretical Physics, 1994.

[78] M. Tuchler, R. Koetter, and A. C. Singer. Turbo equalization: principles and new results. *IEEE Transactions on Communications*, 50(5):754–767, 2002.

[79] R. D. Cideciyan, F. Dolivo, R. Hermann, W. Hirt, and W. Schott. A PRML system for digital magnetic recording. *IEEE Journal on Selected Areas in Communications*, 10(1):38–56, 1992.

[80] J. R. Barry, A. Kavcic, S. W. McLaughlin, A. Nayak, and W. Zeng. Iterative timing recovery. *Signal Processing Magazine, IEEE*, 21(1):89–102, Jan 2004.

[81] P. M. Aziz and S. Surendran. Symbol rate timing recovery for higher order partial response channels. *IEEE Journal on Selected Areas in Communications*, 19(4):635–648, Apr 2001.

[82] Z. Wu and J. M. Cioffi. A MMSE interpolated timing recovery scheme for the magnetic recording channel. In *Proc. IEEE International Conf. Communications (ICC)*, Montreal, Canada, Jun. 1997.

[83] K. H. Muller and M. Mülle. Timing recovery in digital synchronous data receivers. *IEEE Trans. Communications*, 24(5):516–531, May 1976.

[84] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Information Theory*, 13(2):260–269, Apr. 1967.

[85] G. D. Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[86] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: Turbo-codes. *IEEE Transactions on Communications*, 44(10):1261–1271, 1996.

[87] J. Hagenauer and P. Hoeher. A Viterbi algorithm with soft-decision outputs and its applications. In *1989 IEEE Global Telecommunications Conference and Exhibition 'Communications Technology for the 1990s and Beyond'*, pages 1680–1686 vol.3, 1989.

[88] J. Hagenauer, E. Offer, and L. Papke. Iterative decoding of binary block and convolutional codes. *IEEE Transactions on Information Theory*, 42(2):429–445, 1996.

[89] P. J. Black and T. H. Meng. A 140-Mb/s, 32-state, radix-4 Viterbi decoder. *IEEE Journal of Solid-State Circuits*, 27(12):1877–1885, 1992.

[90] A. Dey, S. Jose, K. Varghese, and S. S. Garani. A high-throughput clock-less architecture for soft-output Viterbi detection. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 779–782, 2017.

[91] M. Kaynak, S. Parthasarathy, S. Valle, and S. S. Garani. Adaptive data dependent noise prediction. In *U.S. Patent No. 8,290,102*, Oct. 2012.

[92] G. Mathew, E. Hwang, J. Park, G. Garfunkel, and D. Hu. Capacity advantage of array-reader-based magnetic recording (ARMR) for next generation hard disk drives. *IEEE Trans. Magnetics*, 50(3):155–161, Mar. 2014.

[93] R. M. Todd et al. Two-dimensional Voronoi-based model and detection for shingled magnetic recording. *IEEE Trans. Magn.*, 48(11), Nov. 2012.

[94] H. Xia, L. Lu, S. Jeong, L. Pan, and J. Xiao. Signal processing and detection for array reader magnetic recording system. *IEEE Trans. Magnetics*, 51(11):1–4, Nov 2015.

[95] L. C. Barbosa. Simultaneous detection of readback signals from interfering magnetic recording tracks using array heads. *IEEE Trans. Magnetics*, 26(5):2163–2165, Sep 1990.

[96] E. Soljanin and C. N. Georghiades. Multihead detection for multitrack recording channels. *IEEE Trans. Information Theory*, 44(7):2988–2997, Nov 1998.

[97] B. Fan, H. K. Thapar, and P. H. Siegel. Multihead multitrack detection for next generation magnetic recording, Part I: Weighted sum subtract joint detection with ITI estimation. *IEEE Trans. Comm.*, 65(04), Apr. 2017.

[98] B. Fan, H. K. Thapar, and P. H. Siegel. Multihead multitrack detection for next generation magnetic recording, Part II: Complexity reduction – algorithms and performance analysis. *IEEE Trans. Comm.*, 65(04), Apr. 2017.

[99] Elnaz Banan Sadeghian. *Synchronization and Detection for Two-Dimensional Magnetic Recording*. PhD thesis, Georgia Institute of Technology, 2016.

