ISI/ITI Turbo Equalizer for TDMR using Trained Local Area Influence Probabilistic Model

Xueliang Sun[1], Jinlu Shen[2], Benjamin J. Belzer[2], *Member, IEEE*, Krishnamoorthy Sivakumar[2], *Senior Member, IEEE*, Ashish James[3], Kheong Sann Chan[3,4*], and Roger Wood[2], *Fellow, IEEE*

[1]Quantamental Technologies LLC, White Plains, NY 10606 USA
[2]School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164-2752 USA
[3]Institute for Infocomm Research (I2R), A*STAR, Singapore
[4]Nanjing Institute of Technology, Nanjing, China
[*]The work was done while the author was with A*STAR

In this paper, a local area influence probabilistic (LAIP) detector for estimating magnetic grain interactions with coded data bits in two dimensional magnetic recording (TDMR) is combined with a two dimensional Bahl-Cocke-Jelinek-Raviv (BCJR) based detector for joint removal of intertrack interference (ITI) and intersymbol interference (ISI). The LAIP detector sends log-likelihood-ratio (LLR) estimates of coded bits and an estimate of the local ISI/ITI convolution mask to a BCJR-based ISI/ITI detector followed by an irregular-repeat-accumulate (IRA) decoder. Simulation results on a random Voronoi grain media model with ISI and ITI show that the concatenated LAIP/BCJR system, which detects three tracks simultaneously, achieves user information bit areal densities competitive or higher than results reported in a previous paper that employed the LAIP detector alone on a Voronoi grain channel without ISI/ITI. Simulation results on a grain-flipping-probability (GFP) media model based on micromagnetic simulations show that the proposed detector achieves an 11.3% bit error rate (BER) reduction compared to a recently proposed system with a two-dimensional linear equalizer followed by a two-track BCJR detector with two-dimensional pattern dependent noise prediction (2D-PDNP).

***Index Terms*—Two-dimensional magnetic recording, iterative detection and decoding, local area influence probability, Voronoi grain model, grain-flipping-probability model**

I. Introduction

THIS paper considers detection strategies for an emerging technology called two dimensional magnetic recording (TDMR) [1]. In TDMR, channel coded bits are read and written in two dimensions on conventional magnetic hard disks, which have magnetic grains of different sizes packed randomly onto the disk surface. In future high-density versions of TDMR (e.g., between 1 and 4 magnetic grains per coded bit), the number of channel bits may be larger than the number of media grains to support them in a given local area. Thus, occasionally a bit will not be written on any grain, and hence will effectively be "overwritten" by bits on the surrounding grains. The correct detection of these overwritten bits is an important goal in TDMR detector design at high densities. A comprehensive overview of state-of-the-art signal processing and coding techniques for TDMR is provided in [2]. In the following paragraphs, we summarize a number of techniques described in [2] and elsewhere that are most closely related to the present paper.

In a previous paper [3], we design a local area influence probabilistic (LAIP) detector for TDMR with a Voronoi magnetic grain model. In an initial offline training step, the LAIP detector collects grain influences and discretizes their frequencies under different parameters into a multi-dimensional probability mass function (PMF), organized as a lookup table (LUT). In detection mode, the LAIP detector searches the LUT for estimated grain influences, and compares the readback signal with estimated overall grain influences to obtain log-likelihood ratios (LLRs). Then the LAIP detector exchanges LLRs with an irregular repeat-accumulate (IRA) decoder until LLR convergence; an IRA code is a type of low-density parity check (LDPC) code with a simple linear-time encoding algorithm [4]. The simulations in [3], which were run with length 32 kbit IRA codewords, achieve 0.5584 user bits per grain (U/G) on a Voronoi grain model with known boundary conditions, and 0.4909 U/G under more realistic random boundary conditions. In a real magnetic read/write process, the read back bits are subject to ISI from bits on the same track and inter-track interference (ITI) from bits on adjacent tracks; we use the term *two dimensional intersymbol interference* (2D-ISI) to refer to this combination of ISI/ITI. The grain model in [3] does not include 2D-ISI, and hence is somewhat oversimplified compared to a real magnetic read/write process. In the present paper, we employ a Voronoi grain model with a more realistic read model that includes 2D-ISI. As before, the LAIP detector goes through an initial off-line computation of the grain influence multi-dimensional joint probability mass functions (PMFs). As the LAIP detector only considers a local area (typically 3×3 to 3×5 bits), it cannot fully exploit larger-area 2D correlation induced by the 2D-ISI. Hence, we combine the LAIP detector and the row-wise 2D-BCJR detector of [5] to work together with the IRA decoder to estimate bit polarities.

Manuscript received January 15, 2019. Corresponding author: B. Belzer (email: belzer@eecs.wsu.edu)

The magnetic recording assumptions employed in our Voronoi grain model simulations are summarized as follows, and are described in more detail in section II: 1) media grains are modeled using a random Voronoi model; 2) the write head can magnetize at the grain level, and uses the centroid write model of [1]; 3) at the center of each square bit cell, the read head records the area integral (over a 3×3 bit area) of the grain magnetizations (which take values ± 1) weighted by a Gaussian impulse response function. We note that these assumptions are commonly used in other papers employing the Voronoi model, and are similar to the Voronoi modeling assumptions described in [2].

This paper also considers design of a LAIP detector to handle read-head data generated by a much more realistic grain flipping probability (GFP) model. The GFP model provides fast and accurate 2D readback waveforms that include effects captured from micromagnetic simulations and the statistical effects derived from the granularity of the recording medium [6]. The GFP model has been validated in previous studies against both spin-stand [7], [8] and HDD [9] signals, and an areal density estimate of HDD of that time-period has been made in [10]. In this paper, we show how the LAIP detector's PMF tables can be trained from a special GFP model training data set, such that good U/G performance is obtained on a GFP generated test data set.

A. Literature survey: addressing grain-bit interactions

The LAIP detector considered in this paper is a form of dynamic grain state estimation (DGSE), wherein estimates (direct or indirect) of the underlying grain pattern are used to improve estimates of the recorded coded bits. Previous DGSE papers [11]–[13] employed BCJR-based [14] detectors for the four-rectangular grain model (FRGM) of [15]; [13] showed that FRGM information densities greater than 0.5 bits/grain could be achieved, and that, surprisingly, the FRGM-based detector could achieve densities of 0.25 bits/grain even on data generated by the random Voronoi grain model of [16].

There have also been DGSE papers based on generalized belief propagation (GBP) [17]. GBP passes probability messages on region graphs; the regions are collections of variable nodes of interest (e.g., TDMR bit-cells). Starting from a set of smaller local regions which cover a larger region, GBP can derive an estimate of the joint PDF of the variables over the larger region. GBP-based TDMR detection was previously considered in [18], which employed a microcell model to account for grains overlapping bit-cell boundaries, under typical densities of about 9 grains per bit. In [19], authors of the present paper present a GBP-based TDMR turbo-detector that employs a random Voronoi model at 1.1 grains per coded bit (GPB) with known boundary conditions; this detector-decoder achieves densities of 0.4515 user bits/grain (U/G), but has high computational complexity compared to the LAIP detector proposed in [3] or to the combined LAIP/BCJR detector considered in this paper. GBP has also been applied in constrained modulation coding for TDMR in [20].

Another important competing approach to the proposed LAIP detector is pattern dependent noise prediction (PDNP). In one-dimensional PDNP (1D-PDNP) [21], [22], [23, chapt. 33], used on conventional HDDs with recording on 1D tracks, an MMSE filter equalizes the read head data to a partial response (PR) target of finite length I. The MMSE filter output flows into a trellis-based (Viterbi [24] or BCJR [25]) detector that employs a super-trellis based on the PR target and the 1D PDNP noise model. The 1D PDNP employs an Lth order autoregressive (AR) model of the media noise $\mathbf{n}$ [23, chapt. 33]: $\tilde{n}_k(\mathbf{u}_k) = \sum_{i=1}^{L} a_i(\mathbf{u}_k) n_{k-i}(\mathbf{u}_k) + e_k(\mathbf{u}_k)$, where the a_i, $i = 1, \ldots, L$ are the AR coefficients, and the model error e_k is assumed to be time-uncorrelated Gaussian noise with variance that depends on the coded bit pattern vector $\mathbf{u}_k = [u_{k+\Delta}, \ldots, u_k, \ldots, u_{k-(I+L)}]$, where Δ is the predictor's look-ahead length. The model coefficients $a_i(\mathbf{u}_k)$ are trained by collecting many instances of the media noise $n_k(\mathbf{u}_k)$ corresponding to a particular bit pattern $\mathbf{u}_k$, and then using the normal equation solution for the model coefficients that minimizes the mean squared error $E[|e_k(\mathbf{u}_k)|^2]$.

Several papers have proposed generalizations of 1D-PDNP to the 2D case. Papers [26]–[28] explore variations of a two-track hybrid 1D/2D PDNP scheme with 2D-BCJR, in which 1D-PDNP coefficients on both tracks depend on the 2D patterns defined by the state and input variables on both tracks. These papers demonstrate that significant SNR gains (up to 4 dB in [26] and 2.9 dB in [27]) and density gains (up to 22% in [28], which also employs a novel 2D write-precompensation scheme) result from the proposed 1D/2D PDNP scheme. More recently, a Viterbi-algorithm (VA) based two-track 2D-PDNP scheme is proposed in [29], wherein the 2D-pattern dependent prediction coefficients are 2×2 matrices, and noise prediction and bit estimation are done simultaneously for both tracks; the proposed scheme gives 4% density gains compared to a 2D-VA without PDNP. The number of trellis states in the 2D-BCJR or 2D-VA employed in [26]–[29] grows as $4^{(\Delta+I+L)}$, which can become large with increasing PR target length I and AR model order L. By contrast, the 2D data dependent noise predictive (DDNP) detection scheme in [30] limits computational complexity by providing 2D noise-prediction filters only for a finite set of problematic 2D data patterns; this scheme gives up to 10% density gains on a Voronoi grain media model near 5 GPB.

B. Literature survey: 2D-ISI detection/equalization

The 2D-BCJR 2D-ISI detector in the proposed LAIP/BCJR scheme in this paper plays an important role, especially for the considered Voronoi grain channels. As 2D-ISI detection is a key part of TDMR detection, we now present a brief time-line of 2D-ISI and related detection algorithms.

In [31], a separable 2D-ISI channel is assumed, and separability is exploited to construct an iterative row-column algorithm in which a non-binary column BCJR detector is followed by a binary row BCJR detector, followed by an iteration of an LDPC decoder, etc. In [32], soft information is exchanged between BCJR row and column detectors; this scheme makes decisions on multiple rows/columns, and does not assume channel separability. An iterative row-column soft decision feedback (SDF) algorithm (IRCSDFA) similar to that of [32] is reported in [33], in which statistical independence of the input and feedback pixels is assumed; performance gains over

[32] are achieved by use of LLR weight schedules during row-column iterations. A generalized belief propagation (GBP) [17] based 2D-ISI equalizer is developed in [34]. The GBP-based equalizer uses exact inference over the sub-region of the image covered by the ISI mask, and passes messages between adjacent overlapping sub-regions. The GBP equalizer achieves maximum-likelihood (ML) performance for the cases tested in [34], but is demonstrated only on small (20×20 or smaller) images with known boundaries that comprise a significant percentage of the total image pixels. In general, GBP-based methods suffer from high computational complexity compared to row-column VA or BCJR-based detection algorithms.

In [5], the row-column algorithm in [33] is generalized by computing joint non-binary LLRs over the input and feedback pixels (i.e., independence of those pixels is not assumed); this approach, while significantly more computationally complex due to joint LLR BCJR updates, achieves SNR gains of up to 1 dB over [33]. In the present paper, we use the rows-only BCJR from [5]; because the row trellis spans all three input tracks, there is no need for column iterations.

In [35], a row-column BCJR detector similar to [5], but with novel feedback pixel configurations and with independence of input and feedback pixels assumed, is combined with a soft input self-iterating equalizer (SISE). The SISE is a 2D MMSE 2D-ISI equalizer that employs self-iterations using turbo-equalization techniques from [36]. This two-dimensional equalizer and detector (JTED) exchanges LLRs between BCJR and SISE, and thereby achieves significant performance gains over [32] with little additional complexity cost, due to the SISE's relatively low complexity. When JTED is combined with an LDPC code in a turbo equalization scheme, SNR gains of 8 dB (over uncoded JTED) are achieved.

