Convolutional Neural Network-Based Media Noise Prediction and Equalization for TDMR Turbo-Detection With Write/Read TMR

Amirhossein Sayyafan[1], Ahmed Aboutaleb[1], Benjamin J. Belzer[1], Krishnamoorthy Sivakumar[1], Simon Greaves[2], Kheong Sann Chan[3], and Ashish James[4]

[1]School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164 USA
[2]Research Institute of Electrical Communication (RIEC), Tohoku University, Sendai 980-8577, Japan
[3]Department of Electrical and Electronic Engineering, University of Nottingham Malaysia, Semenyih 43500, Malaysia
[4]Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore 138632

This article considers a turbo-detection system that includes a convolutional neural network (CNN)-based equalizer, a Bahl-Cocke-Jelinek-Raviv (BCJR) trellis detector, a CNN-based media noise predictor (MNP), and a low-density parity-check (LDPC) channel decoder for two-dimensional magnetic recording (TDMR) in the presence of track misregistration (TMR). The input readings are passed to a 2-D partial response (PR) equalizer, which is either linear or CNN-based. The equalized waveforms are inputs to a 2-D BCJR detector, which generates log-likelihood-ratio (LLR) outputs. The CNN MNP is provided with BCJR LLRs to estimate signal-dependent media noise samples and feed them back to the BCJR. A second pass through the BCJR produces LLRs, which are decoded by an LDPC decoder; achieved areal density (AD) is computed from the LDPC code rate. Spatially varying read- and write-TMR models are developed. We investigate the performance of the proposed system on simulated TDMR readback waveforms generated by grain-switching probabilistic (GSP) simulations. We have two types of GSP datasets. Dataset #1 includes two 10 nm bit length (BL) datasets with 18 and 24 nm track pitch (TP). Dataset #2 has 11 nm BL and 15 nm TP. The comparison baseline is a 1-D BCJR detector with pattern-dependent noise prediction (PDNP) and soft intertrack interference (ITI) subtraction, referred to as 1-D PDNP with LLR exchange. The write-TMR and read-TMR are modeled as cross-track-independent downtrack-correlated random processes. In the presence of joint write- and read-TMR, the proposed turbo-detection system achieves 8.34% and 0.70% AD gain over 1-D PDNP with LLR exchange for TP 18 and 24 nm dataset #1, respectively, and is more robust to TMR compared to the baseline.

Index Terms— **Bahl-Cocke-Jelinek-Raviv (BCJR) detector, convolutional neural network (CNN), CNN equalizer, CNN media noise predictor (MNP), deep neural network (DNN), low-density parity-check (LDPC) decoder, turbo-detection system, two-dimensional magnetic recording (TDMR), write/read track misregistration (TMR).**

I. Introduction

THE detection of two-dimensional magnetic recording (TDMR) subject to write and read track misregistration (TMR) is considered, assuming a streaming application such as video storage. The write-TMR, modeled as slowly varying cross-track-independent downtrack-correlated random processes, induces spatially varying intertrack interference (ITI) in the received signal. The read-TMR is modeled similarly, except that all three readers share the same downtrack-correlated random drift process. In this present article, since TDMR signals are viewed as 2-D images, a convolutional neural network (CNN)-based media noise predictor (MNP) and equalizer are utilized in a turbo-detection system to handle the write- and read-TMR and compare with the pattern-dependent noise prediction (PDNP), which is standard practice for hard disk drive (HDD) magnetic recording [1], [2].

To maximize the areal density (AD) of a TDMR system with single-track detectors, a systematic method for determining the combination of multi-reader geometry, track pitch (TP), and bit-aspect ratio is presented in [3]. In this work, the two-reader geometry that maximizes AD with zero skew and zero misregistration was found to use different-sized readers with significant overlap in the cross-track direction for the considered head and medium.

To enhance the AD for HDD in industry, the transitioning into array-reader-based magnetic recording (ARMR) technology [4] is used to provide an enhanced signal-to-noise ratio for data detection by exploiting the diversity in signal, interference, and noise provided by the multiple read elements. In the ARMR technology, a matched 2-D equalizer matched to the reader location on the track is employed, which may not work well due to TMR arising from the servo control system. To reduce the resulting TMR, Zheng et al. [5] presented the TMR-sensitive equalization, which is an electronic servoing scheme that estimates the location of the dual-reader on a per-fragment basis. The estimated location is then used to transform the reference equalizer to a new equalizer, which is matched to the estimated one. The simulations in [5] show that the read performance has significant improvements in the presence of large read-/write-TMR on density of $\sim$1 Tb/in^2.