[100] E. B. Sadeghian and J. R. Barry. Partial-response maximum-likelihood joint detection of asynchronous tracks. In *ICC 2021 - IEEE International Conference on Communications*, pages 1–6, 2021.

[101] E. B. Sadeghian and J. R. Barry. The rotating-target algorithm for jointly detecting asynchronous tracks. *IEEE Journal on Selected Areas in Communications*, 34(9):2463–2469, Sept 2016.

[102] E. B. Sadeghian and J. R. Barry. Asynchronous multitrack detection with a generalized partial-response maximum-likelihood strategy. *IEEE Transactions on Communications*, 70(3):1595–1605, 2022.

[103] C. K. Matcha and S. S. Garani. Generalized partial response equalization and data-dependent noise predictive signal detection over media models for TDMR. *IEEE Trans. Magnetics*, 51(10):1–15, Oct 2015.

[104] B. P. Reddy, S. S. Garani, and S. Dahandeh. Timing recovery algorithms and architectures for two-dimensional magnetic recording systems. *IEEE Trans. Magnetics*, 51(4), Apr. 2015.

[105] E. Modak, B. P. Reddy, and S. S. Garani. Optimum timing interpolation algorithm for 2-D magnetic recording systems. In *National Conference on Communciations (NCC)*, Guwahati, India, Mar. 2016.

[106] Chaitanya Kumar Matcha. *Signal Processing and Coding for Two-Dimensional Magnetic Recording*. PhD thesis, Indian Institute of Science, Bengaluru, 2018.

[107] Y. Chen and S. S. Garani. Joint self-iterating equalization and detection for two-dimensional intersymbol-interference channels. *IEEE Trans. Comm.*, 61(8):3219–3230, Aug. 2013.

[108] O. Shental, N. Shental, S. Shamai, I. Kanter, A. J. Weiss, and Y. Weiss. Discrete-input two-dimensional Gaussian channels with memory: Estimation and information rates via graphical models and statistical mechanics. *IEEE Trans. Information Theory*, 54(4):1500 – 1513, Apr. 2008.

[109] A. Sayyafan, A. Aboutaleb, B. J. Belzer, K. Sivakumar, S. Greaves, and K. S. Chan. Turbo-detection for multilayer magnetic recording using deep neural network-based equalizer and media noise predictor. *IEEE Transactions on Magnetics*, 58(4):1–11, 2022.

[110] A. Sayyafan, A. Aboutaleb, B. J. Belzer, K. Sivakumar, S. Greaves, K. S. Chan, and A. James. Convolutional neural network-based media noise prediction and equalization for TDMR turbo-detection with write/read tmr. *IEEE Transactions on Magnetics*, 59(3):1–11, 2023.

[111] C. K. Matcha and S. S. Garani. Joint timing recovery and signal detection for two-dimensional magnetic recording. *IEEE Trans. Magnetics*, 53(2), Feb. 2017.

[112] S. Datta and S. S. Garani. Design architecture of a 2-D separable iterative soft-output Viterbi detector. *IEEE Transactions on Magnetics*, 52(5):1–14, 2016.

[113] Y. Wu, J. A. O'Sullivan, N. Singla, and R. S. Indeck. Iterative detection and decoding for separable two-dimensional intersymbol interference. *IEEE Transactions on Magnetics*, 39(4):2115–2120, 2003.

[114] W. G. Bliss. Circuitry for performing error correction calculations on baseband encoded data to eliminate error propagation. *IBM Tech. Disc. Bull.*, 23:4633–4634, 1981.

[115] D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1996.

[116] B. H. Marcus, R. M. Roth, and P. H. Siegel. Introduction to coding for constrained systems. *https://ronny.cswp.cs.technion.ac.il/wp-content/uploads/sites/54/2016/05/chapters1-9.pdf*, 2001.

[117] R. Adler, D. Coppersmith, and M. Hassner. Algorithms for sliding block codes - An application of symbolic dynamics to information theory. *IEEE Transactions on Information Theory*, 29(1):5–22, 1983.

[118] K. A. S. Immink, P. H. Siegel, and J. K. Wolf. Codes for digital recorders. *IEEE Transactions on Information Theory*, 44(6):2260–2299, 1998.

[119] P. E. Bender and J. K. Wolf. A universal algorithm for generating optimal and nearly optimal run-length-limited, charge-constrained binary sequences. In *Proceedings. IEEE International Symposium on Information Theory*, pages 6–6, 1993.