Finally we point out that timing recovery, while not considered here, is an important practical issue in TDMR. In [37], a joint 2D timing recovery and signal detection scheme is realized by running a 2D soft-output Viterbi algorithm (2D-SOVA) over a trellis that spans the joint state space of timing errors and 2D-ISI; to control the 2D-ISI, 2D partial response equalization (using the scheme in [30]) is done.

C. *Novel contributions*

The novel contributions of the present paper are as follows: 1) a combined LAIP/BCJR turbo detection architecture that includes LAIP detector self-loops, LAIP detector loops with the IRA decoder, and LAIP detector passing of coded bit LLRs to the 2D-BCJR detector; 2) an improved method for computing the LAIP conditional PMF tables that accounts for the influence of diagonally adjacent bits on the target bit; 3) a scheme for training the LAIP detector to predict the spatially varying 2D inner products used as branch labels in the BCJR detector; 4) a scheme for training the LAIP detector with more realistic GFP media model waveforms; 5) simulation results on a random Voronoi grain media model with ISI/ITI showing density gains compared to a previous LAIP detector paper that employed a very similar media model without ISI/ITI; 6) simulation results with GFP-modeled waveforms at 3.491 GPB that show a 0.11% U/G gain and 3× throughput gain (assuming three parallel read heads) compared to the more conventional system in [38] that employs a 3-input/1-output 2D MMSE PR equalizer with a 1D BCJR and IRA decoder, and that show a detector BER reduction of 11.3% and a 1.5× throughput gain compared to the 2D-PDNP proposed in [29]. Regarding contribution 6, we note that the present paper does not employ 2D PR equalization; further performance gains should be achievable by adding a 3-input/3-output 2D MMSE PR equalizer as a pre-processing step before the proposed LAIP detector. Unlike PDNP, which must be used as part of a trellis based detector, the proposed LAIP detector is a "stand-alone" soft-input/soft-output detector that outputs LLR estimates of the coded bits and can accept *a priori* LLR estimates of the bits; thus, the LAIP detector can exploit turbo-iterations with a channel decoder or with another detector to lower the overall system BER.

This paper is organized as follows. Section II describes the random Voronoi grain model and the assumed write and read models. Section III describes the combined LAIP and 2D-BCJR TDMR turbo-detector, for both the random Voronoi grain model and the GFP media model. Section IV gives simulation results, and section V concludes the paper.

II. Random Voronoi Grain Read-Write Model

In order to approximate real magnetic grains, we adopt the Voronoi grain model from [16] to generate nuclei locations and grain shapes, such as shown in Fig. 1. In our model, the coded bits are written row-by-row into a block of bit cells with 3 rows and N columns, in order to simulate a three track scenario with N bits per sector. The block width of three is chosen to simulate the proposed three reader configuration for TDMR, with one reader for each of the three tracks.

We define the boundary bits of the $3 \times N$ block as an extra one bit wide boundary on all four sides of the $3 \times N$ block. Thus, the entire block including boundary is of size $5 \times (N + 2)$ bits. All the bits within this $5 \times (N + 2)$ block assume that the Voronoi grain nuclei are randomly placed and uncorrelated with the bit cell center locations. When generating grain nuclei, the largest $1/7$th of the Voronoi grains are randomly split into 2 grains and the smallest $1/7$th are removed, in order to keep the ratio σ_g/μ_g of the grain area standard deviation σ_g to mean μ_g at about 25%, close to that for HDD media grains. This boundary condition is similar to the random Voronoi grain model (RVGM) in paper [3]. The

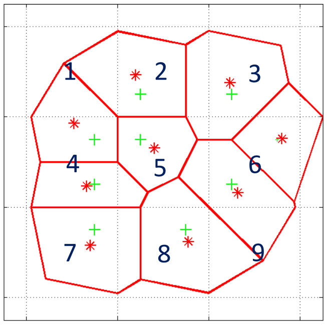

Fig. 1. Generated Voronoi grains (outlined in solid red) for 3×3 channel coded bits (indicated by black dotted lines), with approximately one coded bit per grain. The '+' and '*' indicate the grain nuclei and centroids, respectively.

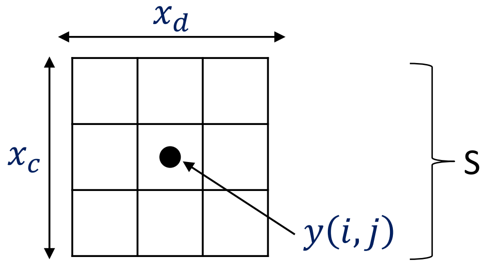

Fig. 2. 2D-ISI integration read model: the read back value $y(i,j)$ for the bit cell at (i,j) is the integral of grain magnetizations weighted by the 2D read-head impulse response within a block S of 3×3 square bit cells; the read position is assumed to be at the center of bit cell (i,j).

only difference is that although the polarities of boundary bits are randomly assigned to ±1, we assume that these polarities are known at the reader. In an actual HDD the boundary bits can be cached so that they are known.

We assume a centroid write model [16], where a given Voronoi grain is magnetized (to a value of ±1) by the channel bit containing the centroid of that grain. Under this model, overwritten bits (with the possibility of detector error) occur in bit cells that contain no grain centroids, as the polarities of the grains within such cells will be determined by the surrounding bits. For example, bits 1 and 9 in Fig. 1 will be overwritten.

Our previous LAIP detector paper [3] assumes a soft-bit read model that computes the value $y(i,j)$ read at the center of the (i,j)th bit cell as the area integral of magnetizations of all grains contained within the bit cell. In a physical system, the read head will not be able to isolate individual bits, but will suffer 2D-ISI from adjacent bits on the same track (ISI) and on adjacent tracks (ITI). Therefore, this paper takes 2D-ISI into consideration by employing a 2D-ISI integration read model, where the value $y(i,j)$ is the integral of the magnetizations of all grains contained within the 3×3 bit cell block with (i,j) as the center bit multiplied by the 2D read head impulse response centered at the (i,j)th bit cell center:

$$y(i,j) = \iint_S h(i-x_c, j-x_d) M(x_c,x_d) dx_c dx_d. \quad (1)$$

Here S is the 3×3 square bit integration area with reader position at (i,j), $M(x_c,x_d)$ (which takes values ±1) is the grain magnetization as a function of cross- and down-track positions x_c and x_d, and $h(x_c,x_d)$ is a 2D circularly symmetric Gaussian read-head impulse response function, given in [39] as

$$h(x_c,x_d) = \exp\left(\frac{-1.34898^2(x_c^2+x_d^2)}{2T_{50}^2}\right), \quad (2)$$

where T_{50} is the rise-time from 25% to 75% in the step response, and is set to the square root of the mean grain area μ_g. Fig. 2 illustrates the integration method, and Fig. 3 gives an example describing how input binary bits are written and read back to noisy analog values.

III. LOCAL AREA INFLUENCE ESTIMATION BASED TDMR TURBO-DETECTOR

A. LAIP Detector

Because the read back value $y(i,j)$ of bit cell (i,j) is the area integral of weighted magnetizations of all grains contained within the 3×3 bit cells surrounding (i,j), the read back value magnitude $|y|$ has an upper limit, which depends on the Gaussian kernel parameter T_{50}. For grain densities from 1 to 3 GPB, y ranges approximately between -2 and $+2$. This is similar to but different from [3], because the read back signal in [3] integrates only over single bit cells with no ISI kernel function, which makes the y values range from exactly -1 to $+1$. As each read back value is influenced by grains magnetized by its surrounding eight bit cells, we define the integral of the magnetization weighted by the impulse response over the area magnetized by a given surrounding bit as the *local area influence* (LAI) on target bit U due to that bit.

The colored areas in Fig. 4 illustrate the grain areas magnetized by surrounding bits A and E that affect the read value y_U of central bit U. We denote the integral of the magnetization weighted by the impulse response over these "affected areas" as the LAIs α_A, α_B, α_C, α_D, α_E, α_F, α_G and α_H, and we define the total LAI on the given bit U as:

$$\alpha_{\text{total}} = \alpha_A + \alpha_B + \alpha_C + \alpha_D + \alpha_E + \alpha_F + \alpha_G + \alpha_H. \quad (3)$$

We define α_U as the part of reading y_U due only to bit U. Thus, $y_U = \alpha_U + \alpha_{\text{total}}$, so that correct estimation of U amounts to correct estimation of $\text{sign}(\alpha_U) = \text{sign}(y_U - \alpha_{\text{total}})$. The key idea is that if y_U is significantly greater (less) than α_{total}, then bit U is most likely $+1$ (-1). But if y_U and α_{total} are approximately equal, then it is likely that bit U is overwritten.

Following this idea, given all the read values, if we can estimate α_{total}, we can estimate the bit U or its log likelihood ratio (LLR). In order to estimate α_{total}, we train a pre-computed table of α_{total}'s discretized probability mass function (PMF). The ideal table should have entries corresponding to all $3 \times N$ read values. However, if we discretize each read value to 40 bins (with y ranging from around -2 to $+2$, the precision should be at least $4/40 = 0.1$ to work well experimentally), then we need to train a PMF with $40^{3\times N}$ bins, which is impractical for typical values of N of around 32000. Therefore, we select a limited number of read back values and train several different, and much smaller, conditional PMFs.

Define y_A, y_B, …, y_H, as the read values from bits A through H adjacent or diagonal to the kth target bit U_k, as shown in Fig. 4. For brevity, we omit the subscript k in the following discussion. The following procedure is used to create and train the conditional PMF of $P(\alpha_A + \alpha_E | y_A, y_E, y_U, u_A, u_E, u_U)$ for use with the LAIP detector. This PMF accounts for the combined influences of bit A and its adjacent diagonal bit E; in our previous paper [3], the effects of diagonal bits are ignored. This conditional PMF is not specific to a particular target bit U; the PMF is computed once and then used to estimate all bits U that occur in the central track. Similar PMFs are trained for bits that occur in the two tracks adjacent to the boundary.

1) Through simulations of writing and reading the media model, we compute many 4-tuples $(y_A, y_E, y_U, \alpha_A + \alpha_E)$, or similar (and symmetric) tuples like $(y_A, y_F, y_U, \alpha_A + \alpha_F)$, $(y_B, y_G, y_U, \alpha_B + \alpha_G)$, etc. For each gathered 3-tuple (y_A, y_E, y_U), we compute $\alpha_A + \alpha_E$ using (1) with a modified integration region

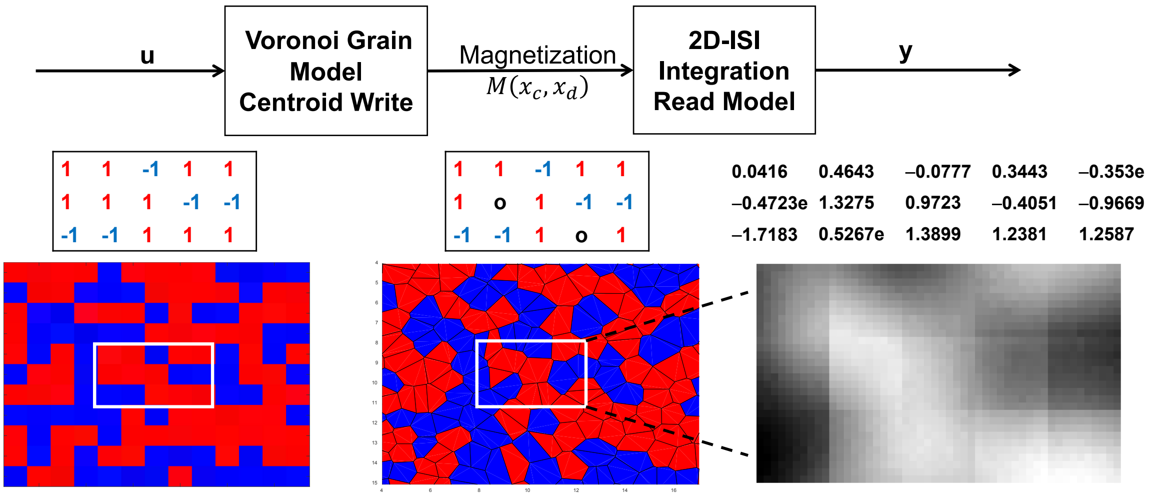
u
Voronoi Grain Model Centroid Write
Magnetization $M(x_c, x_d)$
2D-ISI Integration Read Model
y
1 1 -1 1 1
1 1 1 -1 -1
-1 -1 1 1 1
1 1 -1 1 1
1 o 1 -1 -1
-1 -1 1 o 1
0.0416 0.4643 −0.0777 0.3443 −0.353e
−0.4723e 1.3275 0.9723 −0.4051 −0.9669
−1.7183 0.5267e 1.3899 1.2381 1.2587

Fig. 3. Example of the read write process with 2D-ISI integration. The 3-row-5-column square input bits u (circled in white) are binary and marked red (+1) or blue (−1). After being written on random Voronoi grains with the centroid write model, some bits are missing (not written on any grain) and marked with a black 'o'. Next, the read back values y are calculated from the 2D-ISI integration read model, and displayed using a gray-scale image of resolution 10×10 pixels per bit. Each pixel represents the read back signal as if the reader center is exactly at the pixel location. Numerical readback values at the center of each bit are provided; the values corresponding to erroneous bits are marked with an 'e'. Although none of the missing bits have errors in this example, in practice they are more likely to suffer detection errors.