Two previous papers [6], [7] have considered use of soft adjacent track side information to aid the central track's detector. The channel models used in both these papers are significantly simpler and less realistic than the grain-switching-probabilistic (GSP) shingled magnetic recording (SMR) model

Manuscript received 3 August 2022; revised 16 September 2022; accepted 14 October 2022. Date of publication 25 October 2022; date of current version 24 February 2023. Corresponding author: B. J. Belzer (e-mail: belzer@wsu.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMAG.2022.3216640.

Digital Object Identifier 10.1109/TMAG.2022.3216640

employed in this article, and the technique proposed in this present article has relatively less computational complexity. In [6], a simple bit-patterned magnetic recording (BPMR) separable linear channel model is employed; the model includes electronic noise but no media noise. Chang and Cruz [6] employed a 2-D Bahl-Cocke-Jelinek-Raviv (BCJR) trellis with a 2-D partial response (PR) target equalizer for estimation of the center track bits, with log-likelihood ratios (LLRs) from the adjacent tracks passed to the 2-D BCJR to provide a priori information; simple 1-D BCJRs can be used to develop the adjacent track LLRs. The 2-D BCJR of [6] requires significantly more trellis states and more branches per state than the 1-D BCJR that would be used in this present article if no media noise were present; this present article employs 1-D BCJR with PDNP in order to predict and correct for media noise present in the GSP model. However, Chang and Cruz [6] passed only one LLR per each of two adjacent track bits to estimate a central track bit since they are used to develop a priori probabilities, whereas this present article passes three LLRs per adjacent track in order to develop a soft estimate of the adjacent tracks' ITI. Because channel coding is not considered in [6], AD gains over a comparison baseline are not computed in that paper.

The paper [7] employs a separable linear channel model with jitter noise (a form of media noise) and electronic noise, to simulate an SMR channel. Similar to this present article, three 1-D BCJR detectors (but without PDNP) are employed, and multiple LLRs per central track bit are passed from the outer tracks in order to develop a soft estimate of the ITI that can be subtracted from the central track's intersymbol interference (ISI) equalized signal. As in this present article, Sadeghian and Barry [7] employed per-track low-density parity-check (LDPC) coding. A key difference is that work [7] passes LDPC decoded LLRs from the outer tracks, rather than detector LLRs; hence, three decoding passes must be performed to produce one block of decoded central track bits in [7], whereas this present article requires only one LDPC decoding for the central track, thereby realizing significant complexity savings. While channel decoding in [7] improves the reliability of the outer track bit LLRs relative to the detector LLRs employed in this present article, this present article's use of 1-D PDNP to correct for outer track media noise improves the reliability of its detector LLRs relative to [7].

For a TDMR system with two displaced readers, a machine learning (ML) data detection channel, including a CNN for data recovery, was utilized in [8]. The data recovery over a wide range of ITI was studied to emulate the actual head skew angle change over the entire disk platter. In this work, the sampled input signals from both readers were passed to the system; however, the CNN-based ML channel only detected the main track data during the training and almost eliminated the ITI-caused degradation of bit error rate (BER) completely.

In [9], for real data from a commercial HDD with TDMR dual-reader technology, a CNN-based data detection channel is employed. In this work, it is shown that the proposed CNN-based detection had a comparable or slightly better performance than the state-of-the-art HDD channel. In addition, the CNN-ML channel achieves 6% recording density gain when the second reader is available compared to the detection with a single reader.

A neural network (NN)-based non-linear equalization is designed for TDMR in [10]. This adaptive equalizer utilizes the cross-entropy criterion between the true probability of the bits and estimations of the detector. By using the cross-entropy criterion, the equalizer tries to maximize the likelihood, which leads to a lower BER for the detector compared to the mean square error (MSE) criterion.

A reduced complexity NN-based equalization for TDMR system is investigated in [11]. It is shown that the multilayer perceptron (MLP) non-linear equalizer outperforms the linear equalizer using the cross-entropy criterion. The authors propose some simpler architectures for their MLP-based equalizer to reduce the complexity.