[120] E. Zehavi and J. K. Wolf. On runlength codes. *IEEE Trans. Information Theory*, 34(1):45–54, Jan. 1988.

[121] B. H. Marcus, P. H. Siegel, and J. K. Wolf. Finite state modulation codes for data storage. *IEEE Journal on Selected Areas in Communications*, 10(1):5–37, Jan. 1992.

[122] S. A. Altekar, M. Berggren, B. E. Moision, P. H. Siegel, and J. K. Wolf. Error-event characterization on partial-response channels. *IEEE Transactions on Information Theory*, 45(1):241–247, 1999.

[123] J. Moon and B. Brickner. Maximum transition run codes for data storage systems. *IEEE Transactions on Magnetics*, 32(5):3992–3994, 1996.

[124] A. Gallopoulos, C. Heegard, and P. Siegel. The power spectrum of run-length-limited codes. *IEEE Transactions on Communications*, 37(9):906–917, 1989.

[125] R. Karabed and P. H. Siegel. Matched spectral-null codes for partial-response channels. *IEEE Transactions on Information Theory*, 37(3):818–855, 1991.

[126] R. M. Roth, P. H. Siegel, and A. Vardy. High-order spectral-null codes-constructions and bounds. *IEEE Transactions on Information Theory*, 40(6):1826–1840, 1994.

[127] Y. Sankarasubramaniam and S. W. McLaughlin. Fixed-rate maximum-runlength-limited codes from variable-rate bit stuffing. *IEEE Trans. Information Theory*, 53(8):2769–2790, Aug. 2007.

[128] N. J. Calkin and H. S. Wilf. The number of independent sets in a grid graph. *SIAM Journal on Discrete Math*, 11(1):54–60, Jan. 1998.

[129] S. Halevy et al. Improved bit-stuffing bounds on two-dimensional constraints. *IEEE Trans. Information Theory*, 50(5):824–838, May 2004.

[130] W. Weeks IV and R. E. Blahut. The capacity and coding gain of certain checkerboard codes. *IEEE Trans. Information Theory*, 44(5):1193–1203, May 1998.

[131] S. S. Garani and S. W. McLaughlin. Capacity bounds for two-dimensional asymmetric M-ary $(0, k)$ and (d, ∞) runlength-limited channels. *IEEE Trans. Comm.*, 57(6):1584–1587, June 2009.

[132] I. Demirkan and J. K. Wolf. Block codes for the hard square model. *IEEE Trans. Information Theory*, 51(8):2836–2848, Aug. 2005.

[133] Ido Tal, Tuvi Etzion, and Ron M. Roth. On row-by-row coding for 2-d constraints. *IEEE Transactions on Information Theory*, 55(8):3565–3576, 2009.

[134] M. Bahrami, C. K. Matcha, S. Roy, S. S. Garani, and B. Vasić. Investigation into harmful patterns over multi-track shingled magnetic detection using the Voronoi model. *IEEE Trans. Magnetics*, 51(12), Dec. 2015.

[135] B. Dabak, A. Hareedy, and R. Calderbank. Non-binary constrained codes for two-dimensional magnetic recording. *IEEE Transactions on Magnetics*, 56(11):1–10, 2020.

[136] V. Dorfman and J. K. Wolf. A method for reducing the effects of thermal asperities. *IEEE Journal on Selected Areas in Communications*, 19(4):662–667, Apr. 2001.

[137] W. Tan and J. R. Cruz. Detection of media defects in perpendicular magnetic recording channels. *IEEE Trans. Magnetics*, 41(10):2956–2958, Oct. 2005.

[138] T. Richardson. Error floors of LDPC codes. In *Allerton*, Monticello, IL, October 2003.

[139] S. Lin and D. J Costello. *Error control coding*, volume 2. Prentice Hall, IN, 2004.

[140] A. Mondal, S. Thatimattala, V. K. Yalamaddi, and S. S. Garani. Efficient coding architectures for Reed-Solomon and low-density parity-check decoders for magnetic and other data storage systems. *IEEE Trans. Magn.*, 54(2):1–15, Jan. 2018.

[141] W. Peterson and E. J Weldon. *Error-correcting codes*. MIT press, 1972.

[142] X. Zhang. *VLSI architectures for modern error-correcting codes*. Crc Press, 2015.