S that includes all grains with centroids contained in bit cells A or E.

2) We create a $40 \times 40 \times 40$ grid of bins for the (y_A, y_E, y_U) 3-tuples using Lloyd-Max quantization [40], [41]. We denote the index of the bin corresponding to a given 3-tuple of readings (y_A, y_E, y_U) as $\mathrm{ix}_3(y_A, y_E, y_U)$, such that $0 \le \mathrm{ix}_3(y_A, y_E, y_U) \le 63999$.
3) The $\alpha_A + \alpha_E$ values corresponding to all the 3-tuples in each bin $\mathrm{ix}_3(y_A, y_E, y_U)$ are used to populate bin counts among 41 equal bins spanning values −2 to 2; the index of these α bins is denoted $\mathrm{ix}_\alpha(\alpha_A + \alpha_E)$, with $0 \le \mathrm{ix}_\alpha(\alpha_A + \alpha_E) \le 40$.
4) The counts in the α bins are normalized to sum to 1, thereby creating the conditional PMF $P(\mathrm{ix}_\alpha(\alpha_A + \alpha_E)|\mathrm{ix}_3(y_A, y_E, y_U))$. Because this PMF does not use *a priori* information, it is used only during the initial iteration of the turbo-detector.
5) To incorporate *a priori* information from the channel decoder on subsequent turbo-detector iterations, we run simulations to create conditional PMFs $P(\mathrm{ix}_\alpha(\alpha_A + \alpha_E)|\mathrm{ix}_3(y_A, y_E, y_U), u_A, u_E, u_U)$, where u_A, u_E, u_U are input bits which take values ±1.

After the procedure above, we have 64000 conditional PMFs of affected area $\alpha_A + \alpha_E$ given (y_A, y_E, y_U) tuples and 512000 conditional PMFs given $(y_A, y_E, y_U, u_A, u_E, u_U)$ tuples. Each conditional PMF has 41 bins. Sometimes, we don't get 4-tuples $(y_A, y_E, y_U, \alpha_A + \alpha_E)$ or 7-tuples $(y_A, y_E, y_U, u_A, u_E, u_U, \alpha_A + \alpha_E)$ to fall in all the 64000 or 512000 separate PMFs; we call this the zero-count case. In this case, we simply assign probability 1 to the middle 21st bin of the affected area which corresponds to value 0, and probability 0 to all the other bins.

Due to the assumption that input bits at the boundary are known but read values at the boundary are unknown, when target bit U is in the middle row of the last column N (so that bit E is a known boundary bit), we need to train two additional conditional PMFs of $\alpha_A + \alpha_E$ given (y_A, y_U, u_E), and given $(y_A, y_U, u_A, u_U, u_E)$. Similarly, when bit U is in the top row bits A and E are known boundary bits, so we train conditional PMF tables given (y_U, u_A, u_E), and given (y_U, u_U, u_A, u_E). We choose the number of bins to be 40 and 41 in order to balance simulation time and performance. It takes time to train the tables, but it's very fast to use the tables to do real-time LAIP estimation. Accuracy of the PMF tables (compared to the true conditional PMFs) improves if we have sufficient bins and sufficient simulations. We collected at least 80 million tuples for the conditional PMF tables.

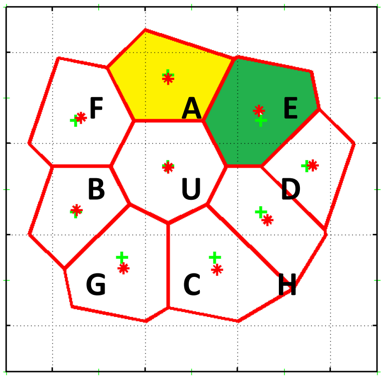

Fig. 4. Areas magnetized by adjacent bits A and E (under the centroid write model) that affect the central coded bit U. The yellow portion is the affected area from bit A to bit U, and the green is the affected area from bit E to bit U. The LAIs α_A and α_E are computed by integrating the magnetization weighted by the impulse response in (2) over these affected areas. In this example, there is approximately one coded bit per magnetic grain (1 GPB).

The PMFs should in theory be anti-symmetric with respect to the conditioning y variables, e.g., $P(\mathrm{ix}_\alpha(\alpha_A + \alpha_E)|\mathrm{ix}_3(y_A, y_E, y_U)) = P(\mathrm{ix}_\alpha(-(\alpha_A + \alpha_E))|\mathrm{ix}_3(-y_A, -y_E, -y_U))$. To make the collected PMFs more anti-symmetric, we use $7 \times 7 \times 7$ symmetric spatial filters to smooth the PMFs. This smoothing improves the LAIP detector's performance, especially when the number of tuples is not enough to sufficiently populate all bins. The discretization of the read back values is performed using Lloyd-Max quantization [40], [41] in order to evenly collect the tuples in case of too many zero-count cases. The affected area bins should be odd in number (e.g., our choice of 41) and evenly spaced, which helps in the zero-count case and in the convolution calculation below. Example 3D plots of conditional PMFs trained with random Voronoi data appear in Figs. 3 and 4 of [3]; these plots do not include the influences of diagonal bits E, F, G or H.

To compute the discrete PMF $P(\mathrm{ix}_y(\alpha_{\mathrm{total}}))$, the LAIP detector looks up the appropriate PMFs and then computes

$$\begin{aligned}
&P(\mathrm{ix}_y(\alpha_{\mathrm{total1}})) = \\
&\quad P(\mathrm{ix}_\alpha(\alpha_A+\alpha_E)|\mathrm{ix}_3(y_A,y_E,y_U)) \\
&\quad * P(\mathrm{ix}_\alpha(\alpha_D+\alpha_H)|\mathrm{ix}_3(y_D,y_H,y_U)) \\
&\quad * P(\mathrm{ix}_\alpha(\alpha_C+\alpha_G)|\mathrm{ix}_3(y_C,y_G,y_U)) \\
&\quad * P(\mathrm{ix}_\alpha(\alpha_B+\alpha_F)|\mathrm{ix}_3(y_B,y_F,y_U)), \\
&P(\mathrm{ix}_y(\alpha_{\mathrm{total2}})) = \\
&\quad P(\mathrm{ix}_\alpha(\alpha_A+\alpha_F)|\mathrm{ix}_3(y_A,y_F,y_U)) \\
&\quad * P(\mathrm{ix}_\alpha(\alpha_B+\alpha_G)|\mathrm{ix}_3(y_B,y_G,y_U)) \\
&\quad * P(\mathrm{ix}_\alpha(\alpha_C+\alpha_H)|\mathrm{ix}_3(y_C,y_H,y_U)) \\
&\quad * P(\mathrm{ix}_\alpha(\alpha_D+\alpha_E)|\mathrm{ix}_3(y_D,y_E,y_U)), \\
&P(\mathrm{ix}_y(\alpha_{\mathrm{total}})) = \frac{P(\mathrm{ix}_y(\alpha_{\mathrm{total1}}))+P(\mathrm{ix}_y(\alpha_{\mathrm{total2}}))}{2},
\end{aligned} \tag{4}$$

where $*$ indicates discrete 1D convolution of the relevant conditional PMFs. After the convolution operations, the bin indices $\mathrm{ix}_y(\alpha_{\mathrm{total}j}), j \in \{1,2\}$ range from 0 to 160, and span a nominal range of $-8 \le \alpha_{\mathrm{total}j} \le 8$. All the eight conditional PMFs are read from the same $P(\mathrm{ix}_\alpha(\alpha_A + \alpha_E)|\mathrm{ix}_3(y_A, y_E, y_U))$ table by symmetry. Eq. (4) assumes statistical independence between the four index variables $\mathrm{ix}_\alpha(\alpha_A + \alpha_E)$, $\mathrm{ix}_\alpha(\alpha_D + \alpha_H)$, etc. involved in each of the two convolution computations for $P(\mathrm{ix}_y(\alpha_{\mathrm{total1}}))$ and $P(\mathrm{ix}_y(\alpha_{\mathrm{total2}}))$. These independence assumptions are approximations; they are necessary because full computation and storage of the PMF table of α_{total} including the joint influences of all eight surrounding bits requires excessive memory. The pairwise statistical independence assumption allows us to compute $P(\mathrm{ix}_y(\alpha_{\mathrm{total}}))$ by pairwise convolutions that proceed in clockwise (counter-clockwise) order for $P(\mathrm{ix}_y(\alpha_{\mathrm{total1}}))$ ($P(\mathrm{ix}_y(\alpha_{\mathrm{total2}}))$); the final estimation of $P(\mathrm{ix}_y(\alpha_{\mathrm{total}}))$ is obtained by averaging $P(\mathrm{ix}_y(\alpha_{\mathrm{total1}}))$ and $P(\mathrm{ix}_y(\alpha_{\mathrm{total2}}))$. Eq. (4) does not make use of *a priori* information, so it's only used in the first iteration of the LAIP detector, when *a priori* bit LLRs $u_A, u_B, \ldots, u_U$ are unavailable.

The PMF convolution computations require that the bin edge boundaries (defining the range of $\alpha_{\mathrm{total}j}$ that lies within each bin) of the convolved pairwise PMFs be equally spaced. In practice, the convolutions are efficiently computed and are fast enough for the required real-time estimation of $P(\mathrm{ix}_y(\alpha_{\mathrm{total}}))$. Because the expanded range $-8 \le \alpha_{\mathrm{total}j} \le 8$ is an artifact of the independence assumption on the index variables, the equally spaced bin edge boundaries are divided by 4 to remap the $\alpha_{\mathrm{total}j}$ PMFs onto the range $-2 \le \alpha_{\mathrm{total}j} \le 2$ that would occur if the true joint distributions were used.

When *a priori* information is unavailable from the channel decoder at the first iteration, a simplified binary output LLR for the coded bit U can be computed as follows:

$$LLR(U) = \log \frac{P(\mathrm{ix}_y(\alpha_{\mathrm{total}}) < \mathrm{ix}_y(y_U))}{P(\mathrm{ix}_y(\alpha_{\mathrm{total}}) > \mathrm{ix}_y(y_U))}. \tag{5}$$

In this expression, $\mathrm{ix}_y(y_U)$ is computed by dividing the range $-2 \le y_U \le 2$ into 161 bins of equal width, and then computing the index of the bin that the given y_U reading falls into. The LLR in (5) is approximately equal to the *a posteriori probability* (APP) LLR $\log[P(U = +1|y_U, y_A, \ldots, y_H)/P(U = -1|y_U, y_A, \ldots, y_H)]$. Since the probability P_{ovw} that bit U is overwritten is $P_{\mathrm{ovw}} = P(\mathrm{ix}_y(\alpha_{\mathrm{total}}) = \mathrm{ix}_y(y_U))$, the LAIP detector actually computes the following modified version of (5) to account for the overwrite case:

$$LLR(U) = \log \frac{P(\mathrm{ix}_y(\alpha_{\mathrm{total}}) < \mathrm{ix}_y(y_U)) + P_{\mathrm{ovw}}/2}{P(\mathrm{ix}_y(\alpha_{\mathrm{total}}) > \mathrm{ix}_y(y_U)) + P_{\mathrm{ovw}}/2}. \tag{6}$$

It is reasonable and beneficial to avoid overflow that half of the overwritten probability corresponds to $P(U = +1|y_U, y_A, \ldots, y_H)$ and the other half to $P(U = -1|y_U, y_A, \ldots, y_H)$.