In [12], a deep neural network (DNN)-based a posteriori probability (APP) detector is proposed for the TDMR system. The reading samples are passed through a 2-D PR linear equalizer; then, the equalized waveforms are fed to the DNN APP detector. The iteration between the detector and an LDPC decoder minimizes the BER. In [13], the designed CNN-based detector performs the tasks of equalization and detection. In this case, the reading samples are input to the CNN-based detector; then, by passing the detector outputs to the LDPC decoder, the AD is measured.

We have explored using a CNN detector as the sole detector for the TDMR signal, similar to [9], [12], and [13]. These approaches, while effective, require relatively more computational complexity than the system proposed here, which limits the CNN to media noise prediction only and uses a relatively low-complexity 2-D BCJR detector for ISI/ITI equalization. The approaches in [12] and [13] did not consider read- and write-TMR.

Recently, there has been increasing attention to developing CNN-based systems for multilayer magnetic recording (MLMR) to increase the areal information density of HDDs [14], [15], [16].

This present article considers the simultaneous detection of three tracks via triple-reader technology, enabling more effective ITI equalization and higher throughput than dual-reader configurations. Input data for this present article is provided by GSP simulations of HDD readback waveforms; such simulations have been shown to give highly accurate reproductions of actual HDD waveforms [17], [18], [19]. In this work, the turbo-detection system includes a CNN-based equalizer, a BCJR trellis detector [20], a CNN-based MNP, and an LDPC channel decoder for TDMR in the presence of write- and read-TMR. The soft information is exchanged among BCJR detector, CNN MNP, and LDPC decoder iteratively to maximize the AD subject to a BER constraint.

The main contributions of this work are given as follows.

1) This article is one of the first to investigate the performance of ML-based turbo-equalization in the presence of both write- and read-TMR, especially given the realistic GSP media models that we use.
2) A modified version of 1-D PDNP is designed to handle the ITI from adjacent tracks in the TDMR system, which

improves the performance compared to standard 1-D PDNP. This modified 1-D PDNP with LLR exchange is the comparison baseline in this work.

3) A spatially varying read-TMR model is developed and used to train the CNN-based noise predictors and equalizers. The proposed system has 8.52% AD gain over the baseline in the presence of the read-TMR at TP 18 nm.
4) A write-TMR model is developed and used to train the CNNs in the TDMR turbo-detection system. The proposed system achieves 8.24% AD gain over the baseline with only write-TMR.
5) A joint write- and read-TMR model is developed and used to train the TDMR turbo-detector. The proposed system is more robust compared to the baseline and has 8.34% AD gain over it.

In the rest of this article, the system model and the characteristics of the datasets are explained in Section II. The process of generating read- and write-TMR is described in Section III. In Section IV, we explain the CNN-based equalizer, and in Section V, we propose the BCJR-LDPC-CNN turbo-detection system. In Section VI, we present the 1-D PDNP with soft ITI subtraction, called 1-D PDNP with LLR exchange. In Section VII, we present and discuss the simulation results. Concluding remarks are presented in Section VIII.

II. Recording Channel Model

The BCJR-LDPC-CNN turbo-detector assumes a channel model for the kth equalizer output $\mathbf{y}(k)$

$$\mathbf{y}(k) = (\mathbf{h} * \mathbf{u})(k) + n_m(k) + n_e(k) \tag{1}$$

where $\mathbf{h}$ is the PR target, $\mathbf{u}$ are the coded bits on the track, $*$ indicates 1-D/2-D convolution, $n_m(k)$ is media noise, and $n_e(k)$ is reader electronics additive white Gaussian noise (AWGN). Unlike PDNP, the media noise term $n_m(k)$ is not modeled as an autoregressive (AR) process; instead, a more general model for $n_m(k)$ is learned by the CNN MNP.

Realistic GSP models were used to train and evaluate our system. Two different GSP models were used, which will be referred to hereafter as the dataset #1 model and the dataset #2 model.

The dataset #1 model was trained on SMR. Datasets with nominal TPs of 18 nm and 24 nm were generated. In each case, the bit length (BL) was 10 nm, each track consisted of 32 768 bits (4 kbytes), and the grain density of the recording media was 13.2 teragrains per square inch (Tg/in^2). One hundred blocks (sectors) with no write-TMR were generated as "dataset #1a" to allow the effect of read-TMR only to be calculated. Fig. 1 shows a cross-track view of the dataset #1 tracks. Six tracks were written in order from left to right. The rightmost track $\mathbf{a}_{b,R_2}$ represents the fat track present in most TDMR systems. The centers of the first and last tracks were at fixed cross-track positions, while the centers of the remaining four tracks were varied systematically by ± 3 nm from their nominal position in 1 nm increments. Thus, a total of 7^4 or 2401 datasets were generated as "dataset #1b" to study write-TMR.