[143] Y. Chen and K. K. Parhi. Small area parallel Chien search architectures for long BCH codes. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(5):545–549, 2004.

[144] D. V. Sarwate and N. R. Shanbhag. High-speed architectures for Reed-Solomon decoders. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 9(5):641–655, 2001.

[145] R. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5):533–547, 1981.

[146] D. J. C. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(2):399–431, 1999.

[147] T. Richardson and R. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. Information Theory*, 47(2):599–618, Feb. 2001.

[148] T. Richardson and R. Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008.

[149] R. G. Gallager. *Low density parity check codes*. PhD thesis, Massachusetts Institute of Technology, 1966.

[150] J. L. Fan. *Array Codes as LDPC Codes*, pages 195–203. Springer US, Boston, MA, 2001.

[151] M. P. C. Fossorier. Quasicyclic low-density parity-check codes from circulant permutation matrices. *IEEE Transactions on Information Theory*, 50(8):1788–1793, 2004.

[152] R. Smarandache and P. O. Vontobel. Quasi-cyclic LDPC codes: Influence of proto- and Tanner-graph structure on minimum Hamming distance upper bounds. *IEEE Transactions on Information Theory*, 58(2):585–607, 2012.

[153] D. Declercq and M. Fossorier. Improved impulse method to evaluate the low weight profile of sparse binary linear codes. In *2008 IEEE International Symposium on Information Theory*, pages 1963–1967, 2008.

[154] S. Jeon and B. V. K. V. Kumar. Performance and complexity of 32 k-bit binary LDPC codes for magnetic recording channels. *IEEE Trans. Magnetics*, 46(6):2244–2247, Jun. 2010.

[155] H. Zhong, T. Zhang, and E. F. Haratsch. Quasi-cyclic LDPC codes for the magnetic recording channel: Code design and VLSI implementation. *IEEE Trans. Magnetics*, 43(3):1118–1121, Mar. 2007.

[156] Z. Li, L. Chen, L. Zeng, S. Lin, and W. H. Fong. Efficient encoding of quasi-cyclic low-density parity-check codes. *IEEE Transactions on Communications*, 54(1):71–81, 2006.

[157] T. Richardson. Error floors of LDPC codes. In *Allerton*, Monticello, IL, October 2003.

[158] S. Landner and O. Milenkovic. Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes. In *Proc. IEEE Int. Conf. Wireless Networks, Communications and Mobile Computing (WIRELESSCOM)*, Honolulu, HI, June 2005.

[159] B. Vasić, S. K. Chilappagari, D. V. Nguyen, and D. Declercq. Trapping set ontology. In *Forty-Seventh Annual Allerton Conference*, Monticello, IL, Oct. 2009.

[160] X. Hu, Z. Li, B. V. K. Vijaya Kumar, and R. Barndt. Error floor estimation of long LDPC codes on magnetic recording channels. *IEEE Transactions on Magnetics*, 46(6):1836–1839, 2010.

[161] M. P. C. Fossorier, M. Mihaljevic, and H. Imai. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation. *IEEE Transactions on Communications*, 47(5):673–680, 1999.

[162] D. E. Hocevar. A reduced complexity decoder architecture via layered decoding of LDPC codes. In *IEEE Workshop on Signal Processing Systems*, Austin, U.S.A, Oct. 2004.

[163] K. K. Gunnam and G. S. Choi. Low density parity check decoder for regular LDPC codes. In *U.S. Patent No. 8,359,522*, Jan. 2013.

[164] H. Song and J. R. Cruz. Reduced-complexity decoding of Q-ary LDPC codes for magnetic recording. *IEEE Transactions on Magnetics*, 39(2):1081–1087, 2003.

[165] D. Declercq and M. Fossorier. Decoding algorithms for nonbinary LDPC codes over GF(q). *IEEE Transactions on Communications*, 55(4):633–643, 2007.

[166] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard. Low-complexity decoding for non-binary LDPC codes in high order fields. *IEEE Transactions on Communications*, 58(5):1365–1375, 2010.

[167] A. Hareedy, B. Amiri, R. Galbraith, and L. Dolecek. Non-binary LDPC codes for magnetic recording channels: Error floor analysis and optimized code design. *IEEE Trans. Communications*, pp(99):1–1, to appear 2016.