After the first iteration, *a priori* information is available from the channel decoder, and the LAIP detector computes conditional $\alpha_A + \alpha_E$ PMFs as

$$\begin{aligned}
&P(\mathrm{ix}_\alpha(\alpha_A+\alpha_E)|\mathrm{ix}_3(y_A,y_E,y_U)) = \\
&\quad \sum_{u_A,u_E,u_U} [P(\mathrm{ix}_\alpha(\alpha_A+\alpha_E)|\mathrm{ix}_3(y_A,y_E,y_U),u_A,u_E,u_U) \\
&\qquad\qquad \times P(u_A,u_E,u_U)]
\end{aligned} \tag{7}$$

where $P(u_A, u_E, u_U) = P_{\mathrm{in}}(u_A)P_{\mathrm{in}}(u_E)P_{\mathrm{in}}(u_U)$, and the incoming *a priori* probabilities from the channel decoder are denoted $P_{\mathrm{in}}(u_A)$, $P_{\mathrm{in}}(u_E)$, and $P_{\mathrm{in}}(u_U)$. Then we can reuse (4) and (6) to compute $P(\mathrm{ix}_y(\alpha_{\mathrm{total}}))$ and the LLRs respectively.

B. LAIP and BCJR turbo TDMR Detector

The LAIP detector estimates LLRs for the coded bits by accounting for local interactions between the target bit U and grains in its immediate surrounding eight-bit neighborhood. However, the storage required for the conditional PMF tables makes it impractical for the LAIP detector to consider longer range neighborhoods. To account for the effect of longer range interactions caused by ITI and ISI, we propose to process the $3 \times N$ array of readings $\mathbf{y}$ by a 2D-BCJR detector with N trellis stages, and use LLRs from the LAIP detector as *a priori* information for the BCJR. This section describes the combined LAIP and BCJR turbo detector, including the interaction of both LAIP and BCJR detectors with the IRA decoder.

Fig. 5 shows a block diagram of the proposed LAIP/BCJR TDMR turbo-detector. Information bits are encoded into channel bits $\mathbf{u}$ by an IRA encoder that encodes each of the three rows into separate code words. After passing through a row-by-row bit-wise interleaver (denoted π), the channel bits $\mathbf{u}$ flow into the random Voronoi grain write/read model, resulting in $3 \times N$ sample array $\mathbf{y}$, which flows into the LAIP detector. The LAIP detector jointly considers all three rows and computes LLRs following (6). Then the LAIP detector

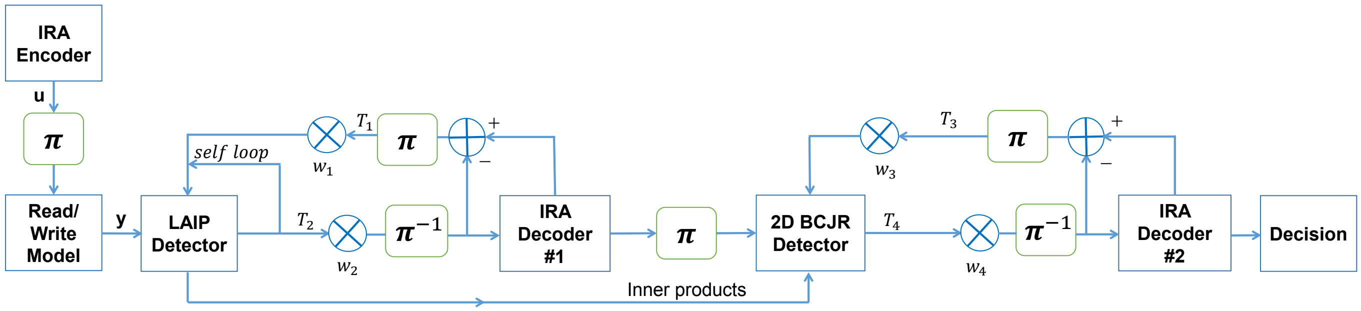

Fig. 5. Block diagram of the LAIP/BCJR TDMR turbo-detector.

output LLRs are fed back as *a priori* inputs for subsequent iterations of the LAIP detector. This self-loop process iterates until the LLRs converge; typically five self-loops are sufficient. Next, the LLRs flow through a row-wise deinterleaver and then into the first IRA decoder where they are decoded row-by-row; several iterations of coded bit LLRs between the first IRA encoder and the LAIP detector are done. Then the output LLRs from the first IRA decoder enter the BCJR detector. The LAIP detector also sends the BCJR local estimates of the 2D inner products needed to compute the expected trellis branch outputs at each trellis stage; this accounts for the effect of local grain-bit interactions on the effective 2D-ISI mask assumed by the BCJR detector. The BCJR detector exchanges coded bit LLRs (jointly estimated over all three rows) with the second row-by-row IRA decoder, and the final decision is made on the second IRA decoder's information bit output LLRs. We note that the first and second IRA decoders use the same code (corresponding to the IRA encoder).

The multiplicative LLR weights $w_1, \ldots, w_4$, $0 < w_i \leq 1$, slow the convergence of the turbo-equalizer, thereby lowering the final bit error rate (BER). The thresholds $T_1, \ldots, T_4$ restrict the maximum absolute values for LLRs from the IRA decoders and the LAIP and BCJR detectors in order to prevent numerical under- or overflow. Subtraction of the incoming LLRs to the LAIP detector is not done at the detector's output, because the processing done in equations (7) and (4) prevents expressing the input LLRs as a separate additive term in (6). Similarly, marginalization operations done in the 2D-BCJR detector also prevent separation of the detector's output LLRs into a sum of the input LLRs and extrinsic LLRs [5]; hence, LLR subtraction is also not done at the BCJR detector output.

The LAIP detector self-loops significantly reduce the BER of the LAIP output LLRs, which in turn greatly reduces the BER of the first IRA decoder. We find experimentally that the proposed architecture works better than an alternative scheme that iteratively exchanges LLRs directly between the LAIP and BCJR detectors, because the LAIP detector's output LLRs (even with self-loops) are not accurate enough to significantly lower the BCJR's BER when used as *a priori* inputs. The reason might be that the BCJR's LLR estimates are based on the entire $3 \times N$ block of readings, whereas the LAIP detector's estimates are based on 3×3 neighborhoods. We also find that the set of erroneous bit positions from the LAIP detector alone has about 70% overlap with the set of erroneous positions from the BCJR detector alone (without *a priori* input from the LAIP detector); thus, the LAIP can help the BCJR correct about 30% of its erroneous bits, which helps explain why the two detectors together outperform either one alone.

C. Inner Product Prediction

Conventional 2D-ISI detectors, such as those in, e.g., [5], [42], [43], assume that the readings $y(m,n)$ from the $3 \times N$ array follow the model

$$y(m,n) = \overline{y}(m,n) + w(m,n), \tag{8}$$

where $w(m,n)$ is discrete additive white Gaussian noise (AWGN) with spatially invariant statistics, and $\overline{y}(m,n)$ is given by the 2D convolution of the coded bit array $u(m,n)$ with a discrete, finite extent, and spatially invariant mask $h(k,l)$:

$$\overline{y}(m,n) = \sum_{k,l} h(k,l)u(m-k,n-l). \tag{9}$$

The convolution mask $h(k,l)$ is typically a sampled version of a continuous read-head response such as (2), and is typically assumed to be rectangular (e.g., $-1 \leq (k,l) \leq 1$ for a 3×3 mask). In 2D-BCJR detectors such as [5], [43], the gamma state transition probability $P(y(m,n)|U = i, S_k = s, S_{k-1} = s^{'})$ is assumed to be Gaussian distributed with mean $\overline{y}(m,n)$ computed according to (9). For fixed (m,n), (9) can be viewed as a 2D inner product between the 2D spatially-invariant mask $\mathbf{h}$ and the surrounding neighborhood $\mathcal{N}_{m,n}$ of bits $u(m-k,n-l)$ defined by the trellis input U and current and previous states S_k and S_{k-1}.

Most recently proposed TDMR detectors (e.g., [26]–[29]) adopt the modeling convention widely used in 1D MR detectors, i.e., they add a spatially varying media noise term $\nu(m,n)$ to the 2D-ISI model of (8) and (9): $y(m,n) = \overline{y}(m,n) + \nu(m,n) + w(m,n)$. They then integrate noise prediction into multi-track trellis detectors, so that by predicting $\nu(m,n)$ and subtracting it from $y(m,n)$, the resulting corrected reading closely approximates $\overline{y}(m,n) + w(m,n)$.

By contrast, the 2D-BCJR detector proposed here does not employ integrated media noise prediction. Instead, it relies on the LAIP detector to provide *a priori* LLRs that depend on the local grain geometries around recorded bits. The LAIP detector employs read back value dependent PMFs to estimate the expected read value (which is $\alpha_{\text{total}} + \alpha_U$). This approach can be viewed as computing $\overline{y}(m,n)$ as the 2D inner product between a *spatially varying* 2D mask $\mathbf{h}_{m,n}$ and the bits in $\mathcal{N}_{m,n}$. Thus, the proposed model is

$$\begin{aligned} y(m,n) &= \overline{y}(m,n) + w(m,n), \\ \overline{y}(m,n) &= (\mathbf{h}_{m,n} * \mathbf{u})(m,n), \end{aligned} \tag{10}$$

where $*$ indicates 2D convolution, and $\mathbf{u}$ is the $3 \times N$ array of coded bits. This model accounts for the spatially varying media effects through $\mathbf{h}_{m,n}$ instead of through an additive media noise term $\nu(m,n)$.

To design a 2D-BCJR detector consistent with the model proposed in (10), we train an inner product prediction table which stores expected read back values $\overline{y}_U$ conditioned on surrounding read back values and bits. The 2D-BCJR detector uses this table to look up $\overline{y}_U$ based on the target bit U's bit neighborhood $\mathcal{N}_U$ associated with a given trellis branch, and also on bit U's reading y_U as well as its surrounding readings $(y_A, \ldots, y_H)$. The table training procedure is as follows:

1) Through simulations of writing and reading the media model, many tuples of length 14 $(y_U, y_A, \ldots, y_H, u_A, u_B, u_C, u_D, u_U)$ are gathered.
2) The eight surrounding read back values $(y_A, \ldots, y_H)$ are quantized on a 3^8 grid, which is computed by Lloyd-Max quantization. Thus each surrounding read back value is quantized to discretized bins of length 3.
3) All the y_Us collected in the same conditional bin $(y_A, \ldots, y_H, u_A, u_B, u_C, u_D, u_U)$ are averaged, and the averaged value is used as the expected value $\overline{y}_U$.

Above describes how to train a table of expected read back values $\overline{y}_U$ conditioned on surrounding eight read back values and five adjacent bits. The discretization precision of three is chosen to limit the required storage. The total size of this table is $3^8 \times 2^5 = 209952$, which is less than the LAIP detector's conditional LAI PMF tables. Similar to the LAIP detector, smaller tables are trained to deal with the case that target bit U is at the border of the $3 \times N$ bit block; for example, the top border table stores $(\overline{y}_U | y_B, y_C, y_D, y_G, y_H, u_A, \ldots, u_D, u_U)$.

Table I displays the performance of inner product prediction in the system shown in Fig. 5 under several cases. The three error rates in the BER columns come from the top, middle and bottom rows of the $3 \times N$ readings array respectively, and are BCJR detector error rates after the first iteration (and therefore do not include any *a priori* information from the IRA decoder). The table shows that inner product prediction can decrease the BCJR detector's BER by about 7% for the GFP model and 15% for the 2.0 GPB Voronoi model.