Readback signals $\mathbf{r}_0, \ldots, \mathbf{r}_4$ were calculated from five readers, as shown in Fig. 1. The readback signals were obtained

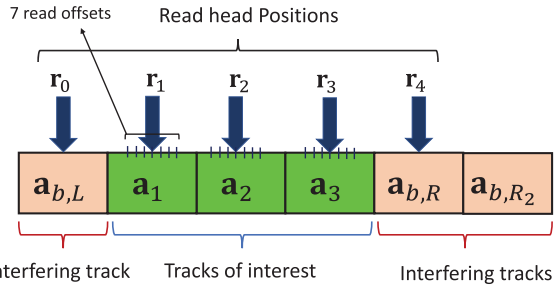

Fig. 1. Cross-track view of the dataset #1 model geometry.

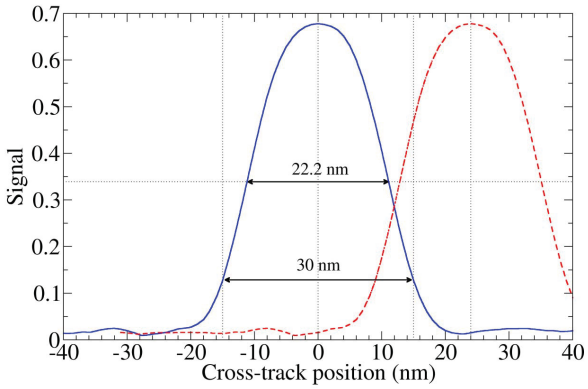

Fig. 2. Cross-track signal obtained from the magnetization of a single track written using the dataset #1 model with TP 24 nm.

from the three central tracks $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$ and the boundary signals were read from $\mathbf{a}_{b,L}$ and $\mathbf{a}_{b,R}$. The reader positions were varied by ± 3 nm from the track centers for a total of seven offsets for each track of interest with a resolution of 1 nm.

The written track width for the dataset #1b model can be determined from the cross-track profile of a single written track. Fig. 2 shows an example, taken from a direct analysis of the medium magnetization for the TP 24 nm dataset #1b. The full-width at half-maximum (FWHM) track width was 22.2 nm, but the magnetization of the medium was changed over a wider area than this. As a result, a TP of 24 nm will lead to the tracks overlapping to some extent, as exemplified by the red curve in Fig. 2, which has been shifted 24 nm from the blue curve. The written track width is determined by the write head and spin-torque oscillator (STO) geometry. If either of these were changed, the GSP model would need to be retrained.

For dataset #1b with TP 18 nm, the read-head positions for tracks $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$ have minimum BER (using a 1-D BCJR detector) at 2 nm to the left of the write head location. This indicates that the tracks were shingled to some extent when they were written.

The "dataset #2" model was used to write sets of five, partially overlapping (shingled) tracks, each with 11 nm BL, 15 nm TP, and 41 206 bits. In this case, the dataset #2 model was based on micromagnetic simulations of conventional recording [19] and the grain density was 11.4 Tg/in^2. For dataset #2, tracks are written at nominal TP only; hence,

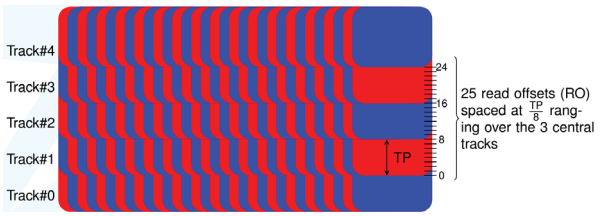

Fig. 3. Schematic of dataset #2 model geometry.

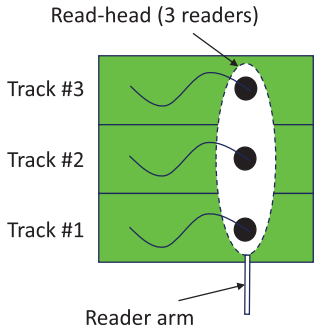

Fig. 4. Read-head model.

we only consider read-TMR and there is no simulation of write-TMR.