[168] A. Hareedy, C. Lanka, N. Guo, and L. Dolecek. A combinatorial methodology for optimizing non-binary graph-based codes: Theoretical analysis and applications in data storage. *IEEE Transactions on Information Theory*, 65(4):2128–2154, 2019.

[169] F. García-Herrero, E. Li, D. Declercq, and J. Valls. Multiple-vote symbol-flipping decoder for nonbinary LDPC codes. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(11):2256–2267, 2014.

[170] E. Li, D. Declercq, and K. Gunnam. Trellis-based extended min-sum algorithm for non-binary LDPC codes and its hardware structure. *IEEE Transactions on Communications*, 61(7):2600–2611, 2013.

[171] J. O. Lacruz, F. García-Herrero, D. Declercq, and J. Valls. Simplified trellis min–max decoder architecture for nonbinary low-density parity-check codes. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(9):1783–1792, 2015.

[172] A. Risso. Layered LDPC decoding over GF(q) for magnetic recording channel. *IEEE Trans. Magnetics*, 45(10):3683–3686, Oct. 2009.

[173] A. J. Felstrom and K. S. Zigangirov. Time-varying periodic convolutional codes with low-density parity-check matrix. *IEEE Transactions on Inform. Theory*, 45(9):2181–2191, 1999.

[174] A. R. Iyengar, P. H. Siegel, R. L. Urbanke, and J. K. Wolf. Windowed decoding of spatially coupled codes. *IEEE Transactions on Inform. Theory*, 59(4):2277–2292, 2013.

[175] H. Esfahanizadeh, L. Tauz, and L. Dolecek. Multi-dimensional spatially-coupled code design: Enhancing the cycle properties. *IEEE Trans. Communications*, 68(5):2653–2666, May 2020.

[176] A. Hareedy, R. Kuditipudi, and R. Calderbank. Minimizing the number of detrimental objects in multi-dimensional graph-based codes. *IEEE Transactions on Communications*, 68(9):5299–5312, 2020.

[177] S. Yang, A. Hareedy, R. Calderbank, and L. Dolecek. Breaking the computational bottleneck: Probabilistic optimization of high-memory spatially-coupled codes. *IEEE Transactions on Information Theory*, 69(2):886–909, 2023.

[178] H. Esfahanizadeh, A. Hareedy, R. Wu, R. Galbraith, and L. Dolecek. Spatially-coupled codes for channels with SNR variation. *IEEE Transactions on Magnetics*, 54(11):1–5, 2018.

[179] S. S. Garani, N. Richardson, and X. Hu. Channel constrained code aware interleavers. In *U.S. Patent No. 8,055,973*, Nov. 2011.

[180] C. K. Matcha, S. Roy, M. Bahrami, B. Vasić, and S. S. Garani. 2D LDPC codes and joint detection and coding for two-dimensional magnetic recording. *IEEE Trans. Magnetics*, (8), Aug. 2017.

[181] Y. Cassuto and A. Shokrollahi. LDPC codes for 2D arrays. *IEEE Transactions on Information Theory*, 60(6):3279–3291, 2014.

[182] H. Kamabe and S. Lu. Burst erasure correction by 2D LDPC codes. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7, 2019.

[183] A. Mondal and S. S. Garani. Efficient parallel decoding architecture for cluster erasure correcting 2-D LDPC codes for 2-D data storage. *IEEE Transactions on Magnetics*, 57(12):1–16, 2021.

[184] Arijit Mondal. *Efficient Hardware Architectures for Error Correcting Codes Applicable to Data Storage*. PhD thesis, Indian Institute of Science, Bengaluru, 2022.

[185] S. W. Yoon and J. Moon. Two-dimensional error-pattern-correcting codes. *IEEE Transactions on Communications*, 63(8):2725–2740, 2015.

[186] S. Roy and S. Garani Srinivasa. Two dimensional error-correcting codes using finite field Fourier transform. In *2015 IEEE Information Theory Workshop - Fall (ITW)*, pages 119–123, 2015.

[187] H. S. Madhusudhana and M. U. Siddiqi. On Blahut's decoding algorithms for two-dimensional BCH codes. *IEEE Transactions on Information Theory*, 44(1):358–367, 1998.

[188] R. E. Blahut. Transform techniques for error control codes. *IBM Journal of Research and Development*, 23(3):299–315, 1979.