D. LAIP Detector for GFP Model

The GFP model of [6] is a more realistic TDMR model than the Voronoi model. Therefore, this paper also utilizes GFP waveform data from the Data Storage Institute (DSI) of Singapore to validate the proposed system's performance. Each GFP data block has three tracks with 3×41207 input bits and corresponding read back values; the simulated track pitch (TP) is 18 nm, the bit length (BL) is 11 nm, and the media has 3.491 GPB. Each of the three rows (tracks) has a different set of random input bits (not produced by a channel encoder). Thus, the initial IRA encoder, interleaver, and write/read model blocks in Fig. 5 are not used; instead, the 3×41207 GFP data flows directly into the LAIP and BCJR detectors as readback data $\mathbf{y}$. The deinterleaver and interleaver within the LAIP and BCJR turbo loops are used, and the IRA decoder uses coset decoding to decode its received LLRs.

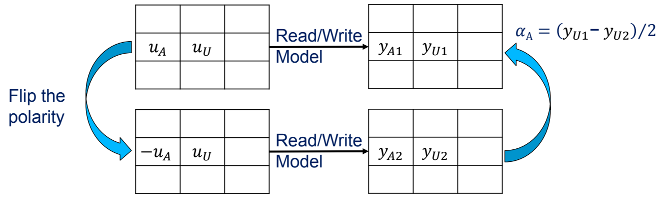

Fig. 6. Designed training pattern approach for GFP model. Pairs of bits (u_A, u_U) are written, and the sample y_{U1} for target bit U is read back. Then $(-u_A, u_U)$ is written at the exact same location on the simulated media, and the sample y_{U2} for target bit U is read back again. As the difference between the two read back values is caused by the change to bit A, the affected area α_A of A on U can be estimated as $(y_{U1} - y_{U2})/2$.

In the GFP model the bit cell is rectangular instead of square. Therefore, the symmetry around square bits assumed by the Voronoi model does not hold, which means, e.g., that $P(\mathrm{ix}_\alpha(\alpha_A + \alpha_E)|\mathrm{ix}_3(y_A, y_E, y_U))$ does not have the same PMF as $P(\mathrm{ix}_\alpha(\alpha_B + \alpha_G)|\mathrm{ix}_3(y_B, y_G, y_U))$ Due to the shingled writing characteristics, $P(\mathrm{ix}_\alpha(\alpha_A + \alpha_E)|\mathrm{ix}_3(y_A, y_E, y_U))$ does not have the same PMF as $P(\mathrm{ix}_\alpha(\alpha_A + \alpha_F)|\mathrm{ix}_3(y_A, y_F, y_U))$ either. Hence, the original conditional PMF table of $P(\mathrm{ix}_\alpha(\alpha_A + \alpha_E)|\mathrm{ix}_3(y_A, y_E, y_U))$ is split into 8 separate tables as $P(\mathrm{ix}_\alpha(\alpha_A + \alpha_E)|...)$, $P(\mathrm{ix}_\alpha(\alpha_A + \alpha_F)|...)$, $P(\mathrm{ix}_\alpha(\alpha_B + \alpha_F)|...)$, $P(\mathrm{ix}_\alpha(\alpha_B + \alpha_G)|...)$, ..., and $P(\mathrm{ix}_\alpha(\alpha_D + \alpha_E)|...)$. For the same reason, the inner product prediction needs to train an extra table for the conditional inner products at the bottom row, which is $(\overline{y}_U | y_A, y_B, y_D, y_F, y_E, u_A, \ldots, u_D, u_U)$.

1) Designed Patterns for Training

Because the GFP model directly provides read back waveforms $\mathbf{y}$ and input data bits $\mathbf{u}$ and does not provide information about the underlying grain geometries, the affected area α cannot be directly computed as it is with the Voronoi model. Thus, special training patterns are used to collect the αs as shown in Fig. 6.

In practice, 512 different 3×3 patterns of input bits for $(u_A, \ldots, u_H)$ are designed, ranging from 0 to 511. For each 3×3 bit pattern, the pattern plus one 3×1 bit random guard band column (i.e., 3×4 bits) repeatedly is written on and read from the simulated media 10301 times, with two initial random guard bands and one final guard band, for a total of 3×41207 readings on three media tracks per pattern file. The GFP simulation writes each of the 3×41207 input bit pattern files on exactly the same underlying simulated grain pattern at exactly the same starting location. There are ten separate 3×41207 readings files for each of the 512 bit patterns, in order to account for the random grain flipping that occurs during the GFP simulated write process. The guard bands are the same for each pattern file. The guard bands protect the computed αs from being influenced by adjacent 3×3 bit patterns.

The readback signal from the GFP model is noisy as grains are flipped through the use of a random number generator, thus multiple affected areas computed as in Fig. 6 are averaged at

TABLE I
BER PERFORMANCE OF BCJR DETECTOR'S INNER PRODUCT PREDICTION ON DATA TRACKS 1, 2 AND 3

Test Case	BER without IP Prediction (Tr1 Tr2 Tr3)	BER with IP Prediction	Improvement Percentage
0.8235 code rate, Voronoi Model at 2.0 GPB	1.73% 2.06% 1.62%	1.51% 1.75% 1.33%	12.97% 15.29% 17.91%
0.7846 code rate, Voronoi Model at 1.7 GPB	3.99% 4.18% 4.08%	3.66% 3.84% 3.79%	8.39% 8.15% 7.00%
0.6971 code rate, GFP model at TP18 BL11	4.28% 4.28% 3.93%	3.92% 3.95% 3.76%	8.59% 7.72% 4.23%

TABLE II
3×3 2D-ISI MASKS FOR THE VORONOI AND GFP MODELS ESTIMATED BY AVERAGING THE α LAIS (FIRST THREE COLUMNS), AND VIA THE LEAST SQUARES METHOD (LAST COLUMN)

2.0 GPB Voronoi Model	1.7 GPB Voronoi Model	GFP TP18 BL11	GFP TP18 BL11 (LS method)
0.0591 0.1985 0.0584	0.0818 0.2371 0.0817	0.1995 0.2275 0.0620	0.1937 0.2260 0.0530
0.1985 0.6589 0.1960	0.2378 0.6854 0.2381	0.2810 0.5380 0.1727	0.2825 0.5378 0.1659
0.0586 0.1970 0.0581	0.0817 0.2384 0.0827	0.0291 0.1450 0.0564	0.0245 0.1457 0.0627

the same grain locations. For example, 128 $\alpha_A+\alpha_E$ estimates from 512 patterns at the same grain location are averaged to train a conditional frequency table; every $\alpha_A+\alpha_E$ is estimated by flipping bits A and E and computing $(y_{U1}-y_{U2})/2$ while keeping the other 7 bits unchanged.

Using the designed patterns, affected area tables for the GFP model can be trained, and the estimated values of $\alpha_A, \alpha_E, ...$ can be averaged over all 3×3 bit patterns to create an estimated 2D-ISI mask as in the third column of Table II. The 3×3 masks in the first two columns of this table are computed by averaging αs computed for the Voronoi model over all possible 3×3 input bit combinations, while the last column is estimated by using the least squares (LS) method on the GFP data set. In employing the LS method, the GFP readback data is scaled by dividing by half the maximum reading magnitude, so that the scaled data is limited to ±2. The masks for the Voronoi models should be symmetric; the slight asymmetries are due to insufficient training data. The table reveals that low GPB values lead to higher ISI and ITI in Voronoi grain models, and that the GFP model has much more ISI and ITI than the Voronoi model. The LS estimated mask is almost the same as the LAIP detector trained mask, which verifies experimentally that the GFP bit-pattern-file training approach is correct. An important difference between the LAIP and LS methods is that LAIP computes the influence from one bit to another bit through the grain state, and so it has physical meaning. The difference between LAIP and traditional approaches like BCJR is that LAIP uses conditional frequency tables instead of a fixed mask.

2) *Interleaver*

The system in Fig. 5, when tested with the Voronoi read/write model, proved insensitive to choice of interleaver; for that reason, the identity interleaver (with same order at output as input) is used to produce the Voronoi model simulation results in section IV. By contrast, for the GFP model, experiments show that the IRA decoder is quite sensitive to error positions at either the LAIP or BCJR detector output, indicating that significant correlation is present among the GFP model error patterns. This fact is consistent with the much higher average level of 2D-ISI present in the GFP data, as seen in Table II. Experiments also show that the decoded BER is significantly reduced when the error locations are decorrelated by an appropriately designed interleaver. For the GFP model simulation results in section IV, a per row S-interleaver is used [44], such that bits adjacent at each row's input are separated by at least $S = 30$ positions at the output.

3) *Expected Read Back Value* $E[y_U]$

In training the conditional affected area PMFs for the GFP model, we estimate αs rather than directly compute them as in the Voronoi model. In the GFP training we obtain an affected area that is averaged over the probabilistic grain polarity flipping, which occurs even when identical bit patterns are written at the exact same location on the simulated media. Thus in Fig. 6, the averaged α is obtained by averaging over the read back values (y_{U1}, y_{U2}). This suggests that, when computing LLRs for the GFP data with (6), α_{total} computed from (3) should actually be compared to an *expected* read back value $E[y_U]$ rather than to the read back value y_U for target bit U. Therefore, we train an additional conditional PMF $P(\text{ix}_\alpha(E[y_U])|\text{ix}_\alpha(y_U), u_U, u_A, u_B, u_C, u_D, u_F)$, where $E[y_U]$ and y_U are both quantized to 41 bins equally spaced on the interval $[-2, 2]$, and $u_U, u_A, \ldots, u_D, u_F$ are coded bit estimates received by the LAIP detector as *a priori* information from the channel decoder. We find experimentally that conditioning on diagonal bit u_F improves the derived LLR reliability, probably because of the non-symmetrical bit shapes due to shingled writing. Similar to (6), define LLR_k as

$$LLR_k = \log \frac{P(\text{ix}_\alpha(\alpha_{\text{total}}) < k) + P_{\text{ovw},k}/2}{P(\text{ix}_\alpha(\alpha_{\text{total}}) > k) + P_{\text{ovw},k}/2}, \tag{11}$$

where $P_{\text{ovw},k} = P(\text{ix}_\alpha(\alpha_{\text{total}}) = k)$ and $0 \le k \le 40$. In (11), the central 41 bin indices $(60, \ldots, 100)$ of the 161 bin PMF $P(\text{ix}_y(\alpha_{\text{total}}))$ are mapped to the range $0 \le \text{ix}_\alpha(\alpha_{\text{total}}) \le 40$; this mapping is done before the range of the PMF computed in (4) is re-scaled by $1/4$, so that the central 41 bins span the range $-2 \le \alpha_{\text{total}} \le 2$. As the area under the central part of the the PMF contains most of the probability, this is a reasonable approximation.

Using (11), the modified LLR for coded bit U is computed by taking the expectation of LLR_k over the conditional PMF for $E[y_U]$ and the *a priori* probabilities $P(u_U), P(u_F), P(u_A), \ldots, P(u_D)$ computed from the LLRs received from the channel decoder:

$$LLR(U) = \sum_{\mathcal{N}_u}\sum_{k} LLR_k P(\mathrm{ix}_\alpha(E[y_U]) = k|\mathrm{ix}_\alpha(y_U), \mathcal{N}_u)P(\mathcal{N}_u), \tag{12}$$

where $\mathcal{N}_u = \{u_U, u_F, u_A, \ldots, u_D\}$, and the approximation $P(\mathcal{N}_u) \approx P(u_U)P(u_F)P(u_A)\cdots P(u_D)$ is used. By using (12), the LAIP detector's output BER is slightly reduced when the LLRs from the channel decoder are not very reliable (e.g., in early iterations between the LAIP detector and the IRA decoder), but greatly reduced when the LLRs are highly reliable (as in later iterations).

E. Summary of the LAIP/BCJR Detection Algorithm

In this subsection, we present a high-level overview of the LAIP/BCJR detection algorithm. It is helpful to refer to Fig. 5 for an overview of the real-time processing phase.

1) Offline Training

In this phase, the LAIP PMF tables are trained according to the procedures outlined in subsections III.A, III.C, and III.D. A single set of tables is trained, and used to process all samples during the real-time processing phase. During training the αs for the Voronoi model are computed by numerical integration as described in step 1 of the training procedure in subsection III.A; for the GFP model, they are computed by the procedure illustrated in Fig. 6. The training data for both models is different than the data used for testing during real-time processing.

2) Real-Time Processing

We assume that *a priori* information from the IRA decoder is available to the LAIP.