Fig. 3 shows a schematic of the dataset #2 model. The red and blue colors represent bits magnetized up (+1) and down (−1). Track 0 was the first track to be written. Next, track 1 was written, partially overlapping (erasing) part of track 0. The writing process continued, ending with track 4. Since track 4 was the last track to be written, it was not shingled and preserved the original magnetic write width (MWW) of 75 nm.

As shown in Fig. 3, there were 25 read-head offsets, numbered from 0 to 24, spaced at intervals of TP/8 (1.875 nm) and covering tracks 1, 2, and 3. The centers of these three tracks were close to the read-head offsets 4, 12, and 20. Readback signals from tracks 1, 2, and 3 were used as inputs to the TDMR system. In total, 100 different blocks (sectors) were calculated to investigate the effect of read-TMR.

The boundary track information is assumed to be provided by two additional readers not ganged together with the central three readers. It is assumed that the two outer boundary tracks are processed by a relatively simple 1-D BCJR algorithm, and the assumed BER for those tracks is set to be consistent with that assumption. Because the boundary track information BER is relatively high due to simplified processing, we do not account for reader TMR in the boundary tracks.

III. Read- and Write-TMR Model

We synthesize position errors by driving a first-order low-pass filter (LPF) with an independent and identically distributed (i.i.d.) Gaussian random process with transfer function

$$H(z) = \frac{1-\alpha}{1-\alpha z^{-1}}. \tag{2}$$

The downtrack coherence length of the TMR is controlled by the value of α. For datasets #1, α is set such that the impulse response falls to 0.5 after about 66 000 bits corresponding to about two 4 kbyte sectors, i.e., $\alpha = 0.9999788$.

Algorithm 1 Read-TMR

1. Tracks 0 and 4 have zero read offset, and from them we obtain readings $r_0(k)$ and $r_4(k)$ at bit position k.
2. Generate one TMR read random process $z(k)$, $-o_r \le z(k) \le o_r$, $o_r = 3$, with transfer function $H(z)$ as in (2).
3. For central track i, $i = 1, 2, 3$, linearly interpolate the two readings $r_{i,\lfloor z(k)\rfloor}(k)$ and $r_{i,\lceil z(k)\rceil}(k)$ to compute the readings $r_i(k)$, where $r_{i,l}(k)$ indicates the lth reader offset, $-o_r \le l \le o_r$, for the ith track at bit position k.

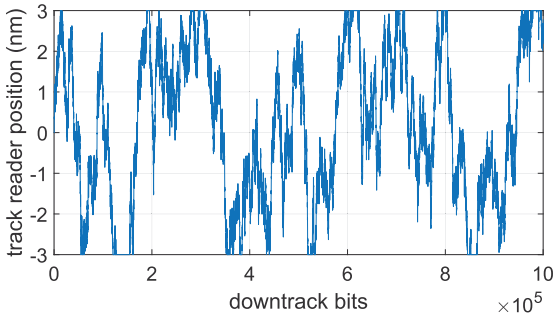

Fig. 5. Example of position error used in the simulations. The actual disturbance is quantized to the seven integer nanometer values from −3 to +3 nm.

For the power spectral density (PSD) analysis, if we assume a Gaussian random variable with zero mean and variance $N_0/2$ as the input, the PSD of the input would be $S_X(z) = N_0/2$. Hence, by considering $H(z)$ as in (2) and $z = e^{2j\pi f}$, the PSD of the LPF output can be computed as follows:

$$S_Y(f) = \left(\frac{N_0}{2}\right)\frac{(1-\alpha)^2}{1-2\alpha\cos(2\pi f)+\alpha^2}. \tag{3}$$

Therefore, the value f_0 such that $S_Y(f)$ drops to 0.5 of its maximum value is equal to $f_0 = 3.367 \times 10^{-6}$ or $1/f_0 \approx 297$ kbits.