[189] A. Mondal and S. S. Garani. Efficient hardware architectures for 2-D BCH codes in the frequency domain for two-Dimensional data storage applications. *IEEE Transactions on Magnetics*, 57(5):1–14, 2021.

[190] R. E. Blahut. *Algebraic Codes on Lines, Planes, and Curves: An Engineering Approach*. Cambridge University Press, 2008.

[191] Z. A. Keirn, V. Y. Krachkovsky, E. F. Haratsch, and H. Burger. Use of redundant bits for magnetic recording: Single-parity codes and Reed-Solomon error-correcting code. *IEEE Trans. Magnetics*, 40(1):225–230, Jan. 2004.

[192] J. K. Wolf. ECC performance of interleaved RS codes with burst errors. *IEEE Trans. Magnetics*, 34(1):75–79, Jan. 1998.

[193] S. S. Garani, P. Lee, and S. W. McLaughlin. Post-error correcting code modeling of burst channels using hidden Markov models with applications to magnetic recording channels. *IEEE Trans. Magnetics*, 43(2):572–579, Feb. 2007.

[194] M. Fossorier. Universal burst error correction. In *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Seattle, USA, July 2006.

[195] E. Paolini and M. Chiani. Construction of near-optimum burst erasure correcting low-density parity-check codes. *IEEE Trans. Communications*, 57(5):1320–1328, May 2009.

[196] K. Li, A. Kavčić, and F. Erden. Construction of burst-erasure efficient LDPC codes for use with belief propagation decoding. In *Proc. IEEE International Conf. Communications (ICC)*, Cape Town, South Africa, May 2010.

[197] W. Tan et al. Erasure detection algorithms for magnetic recording channels. In *Proc. IEEE Global Telecommunications Conf. (GLOBECOM)*, New Orleans, U.S.A, Nov. 2008.

[198] S. S. Garani and S. Parthasarathy. Identifying a defect in a data-storage medium. In *U.S. Patent No. 9,324,370*, Apr. 2016.

[199] M. Blaum, J. Bruck, and A. Vardy. Interleaving schemes for multidimensional cluster errors. *IEEE Trans. Information Theory*, 44(02):730–743, Mar. 1998.

[200] B. M. Kurkoski, P. H. Siegel, and J. K. Wolf. Joint message-passing decoding of LDPC codes and partial-response channels. *IEEE Transactions on Information Theory*, 48(6):1410–1422, 2002.

Dr. Shayan Srinivasa Garani is an Associate Professor within the Division of EECS at the Indian Institute of Science, where the directs the Physical Nano-memories Signal and Information Processing Laboratory (PNSIL). Prior to joining IISc, Prof. Garani was leading various research activities, managing and directing research and external university research programs within Western Digital. He was the Chairman for signal processing for the IDEMA-ASTC and a co-chair for the overall technological committee. He was the past Chair for IEEE Data Storage Technical Committee (2018-2020).

He was a lead architect for read channels architectures and pioneered advanced coding and signal processing solutions for magnetic recording and flash memories that were put to practice in HDDs and SSDs. He holds 14+ U.S patents in the area of data storage. He has been the primary advisor to 12 PhD students and 21 Masters' students since 2012. He was awarded Prof. Satish Dhawan young engineers state award by the Govt. of Karnataka, India for outstanding contributions in the field of engineering sciences for the year 2018.

Dr. Bane Vasić is a Professor of Electrical and Computer Engineering and Mathematics at the University of Arizona and a Director of the Error Correction Laboratory.

Dr. Vasić is an inventor of the soft error-event decoding algorithm, and the key architect of a detector/decoder for Bell Labs magnetic recording read channel chips which were regarded as the best in industry. Different variants of this algorithm were implemented in virtually all magnetic hard drives.

His pioneering work on structured low-density parity check (LDPC) error correcting codes and invention of codes has enabled low-complexity iterative decoder implementations. Structured LDPC codes are today adopted in a number of communications standards and storage systems. He was a Chair of IEEE Data Storage Technical Committee.

Dr. Vasić's theoretical work on graphs which has led to characterization of the hard decision iterative decoders of LDPC codes, and design of codes and decoders for binary symmetric channel with best error-floor performance known today.

He is a co-founder of Codelucida, a startup company developing advanced error correction solutions for communications and data storage. He is an IEEE Fellow and da Vinci Fellow.