LAIP Processing:

1) For each target bit U on each of the three tracks, readings y_U and $y_A, \ldots, y_H$ are collected; the lettering convention is shown in Fig. 4.
2) The readings are grouped into four pairs in the clockwise direction: (y_A, y_E), (y_D, y_H), (y_C, y_G), and (y_B, y_F).
3) For the (y_A, y_E) pair, eight conditional PMFs $P(\mathrm{ix}_\alpha(\alpha_A + \alpha_E)|\mathrm{ix}_3(y_A, y_E, y_U), u_A, u_E, u_U)$ are looked up in the PMF tables. These PMFs are marginalized over the values of the U, A, and E bits using (7), resulting in the pairwise conditional PMF $P(\mathrm{ix}_\alpha(\alpha_A + \alpha_E)|\mathrm{ix}_3(y_A, y_E, y_U))$.
4) Similar to the previous step, pairwise conditional PMFs are computed for the other three clockwise reading pairs.
5) The four clockwise pairwise conditional PMFs are convolved according to (4), in order to form the clockwise estimate $P(\mathrm{ix}_y(\alpha_{\mathrm{total1}})|y_U, y_A, \ldots, y_H)$.
6) The readings are grouped in the counter-clockwise direction as (y_A, y_F), (y_B, y_G), (y_C, y_H), and (y_D, y_E), and similar processing to the preceding three steps is performed to compute the counter-clockwise estimate $P(\mathrm{ix}_y(\alpha_{\mathrm{total2}})|y_U, y_A, \ldots, y_H)$.
7) The clockwise and counter clockwise estimates are averaged to form $P(\mathrm{ix}_y(\alpha_{\mathrm{total}}|y_U, y_A, \ldots, y_H)$.
8) An $LLR(U)$ for bit U is computed using (6) (for the Voronoi model) or (12) (for the GFP model).
9) The N $LLR(U)$s for all bits on each of the three tracks are sent to the IRA decoder, which decodes one N-bit codeword per track. The IRA decoder returns *a priori* LLRs $LLR_I(U)$ to the LAIP detector, and processing returns to step 3.
10) Once all LAIP/IRA decoder loops are finished, IRA decoder sends the $LLR_I(U)$s to the 2D-BCJR decoder.
11) For each bit U, using y_U and its four surrounding readings, the LAIP sends 32 inner product predictions $\bar{y}_U$ corresponding to the possible bit neighborhoods $(u_U, u_A, u_B, u_C, u_D)$ of U to the 2D-BCJR detector.

2D-BCJR Processing:

1) The BCJR uses the LAIP $\bar{y}_U$s as the conditional means for the gamma branch transition probabilities.
2) The BCJR uses the $LLR(U)$s from the LAIP to compute *a priori* probabilities which multiply the gamma probabilities.
3) The BCJR computes LLR estimates of the coded bits and passes them to the BCJR IRA decoder, which uses the identical code to the LAIP's IRA decoder.
4) The BCJR may exchange LLRs several times with its IRA decoder.
5) After the final BCJR/IRA iteration, final information bit decisions are made by the IRA decoder.

IV. Simulation Results

This section presents Monte Carlo simulation results for the LAIP/BCJR TDMR turbo-detector system. The simulation parameters in Fig. 5 are as follows. The multiplicative LLR weights $w_1, \ldots w_4$ between IRA decoders and LAIP and BCJR detectors are 0.5, 0.65, 0.65, and 1.0 respectively. The LLR thresholds $T_1, \ldots, T_4$ at the outputs of the detectors and decoders are 30, 5, 100, and 1.0 respectively. The number of internal IRA encoder iterations is 30 for the Voronoi simulations, and 100 for the GFP simulations. Four self loop iterations of the LAIP detector are done per each of four outer iterations between the LAIP and the first IRA decoder. Four iterations between the BCJR and the second IRA decoder are done in the Voronoi simulations, but no BCJR/IRA iterations are done for the GFP simulations, as most of the BER reduction in the GFP case occurs in the LAIP/IRA loop. These parameters were partially optimized via repeated simulations; it is likely that further optimization can still be made.

For each fixed value of GPB, the values of U/G reported for the simulations are arrived at by increasing the channel code rate r (via puncturing of parity bits) to the highest value possible such that the decoded BER $\leq 10^{-5}$, and then computing U/G $= r/$GPB. The IRA codes employed in the simulations are systematic, with each codeword of length $N_c = K + N_p$ consisting of K information bits followed by

N_p parity bits. N_c is selected to be as close as possible to the block length N of the readback values, while still maintaining $N_c \geq N$; the $N_c - N$ excess bits at the end of the codeword are punctured.

A. Simulation results for the Voronoi grain model

For the random Voronoi grain model, the block of coded input bits is of size 3×32000, and the performance of the proposed system in Fig. 5 is evaluated at both 1.7 and 2.0 GPB. For all Voronoi model simulations, the boundary bits are known by the receiver.

The puncturing method employed for the Voronoi model data is now described. For the Voronoi model, N_x punctured bits are selected randomly from the N_p parity bits. The punctured bits are not written onto the simulated medium, and hence the LAIP and BCJR detectors detect only the $N_w = N - N_x$ written bits, and the punctured code rate is $r = K/N_w$. For either detector, zero LLRs are inserted in the positions of the punctured bits so that N_c LLRs are sent to the IRA decoder.

Table III shows the simulation results for the proposed LAIP/BCJR turbo TDMR equalizer, on both the random Voronoi grain model (RVGM) and the GFP model. The BERs inside the parentheses are upper bound estimates with 95% confidence level, since there are 0 error counts in these cases. The BER upper bound is calculated as $3/N_w$, where N_w is the number of code bits transmitted [45].

Rows 3 and 4 of Table III are simulation results for a baseline detector consisting of only the BCJR detector and second IRA decoder, without the LAIP detector and first IRA decoder in Fig. 5. The achievable U/G for the RVGM model LAIP/BCJR system is 19.92% higher than baseline detector at 2.0 GPB, and 15.23% higher at 1.7 GPB. The RVGM achieved U/G of 0.4334 at 2.0 GPB is higher than the 0.4315 U/G reported in [3], even though the RVGM in that paper did not include 2D-ISI, and therefore had a much lower raw channel BER of 0.0274 compared to the 0.1060 raw BER in Table III. The improvement is mainly due to expanding the conditional PMF tables from, e.g., the $P(\mathrm{ix}_\alpha(\alpha_A)|\mathrm{ix}_2(y_A, y_U))$ used in [3] to $P(\mathrm{ix}_\alpha(\alpha_A + \alpha_E)|\mathrm{ix}_3(y_A, y_E, y_U))$, which internally captures more correlation among bits and read back values. Applying these enlarged tables should improve the LAIP detector's performance on Voronoi grain models without 2D-ISI as well. If enough memory were available to enlarge the tables to include more adjacent bits (e.g., all 8 bits surrounding the target bit), then the performance would likely improve further.

B. Simulation results for the GFP model

For the GFP model, we evaluate the system's performance (with the changes described in section III-D) on two waveform data sets, both with TP = 18 nm, BL = 11 nm, and GPB = 3.491. The input bits for these sets are of size 5×41207, and the readings are 3×41207, corresponding to the middle 3 tracks within the 5 input tracks; the outer two input bit tracks are considered as boundary tracks without readings. The first GFP data set corresponds to the special sets of 512 training patterns that were used to train the conditional PMFs for the GFP case. The second set is identical to one tested in [38], where simulation results in Fig. 16 of that paper show that an information density of 2.4 Tb/in^2 (corresponding to 0.2105 U/G) can be achieved with this data set by employing a 3-input/1-output 2D MMSE PR equalizer with a 1D BCJR and IRA decoder.

For both GFP model data sets, the left and right boundary bits (a total of six bits) are assumed to be known by the receiver; this overhead is minimal (.0061%) compared to the typical block size of three sectors at 4096 bytes per sector. We evaluate the GFP model data under three assumptions on the top and bottom boundary tracks: 1) both boundary tracks are known; 2) only the top boundary track is known; and 3) neither boundary track is known.

Boundary assumption 1 is mainly of theoretical interest, as no practical TDMR system will waste 25% of disk tracks in order to ensure known boundaries adjacent to every group of three tracks. However, boundary assumptions 2 and 3 are consistent with applying the proposed LAIP/BCJR detector to a disk where all tracks (except the outermost track in the case of boundary assumption 2) contain user data. Boundary assumption 2 might apply, e.g., in a streaming application where the disk is read sequentially from the first to last track, and there is a single known track just before the first data track (which would involve minimal overhead). Assumption 3 would apply to a random access scenario where there are no known tracks on the disk. Thus, under assumptions 2 and 3 for detection on an actual hard disk, the term "boundary" implies only the *location* of the two tracks adjacent to the three data tracks — it does not imply that the boundary tracks contain known data, or that they are wasted tracks on the disk.

The GFP model data is processed with a different puncturing method than the Voronoi model data. The GFP model method also selects N_x random punctured bits, and then reads only $N_w = N - N_x$ readback y values from the waveform file for each of the three rows, corresponding to N_w bits from each row's pre-determined (but randomly generated) channel input bit file. For each row, the coset encoder computes a coset parity word as the mod-2 sum of the N_p IRA encoder parity bits (computed by encoding the first K deinterleaved input bits) and the parity bits in the deinterleaved input bit file with zeros inserted at the punctured positions. After the N_w LAIP or BCJR output LLRs are deinterleaved, zero LLRs are inserted at the punctured positions, and the N_c LLRs are sent to the IRA coset decoder, along with the computed coset parity word. The punctured code rate is $r = K/N_w$. This coset decoding based puncturing scheme exactly simulates the puncturing process that would occur if the channel input bits were output from a channel encoder, instead of randomly generated and stored in a file.

The fifth through tenth rows of Table III are for the GFP model. The GFP test data set #1 used for the fifth through ninth rows has media and read/write parameters identical to those of the designed pattern training data set. When the system with 2D MMSE filter, 1D BCJR with IRA decoder in [38] is tested with a simple 1D PR target of $1 + 2D$ on the fifth row's GFP data, and an interleaver is used between

TABLE III
AREAL DENSITY PERFORMANCE OF THE LAIP/BCJR TURBO TDMR DETECTOR IN USER BITS PER GRAIN, FOR DECODED BER $\leq 10^{-5}$

Turbo Detectors	Grain Model	Grains per Coded Bit	User Bits per Grain	Code Rate	Channel BER	Decoded BER	Decoded FER	Throughput Multiplier
LAIP/BCJR	RVGM	2.0	0.4334	0.8668	0.1060	0 (1.0925e-5)	0	×3
LAIP/BCJR	RVGM	1.7	0.4525	0.7692	0.1427	0 (8.257e-6)	0	×3
BCJR	RVGM	2.0	0.3614	0.7228	0.1060	0 (3.2308e-5)	0	×3
BCJR	RVGM	1.7	0.3927	0.6676	0.1427	0 (2.6149e-6)	0	×3
LAIP/BCJR	GFP #1	3.491	0.1807	0.6308	0.1850	5.5083e-6	0	×3
LAIP/BCJR 1 boundary	GFP #1	3.491	0.1724	0.6017	0.1850	0 (2.5201e-7)	0	×3
LAIP/BCJR 1 boundary	GFP #1	3.491	0.1805	0.6300	0.1850	0 (3.7010e-7)	0	×2
LAIP/BCJR No boundaries	GFP #1	3.491	0.1681	0.5867	0.1850	0 (2.6325e-7)	0	×3
LAIP/BCJR No boundaries	GFP #1	3.491	0.1740	0.6075	0.1850	0 (3.9487e-7)	0	×2
LAIP/BCJR	GFP #2	3.491	0.1858	0.6485	0.1637	0 (8.4941e-6)	0	×3

the BCJR and IRA decoder, it achieves 0.1805 U/G. Thus, the LAIP/BCJR turbo-equalization system provides a U/G gain of 0.11% over the system in [38]; in addition, it provides a factor of three throughput gain, since the system in [38] processes one track at a time, whereas the proposed system processes three. We note that the 0.11% U/G gain occurs without pre-equalization to a 2D PR target before the LAIP detector; it is likely that such pre-equalization, by greatly reducing the ISI along tracks, would significantly lower the LAIP detector's BER and thereby allow increased IRA code rates, which would increase the achieved U/G of the LAIP/BCJR turbo detector.