For the read-TMR, we consider only read misregistration. The written tracks are assumed to be placed perfectly (i.e., no write-to-write TMR); all three read heads move simultaneously by the same amount. The read-head model in which the three read heads are moving together as a group (ganged together) is shown in Fig. 4. To consider the read-TMR, we read the signals from $\mathbf{r}_0$ and $\mathbf{r}_4$ for tracks 0 and 4 due to the zero read offset they have. Then, we generate a TMR read random process $z(k)$ based on (2) and interpolate the two readings $r_{i,\lfloor z(k)\rfloor}(k)$ and $r_{i,\lceil z(k)\rceil}(k)$, $i = 1, 2, 3$ to compute the readings $r_i(k)$, where $r_{i,l}(k)$ indicates the lth reader offset for the ith track at the bit position k. The process to generate the read-TMR is shown in Algorithm 1; this algorithm applies to both dataset #1a (with reader positions separated by 1 nm) and dataset #2 (with reader positions separated by 2 nm) under the assumption of nominal writer position for all tracks. Fig. 5 shows an example for the position error disturbance to produce the read-TMR.

For the write-TMR, the positions of tracks $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$, and $\mathbf{a}_{b,R}$ are varied independently, while $\mathbf{a}_{b,L}$ and $\mathbf{a}_{b,R_2}$ are fixed. Since the individual disturbances are truncated at ±3 nm, the maximum variation in TP or squeeze is ±6 nm. The GSP write

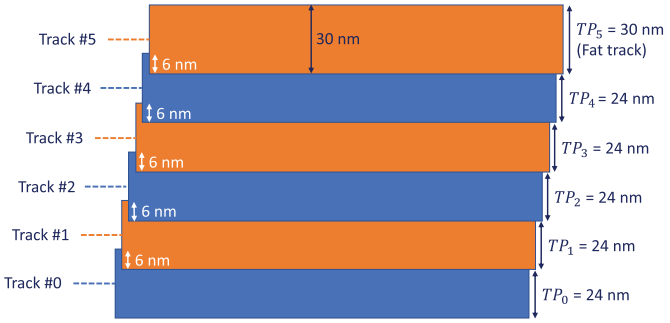

Fig. 6. Nominal position of the write heads for TP 24 nm dataset #1b.

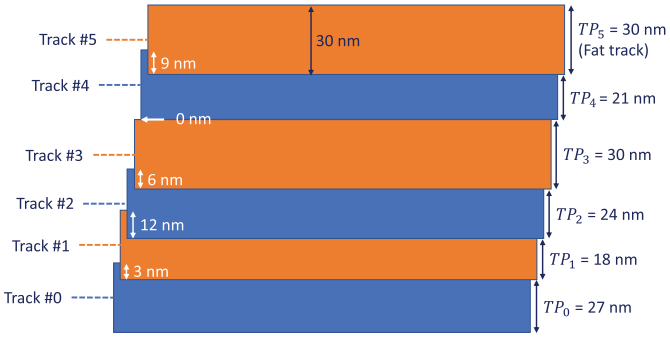

Fig. 7. Position error due to the write-TMR for TP 24 nm dataset #1b.

model first creates the sets of six adjacent 4 kbyte blocks in all of the possible $7^4 = 2401$ written track positions. Readback waveforms are then created for each with seven possible reader positions. The three readers are ganged together nominally covering tracks $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$ for a total of 16 807 possible sets of waveforms to sample from. Then, the addition, independently, of the white Gaussian noise adds a further degree of randomization.

To generate the write-TMR reading signals, we produce four independent TMR write random processes based on (2) and quantize each of them to a number from 0 to 6. By using these random processes, we generate a four-digit base-seven number m_k at downtrack bit k. Then, we select a block from dataset #1b with the index m_k and read readings $\mathbf{r}_0, \ldots, \mathbf{r}_4$ for that block. The nominal writer positions are represented in Fig. 6. As an example, track centers 1, 2, 3, and 4 are shifted by $+3$, -3, -3, and $+3$ nm, respectively, which are shown in Fig. 7. In Figs. 6 and 7, track #5 is shown as the fat track, which corresponds to track $\mathbf{a}_{b,R_2}$ shown in Fig. 1. Algorithm 2 represents the process of producing the write-TMR and only applies to datasets #1b.

To produce the joint write- and read-TMR, we generate four independent TMR write random processes to select a block from dataset #1b similar to Algorithm 2. Then, we produce a TMR read random process and linearly interpolate the two corresponding readings to compute the readings $\mathbf{r}_0, \ldots, \mathbf{r}_4$ similar to Algorithm 1. The process of generating the joint write- and read-TMR is presented in Algorithm 3; this algorithm only applies to datasets #1b.

IV. CNN Equalizer Design

We investigate the CNN-based equalizer for the dataset #1 model to use in the TDMR system. In [21, Fig. 10], the CNN PR equalizers are proposed for the TDMR case. As a comparison baseline, we also design and test with linear PR equalizers designed per the method described in [22].

Algorithm 2 Write-TMR

1. Tracks 0 and 5 are fixed at the nominal writer position (zero TMR).
2. Tracks 1, 2, 3, and 4 are written at one of 7 possible write-head positions, shifts $-o_w$ to $+o_w$, $o_w = 3$, to produce $7^4 = 2401$ blocks of readings of size 4 kbytes each.
3. The blocks are indexed with a numbering scheme from which we can deduce the write-head positions of tracks 1 to 4. The 4-digit base-seven numbers are used, i.e. 0000 through 6666, with 0 through 6 corresponding to write-position offsets $-o_w$ to $+o_w$, $o_w = 3$.
4. Generate four independent TMR write random processes $v_1(k), \ldots, v_4(k)$, $-o_w \le v_i(k) \le o_w$, $o_w = 3$, with transfer function $H(z)$ as in (2).
5. At downtrack bit k, quantize each of the four write-TMRs $v_i(k)$ to integer numbers between 0 and 6 to produce a 4-digit base-seven integer number m_k.
6. Select a block from the writer dataset with the index m_k in step 5, and from that block we obtain the readings $r_i(k)$ for the five tracks from $i = 0$ to $i = 4$.

Algorithm 3 Joint Write- and Read-TMR

1. Select a block from the writer dataset with index m_k, as in steps 1-5 of Algorithm 2.
2. Using the selected block m_k in step 1, compute the readings $r_0(k)$ through $r_4(k)$ as in Algorithm 1.

A. Nonlinear CNN Equalizer System

The CNN equalizer and PR target are jointly trained by an iterative algorithm to minimize the MSE between the equalized outputs and the equalization target; this process is described in detail in [14] and [21]. The readings $\mathbf{r}_0, \ldots, \mathbf{r}_4$ are provided to the CNN equalizer. The equalized waveforms $\mathbf{y}$ are produced for tracks 1, 2, and 3 by the CNN. Since the size of the window for the input image of CNN equalizer is 17, the readings to be equalized use a sliding window of size $[5 \times 17]$. A $[3 \times 3]$ PR target is employed for the equalizer. The process of training the CNN equalizer and PR target is shown in [14, Fig. 4] and [21, Fig. 10].

B. CNN Equalizer Architecture

The proposed CNN equalizer contains 28 layers. These layers consist of one input image layer, five convolutional units, a fully connected (FC) layer, and one output layer. Each 5×17 block of reader signals used as inputs to the CNN equalizer is normalized to have zero mean and unit variance. The architecture of the CNN equalizer is shown in Fig. 8.

Details of a very similar CNN equalizer architecture are described in [16]; the difference is that the architecture in this present article processes three tracks simultaneously, but the CNN equalizer in [16] processes two tracks at a time.

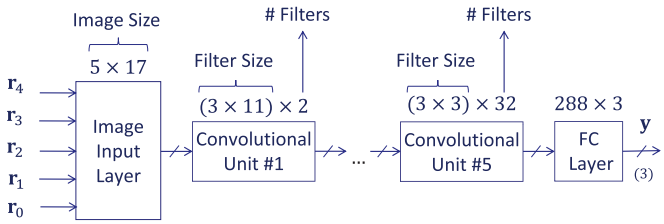

Fig. 8. CNN equalizer architecture for dataset #1 model.

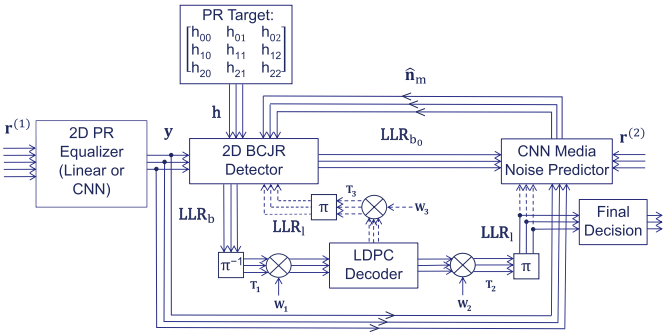

Fig. 9. Block diagram for the TDMR turbo-detection system.