In order to investigate the effect of having one or both of the boundary tracks unknown (under boundary assumptions 2 and 3), we propose to estimate the unknown boundary tracks on an actual hard disk by pre-processing them with a simple 1D BCJR detector with four trellis states that employs a three tap 1D ISI mask h, where h is estimated offline from training data by minimizing the MSE between the readings and the filtered data bits. Testing of this 1D BCJR on individual GFP data tracks shows that it gives a BER of 17.59%.

Under the scenario where the bottom boundary track is unknown (boundary assumption 2), we assume a decision feedback architecture where the top boundary track of the current three-track block has a BER of 10^{-5} (due to LAIP/BCJR processing of the previous block of three tracks), and the unknown bottom boundary has a BER of 17.59% due to pre-processing by the simple 4-state 1D BCJR algorithm. To simulate this decision feedback process, before the three-track LAIP/BCJR detector GFP simulation is run, random bit errors are inserted into the top and bottom GFP boundary tracks at BERs of 10^{-5} and 17.59% respectively.

In the case where both boundaries are unknown (boundary assumption 3), we assume that both boundaries are pre-processed by the 4-state 1D BCJR algorithm before the three-track LAIP/BCJR detector is run, and we simulate that pre-processing step by inserting bit errors into both top and bottom GFP boundary tracks at a BER of 17.59%. Because the LAIP/BCJR detector gives higher BERs on the tracks adjacent to the unknown boundary tracks, we assign different channel coding rates to the three tracks, and find the highest channel coding rates such that a BER of $\leq 10^{-5}$ is achieved on each track.

The LAIP/BCJR turbo detector results with one or two unknown boundaries are shown in rows 6-9 of Table III. In rows 6 and 8, the reported densities and code rates are an average over all three data tracks, and the throughput multiplier of ×3 is in comparison with a 1D detection scheme (e.g., 1D-PDNP). In rows 7 and 9, the densities and code rates are averaged only over the first two tracks, under a scenario where the LAIP/BCJR detector releases only two tracks at a time, and the throughput multiplier is ×2. When only the top boundary track is known, the code rates for tracks 1, 2 and 3 are $\{0.635, 0.625, 0.545\}$, and the densities are $\{0.1819, 0.1790, 0.1561\}$; it is seen that the rate for track three (adjacent to the unknown boundary) must decrease significantly, as the unknown boundary causes a higher detector BER. (The detector BERs for all three tracks for one and two unknown boundaries are shown in Table IV in subsection IV-C below.) When only one boundary is known, there is an average density reduction of 4.6% compared to the case when both boundaries are known; however the density reduction reduces to 0.1% if only two tracks are released, thus demonstrating a trade-off between density and throughput. When neither boundary is known, the code rates and densities for tracks 1, 2, and 3 are $\{0.595, 0.620, 0.545\}$ and $\{0.1704, 0.1776, 0.1561\}$ respectively; there is an average density reduction of 7.0% compared to the case where both boundaries are known, which reduces to 3.7% when only two tracks are released.

The GFP test data set #2 in the tenth row also has TP = 18 nm and BL = 11 nm, but the GFP simulation reader and writer parameters for this test set were different than those for the fifth row, resulting in a lower BER; the fifth row data's raw BER of 0.1850 is about 13% higher than that of the tenth row. Also, the tenth row's ISI mask (calculated using the LS method) shows significantly less 2D-ISI than that seen in the fifth row's mask shown in the third column of Table II. This means that the LAIP detector's trained PMF tables are based on very different data than the tenth row GFP data.

TABLE IV
DETECTOR BER COMPARISON BETWEEN LAIP/BCJR AND THE 2D-PDNP DETECTOR FROM [29], ON DATA TRACKS 1, 2, AND 3

Detector	N_c	N_p	I	J	**Trellis States**	**BER Track 1**	**BER Track 2**	**BER Track 3**
LAIP/BCJR	NA	NA	NA	NA	64	8.67%	8.85%	8.66%
LAIP/BCJR 1 boundary	NA	NA	NA	NA	64	8.69%	9.23%	11.63%
LAIP/BCJR No boundaries	NA	NA	NA	NA	64	9.81%	9.24%	11.69%
2D-PDNP	11	1	1	1	64	10.60%	12.57%	NA
2D-PDNP	23	1	1	1	64	10.60%	12.57%	NA
2D-PDNP	11	2	1	1	256	10.63%	12.48%	NA

Our achieved density of 0.1858 user bits per grain is about 14% lower than the .2105 U/G achieved by the system in [38] on the same TP 18 data set; the density loss is due to model mismatch between the PMF tables and the actual test data. While the loss is significant, it is relatively reasonable given the degree of model mismatch, and suggests that the LAIP detector approach has a certain degree of robustness to model mismatch.

C. Comparison with 2D-PDNP

In order to compare our proposed LAIP/BCJR system to a state-of-the-art 2D-PDNP detector, we implemented a BCJR algorithm that employs the 2D-PDNP of [29] and tested it on GFP data set #1 mentioned in the description of Table III above. We now give a brief description of the detector in [29].

The 2D-PDNP VA in [29] processes two tracks of data simultaneously, and performs 2D autoregressive pattern-dependent noise prediction as [29]

$$\mathbf{n}_k = \sum_{i=0}^{N_p} \mathbf{P}_i(\mathbf{A}_k)\mathbf{n}_{k-i} + \mathbf{\Lambda}(\mathbf{A}_k)\mathbf{w}_k. \tag{13}$$

In (13), $\mathbf{n}_k$ is the 2×1 vector of predicted noise samples from tracks 1 and 2 at downtrack position k, N_p is the predictor memory, the $\mathbf{P}_i$ are the 2×2 matrix coefficients of the 2D autoregressive model, and the $\mathbf{A}_k$ are the 2D pattern matrices of size $2 \times (I + J + 1)$, which consider all $4^{(I+J+1)}$ possible bit patterns on the two tracks that span the downtrack samples $(k-J, \ldots, k, \ldots, k+I)$. The 2×2 diagonal matrix $\mathbf{\Lambda}(\mathbf{A}_k)$ has pattern dependent standard deviations $\sigma_1(\mathbf{A}_k)$ and $\sigma_2(\mathbf{A}_k)$ on its diagonal, and $\mathbf{w}_k$ is a 2×1 vector of 0 mean unit variance i.i.d. Gaussian random variables. The number of trellis states in the VA (or BCJR) with 2D-PDNP is $4^{(N_p+I+J)}$.

A method for joint design of an MMSE equalizer and 2D PR target is also given in [29], and used to design a fractionally spaced equalizer of length $N_c = 22$ for use on a micromagnetic data set from Ehime University with two samples per bit [46]; the MMSE equalizer pre-processes the micromagnetic data samples before they are sent to the 2D-PDNP VA. Since our LAIP/BCJR system uses only 1 sample per bit from the GFP data sets, we design an MMSE equalizer of length $N_c = 11$ according to the method in [29] for use with our BCJR/2D-PDNP implementation. Following [29], we estimate the 2D target $s(\mathbf{A})$ associated with the bit pattern $\mathbf{A}$ by averaging the MMSE filter output readings $\mathbf{y}_k$ associated with pattern $\mathbf{A}$.

Table IV compares the BER of the LAIP/BCJR detector proposed here with that of our implementation of a BCJR that employs the 2D-PDNP proposed in [29], when both algorithms are run on GFP data set #1 mentioned in the description of Table III above. The BERs reported in Table IV are detector-only BERs without channel decoding. The 2D-PDNP algorithm's prediction coefficients, sigmas, and estimated targets are trained with forty 41K blocks of GFP readings and associated known input bits on each of the two tracks; the 2D-PDNP training readings are first pre-filtered by the 2D-MMSE of down-track length N_c designed according to the method in [29]. The LAIP PMF tables are trained according to the procedures described in subsections III.A, III.C and III.D. Table IV gives LAIP/BCJR BER results for all three tracks, and 2D-PDNP BERs for the two tracks that it processes.

Rows 4 through 6 of Table IV summarize the performance of the 2D-PDNP BCJR for several different parameter settings: in row 4, $N_p = I = J = 1$, as in the simulation results presented in [29]. The row 5 detector is identical to that in row 4, except that an increased length $N_c = 23$ MMSE equalizer is designed and deployed before the 2D-PDNP BCJR. The row 6 detector is identical to that in row 4, except that the predictor memory N_p is increased to 2. All parameter settings of the 2D-PDNP give about 10.6% BER on track 1, but setting $N_p = 2$ gives the lowest BER of 12.48% on track 2.

Table IV row 1 shows that, with two known boundaries, the LAIP/BCJR detector's BER is relatively constant: between 8.66% and 8.85% on all tracks. When the bottom boundary adjacent to track 3 is unknown, track 3's BER increases from 8.66% to 11.63% (34.3% increase), and the track 2 and track 1 BERs increase by 4.3% and 0.2%, respectively. The track 2 and 1 BER increases occur because the LAIP jointly estimates all three tracks, so that errors in track 3 affect the other two tracks. When both boundaries are unknown, tracks 1–3 suffer BER increases of 13.2%, 4.4%, and 35.0% respectively relative to the known boundaries case. The track 1 BER increase is less than the track 3 increase because, due to the shingled writing process that produces the GFP data, track 1 overwrites the top boundary track 0, whereas track 3 is overwritten by the bottom boundary track 4. Because boundary track 4 is not overwritten by another track, it has a larger effective track pitch and a higher SNR than boundary track 0, leading to a higher ITI between track 4 and track 3 than that between track 0 and track 1; thus, the 17.59% BERs in tracks 4 and 0 lead to a higher BER in track 3 than in track 1.

The 2D-PDNP does not assume anything about its adjacent boundary tracks. Hence, a fair comparison with 2D-PDNP requires considering the LAIP/BCJR with no boundary information. The best 2D-PDNP performance is achieved by the row 6 case, which has an average BER of 11.555% over two tracks, whereas the unknown-boundary LAIP/BCJR has an average BER of 10.247% over three tracks, which is 11.3% lower than the 2D-PDNP. The lower BER of the LAIP/BCJR is probably due at least in part to the LAIP/BCJR processing three tracks at a time rather than two, enabling it to take into account ITI from both adjacent tracks to any given track, and also giving it 1.5× the throughput of the 2D-PDNP method. The LAIP detector's ability to form a more general probabilistic model of grain-bit effects than the 2D-PDNP's autoregressive model may also partially explain its lower BER.

D. Computational Complexity

We end this section by summarizing the computational and storage resources required by the proposed LAIP/BCJR detection system, and drawing comparisons to some other recently published TDMR detectors. When the versions of the LAIP and 2D-BCJR that simultaneously process three tracks of GFP data are run on a 2.4 GHz Intel Xeon CPU with 25 MB cache, the LAIP (coded in MATLAB) requires 0.75 ms per coded bit, and the 64-state 2D-BCJR (coded in C++) requires 0.26 ms per bit. By comparison, the system in [38] processes one track at a time and employs a four-state BCJR detector. Thus, the proposed system has about $(16 + 3 \times 16) = 64\times$ the computational complexity of the system in [38]. If both boundaries are unknown, the boundary pre-processing by the 4-state 1D BCJR would cause the total complexity of the LAIP/BCJR system to be about 66× the computational complexity of the system in [38]. The complexity increase to achieve 3× throughput (at approximately the same U/G as [38]) is significant. We note, however, that both the one-track BCJR detector in [38] and the two-track 2D-PDNP detector in [29] employ 2D MMSE linear PR equalizers before their trellis detectors, whereas no MMSE PR equalizer is used in the present paper. We believe that significant additional density gains are achievable if an appropriately designed 3-input/3-output 2D-MMSE PR equalizer is used as a pre-processor before the LAIP/BCJR detector proposed in this paper; design of such a system will be the topic of a future publication.

The number of states shown for the LAIP/BCJR detector in Table IV is the number of states in the BCJR detector. Because (per run-time measurements) the LAIP detector has about 3× the computational complexity of the 64-state BCJR, the equivalent number of trellis states of the LAIP/BCJR system is about 256, which is equal to the 2D-PDNP with $N_p = 2$ and $I = J = 1$ considered in Table IV row 6. Thus, the LAIP/BCJR detector has roughly equal complexity to the 2D-PDNP detector in [29], when $N_p = 2$ and $I = J = 1$.

The final complexity comparison is with the GBP based TDMR detector in [19]. Run-time measurements reported in [3] show that the GBP based detector in [19] has about 10000× the computational complexity of the LAIP detector considered in [3], which is somewhat less complex than the LAIP detector considered in this paper, as it uses a simpler structure for its joint PMF tables, and does not employ self loops. Considering the four self loops of the current paper's LAIP detector, and conservatively estimating about a factor of 2× greater complexity for one pass through this paper's LAIP detector due to the larger joint PMF tables, the LAIP/BCJR detector in the present paper has about 11× the complexity of that in [3]. It follows that the GBP detector of [19] has about 900× the complexity of the combined LAIP/BCJR detector considered in this paper.

The storage requirements of the LAIP detector are now considered. The LAIP uses a number of pre-trained PMF tables. The tables are trained offline once and stored. The PMF tables use a 4-byte floating point representation, and require about 0.78 GB of storage, which is quite modest compared to modern HDDs with typically $\geq$ 1 TB of storage.

V. Conclusion

The LAIP detector approach trains conditional PMFs of the influences of adjacent bits on target bits, and compares the read value of the target bit to the estimated total influence of its surrounding bits. The influences of adjacent bits can be viewed as the coefficients in a traditional 2D-ISI convolution mask. Because the influences are conditioned on read values, the LAIP detector can be viewed as estimating the coded bits based on a spatially varying 2D-ISI mask. This paper has demonstrated that the LAIP detector can be trained to handle both grain/bit interactions as well as 2D-ISI effects, and that the proposed combination of the LAIP and BCJR turbo detectors can provide significant performance gains compared to the BCJR turbo detector alone, and can also compare favorably to previously proposed detection schemes on the considered Voronoi and GFP grain models.

We also note that the the LAIP detector is computationally efficient; since the component PMF tables are pre-computed, the LAIP detector's main computational task consists of vector convolution, which can be optimized for speed in several ways (e.g., by frequency domain methods). In addition, the LAIP's conditional PMF tables give the detector a high degree of robustness to different media models: they can be trained on geometrically well defined grain models like the Voronoi grain model, or even on more complex models such as the GFP where details of the underlying grain geometry may not be available. The storage requirements of the LAIP PMF tables are somewhat high but not unreasonable; in fact, the LAIP essentially trades off PMF table storage requirements for improved estimation performance and computational efficiency. Moreover, the LAIP detector's performance can be improved as memory storage resources increase in the future, thereby allowing training and storage of joint conditional PMFs of groups of 5, 6 or more influences, because this could give a more accurate statistical model of the spatial dependencies between the bits and the readings in a local area around the target bit. However, expanding the size of the joint influence regions would require larger training sets to ensure reliable estimates of the LAIs.

Acknowledgment

This work was supported by NSF grants CCF-1218885 and CCF-1817083, and also by the Advanced Storage Technology Consortium (ASTC).

References

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

[2] S. S. Garani, L. Dolecek, J. Barry, F. Sala, and B. Vasic, "Signal processing and coding techniques for 2-D magnetic recording: An overview," *Proceedings of the IEEE*, vol. 106, no. 2, pp. 286–318, Feb 2018.

[3] X. Sun, K. Sivakumar, B. Belzer, and R. Wood, "High density turbo TDMR detection with local area influence probabilistic model," *IEEE Trans. Magnetics*, vol. 53, no. 2, pp. 1–8, Feb. 2017, article no. 9400208.

[4] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. 2nd Int. Symp. on Turbo Codes and Rel. Top.*, Brest, France, Sept. 2000, pp. 1–8.

[5] Y. Chen, B. Belzer, and K. Sivakumar, "Iterative row-column soft-decision feedback algorithm using joint extrinsic information for two-dimensional intersymbol interference," in *Proc. 44th Conf. Info. Sci. and Syst. (CISS 2010)*, Princeton, NJ, Mar. 2010.

[6] K. S. Chan, R. Radhakrishnan, K. Eason, M. R. Elidrissi, J. J. Miles, B. Vasic, and A. R. Krishnan, "Channel models and detectors for two-dimensional magnetic recording," *IEEE Trans. Magnetics*, vol. 46, no. 3, pp. 804–811, March 2010.

[7] K. S. Chan, E. M. Rachid, K. Eason, R. Radhakrishnan, and K. K. Teo, "Comparison of one- and two-dimensional detectors on simulated and spin-stand readback waveforms," *Journal of Magnetism and Magnetic Materials*, vol. 324, no. 3, pp. 336 – 343, 2012, selected papers of the 9th Perpendicular Magnetic Recording Conference(PMRC 2010). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304885310009170

[8] K. Keng Teo, M. R. Elidrissi, K. S. Chan, and Y. Kanai, "Analysis and design of shingled magnetic recording systems," *Journal of Applied Physics*, vol. 111, no. 07B716, 2012. [Online]. Available: http://aip.scitation.org/doi/pdf/10.1063/1.3679383

[9] K. S. Chan, K. K. Teo, M. Y. Lin, H. C. Kee, M. R. Elidrissi, Q. Li, B. Ko, Q. Choo, and M. N. Maung, "Comparison of signals from micromagnetic simulations, GFP model, and an HDD readback," *IEEE Trans. Magnetics*, vol. 51, no. 11, pp. 1–4, Nov 2015.

[10] M. R. Elidrissi, K. S. Chan, K. K. Teo, K. Eason, E. Hwang, B. V. K. V. Kumar, and Z. Qin, "Modeling of 2-D magnetic recording and a comparison of data detection schemes," *IEEE Trans. Magnetics*, vol. 47, no. 10, pp. 3685–3690, Oct 2011.

[11] L. Pan, W. E. Ryan, R. Wood, and B. Vasic, "Coding and detection for rectangular-grain TDMR models," *IEEE Trans. Magnetics*, vol. 47, no. 6, pp. 1705–1711, June 2011.

[12] M. Carosino, Y. Chen, B. J. Belzer, K. Sivakumar, J. Murray, and P. Wettin, "Iterative detection and decoding for the four-rectangular-grain TDMR model," in *Proc. 51st Allerton Conf. on Comm., Computing, and Control*, Champaign, IL, Oct. 2013, pp. 653–659.

[13] M. Carosino, J. Yu, Y. Chen, M. Mehrnoush, B. J. Belzer, K. Sivakumar, R. Wood, J. Murray, and P. Wettin, "Iterative detection and decoding for TDMR with 2D intersymbol interference using the four-rectangular-grain model," *IEEE Trans. Magnetics*, vol. 51, no. 7, p. 3000812, July 2015.

[14] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, March 1974.

[15] A. Kavcic, X. Huang, B. Vasic, W. Ryan, and M. F. Erden, "Channel modeling and capacity bounds for two-dimensional magnetic recording," *IEEE Trans. Magnetics*, vol. 46, no. 3, pp. 812–818, Mar. 2010.

[16] R. M. Todd, E. Jiang, R. Galbraith, J. R. Cruz, and R. W. Wood, "Two-dimensional Voronoi-based model and detection for shingled magnetic recording," *IEEE Trans. Magnetics*, vol. 48, no. 11, pp. 4594–4597, Nov. 2012.

[17] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inform. Theory*, vol. 51, no. 7, pp. 2282–2312, Jul. 2005.

[18] M. Khatami and B. Vasic, "Detection for two-dimensional magnetic recording systems," *Journal of Communications*, vol. 8, no. 4, pp. 233–239, Apr. 2013.

[19] X. Sun, B. Belzer, and K. Sivakumar, "Dynamic grain state estimation for high-density TDMR: Progress and future directions," *IEEE Trans. Magnetics*, vol. 52, no. 2, pp. 1–7, Feb. 2016, article 9400107.

[20] M. Khatami and B. Vasic, "Constrained coding and detection for TDMR using generalized belief propagation," in *Proc. IEEE Int. Conf. on Commun. (ICC 2014)*, 2014, pp. 3889–3895.

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

[22] J. Moon and J. Park, "Pattern-dependent noise prediction in signal-dependent noise," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 4, pp. 730–743, Apr 2001.

[23] B. Vasic and E. M. Kurtas, *Coding and Signal Processing for Magnetic Recording Systems*. Boca Raton: CRC Press, 2005.

[24] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.

[25] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.

[26] J. Yao, E. Hwang, B. V. K. V. Kumar, and G. Mathew, "Two-track joint detection for two-dimensional magnetic recording (TDMR)," in *2015 IEEE Int. Conf. on Commun. (ICC)*, June 2015, pp. 418–424.

[27] Y. Wang and B. V. K. V. Kumar, "Multi-track joint detection for shingled magnetic recording on bit patterned media with 2-d sectors," *IEEE Trans. Magnetics*, vol. 52, no. 7, pp. 1–7, July 2016.

[28] ——, "Micromagnetics-based analysis of multi-track detection with simplified 2-d write precompensation on shingled magnetic recording," *IEEE Trans. Magnetics*, vol. 52, no. 9, pp. 1–11, Sept 2016.

[29] S. Shi and J. R. Barry, "Multitrack detection with 2D pattern-dependent noise prediction," in *2018 IEEE Int. Conf. on Commun. (ICC)*, May 2018, pp. 1–6.

[30] C. K. Matcha and S. G. Srinivasa, "Generalized partial response equalization and data-dependent noise predictive signal detection over media models for TDMR," *IEEE Trans. Magnetics*, vol. 51, pp. 1–15, Oct. 2015, article no. 3101215.

[31] Y. Wu, J. A. O'Sullivan, N. Singla, and R. S. Indeck, "Iterative detection and decoding for separable two-dimensional intersymbol interference," *IEEE Trans. Magnetics*, vol. 39, no. 4, pp. 2115–2120, July 2003.

[32] M. Marrow and J. K. Wolf, "Iterative detection of 2-dimensional ISI channels," in *Proc. Info. Theory Workshop*, Paris, France, Mar./Apr. 2003, pp. 131–134.

[33] T. Cheng, B. J. Belzer, and K. Sivakumar, "Row-column soft-decision feedback algorithm for two-dimensional intersymbol interference," *IEEE Signal Processing Letters*, vol. 14, pp. 433–436, July 2007.

[34] 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. Inform. Theory*, vol. 54, no. 4, pp. 1500–1513, April 2008.

[35] Y. Chen and S. G. Srinivasa, "Joint self-iterating equalization and detection for two-dimensional intersymbol-interference channels," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3219–3230, August 2013.

[36] M. Tuchler, R. Koetter, and A. C. Singer, "Turbo equalization: principles and new results," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754–767, May 2002.

[37] C. K. Matcha and S. G. Srinivasa, "Joint timing recovery and signal detection for two-dimensional magnetic recording," *IEEE Trans. Magnetics*, vol. 53, no. 2, pp. 1–11, Feb 2017.

[38] K. S. Chan, A. James, S. Shafi'ee, S. Rahardja, J. Shen, K. Sivakumar, and B. J. Belzer, "User areal density optimization for conventional and 2-D detectors/decoders," *IEEE Trans. Magnetics*, vol. 54, no. 2, pp. 1–12, Feb. 2018, article no. 3100412.

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

[40] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[41] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform. Theory*, vol. 6, pp. 7–12, Mar. 1960.

[42] E. Hwang, R. Negi, and B. V. K. V. Kumar, “Signal processing for near 10 Tbit/in^2 density in two-dimensional magnetic recording (TDMR),” *IEEE Trans. Magnetics*, vol. 46, no. 6, pp. 1813–1816, June 2010.

[43] M. Mehrnoush, B. J. Belzer, K. Sivakumar, and R. Wood, “Turbo equalization for two dimensional magnetic recording using Voronoi model averaged statistics,” *IEEE Jour. on Sel. Areas in Commun.*, vol. 34, no. 9, pp. 2439–2449, Sept. 2016.

[44] D. Divsalar and F. Pollara, “Turbo codes for PCS applications,” in *IEEE Int. Conf. Commun. (ICC’95)*, vol. 1, Jun 1995, pp. 54–59.

[45] F. Scholz, “Confidence bounds and intervals for parameters relating to the binomial, negative binomial, Poisson and hypergeometric distributions with applications to rare events,” *2008*, Online Report: http://www.stat.washington.edu /people /fritz /DATAFILES /Confidence-Bounds.pdf.

[46] J. R. Barry, B. Vasic, 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*, vol. 52, no. 2, pp. 1–7, Feb 2016.