In this present article, we utilize the leaky rectified linear unit (ReLU) and the exponential linear unit (ELU) layer as the first and second activation functions in the CNN equalizer, respectively. The leaky ReLU function with the slope of 0.1 for the negative numbers and the ELU function with non-linearity parameter 1 for negative numbers are employed. The dropout layer before the leaky ReLU layer sets input elements to zero randomly with a probability of 0.01. The output layer is a regression layer. The CNN generates the equalized waveform **y** for tracks 1, 2, and 3. These equalized waveforms are passed to the 2-D BCJR to detect the bits.

V. BCJR-LDPC-CNN Turbo-Detection System

A. Turbo-Detector

In Fig. 9, the block diagram for the TDMR turbo-detection system is shown. We use a trellis detector (a BCJR detector in this work) for TDMR to estimate the LLRs by using the 2-D equalized waveforms. A similar system was employed in [16] and [21], but in those works, there was no TMR in the read-head input waveforms **r**.

In this system, we separate trellis-based ISI/ITI detection and CNN-based media noise prediction. For dataset #2 (with reader positions separated by 2 nm), the simulated HDD read-head output vector **r** contains two samples per coded bit, denoted $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$. These samples are on the same track and are collected by the same read head, but are located at different downtrack locations within a given bit; the odd samples $\mathbf{r}^{(1)}$ (the "first samples" per bit) are located near the center of each bit, and the even samples $\mathbf{r}^{(2)}$ are located at the boundary between bits. The odd samples $\mathbf{r}^{(1)}$ are first filtered by a 2-D linear equalizer designed to minimize the MSE between the filter output **y** and the convolution of the coded bits **u** with the PR mask **h**. For the dataset #1 model, we only have one sample per coded bit denoted as **r**. The samples **r** are equalized with either linear or CNN-based equalizer. The equalized waveforms **y** are input to the 2-D BCJR detector.

The 2-D BCJR is a joint ISI/ITI equalizer; the state-input block has three rows because the ITI typically extends over one adjacent track on either side. In 2-D BCJR, the processing of three tracks is done simultaneously to handle the ITI from a 3×3 PR target mask since the central track is affected by the ITI from its two neighboring tracks. Sayyafan et al. [21, Fig. 5] represented the state-input block for 2-D BCJR over three tracks. The 2-D BCJR trellis detector performs ISI/ITI equalization on filtered input **y** and generates LLR outputs. The PR target **h** is of size 3×3; hence, the system has two state bits per track, the 2-D BCJR state-input window is 3×3, and the trellis has $2^{3\times 2} = 64$ states.

The boundary tracks can be read using two passes of the reader with three read heads ganged together. The first boundary track $\mathbf{a}_{b,L}$ plus center tracks $\mathbf{a}_1$ and $\mathbf{a}_2$ can be read in one pass, and the center tracks $\mathbf{a}_2$ and $\mathbf{a}_3$ plus the boundary track $\mathbf{a}_{b,R}$ can be read in a second pass. Then, the boundary track reading sequences can be individually processed with a simple 1-D BCJR in order to efficiently estimate the boundary bits. The boundary bits can tolerate a BER of 5–10%, and thus, a simple 1-D BCJR without channel decoding is sufficient [23]. Also, in a typical serial readback scenario (e.g., streaming movie), the system will typically already have one of the boundary tracks available from the last TDMR reader pass, so the system could do one pass to pick up center tracks 1–3 and one additional pass to pick up the additional boundary track. Two passes instead of one would drop system throughput gain relative to 1-D PDNP from $3\times$ to $1.5\times$ but still give a throughput gain.

The outer tracks 0 and 4 provide boundary information for the turbo-detector's 2-D BCJR detector. Under the assumption that these two tracks are processed by a simple 1-D BCJR detector and based on measured BERs from such a detector, we inject errors at a BER of 5.42% into the input bits for the outer tracks $\mathbf{a}_{2,L}$ and $\mathbf{a}_{2,R}$ before they are passed to the 2-D BCJR detector.

In Fig. 9, the CNN MNP takes as its inputs the three-track-wide block of LLRs $\mathbf{LLR}_{b_0}$ from the 2-D BCJR and the three-track-wide equalized sample vector **y** from the equalizer. Processing proceeds block-wise, where each block contains readings (or LLRs) corresponding to 4 kbytes (respectively, 41.2 kbits) per track of coded bits for dataset #1 (respectively, dataset #2). The CNN MNP sends its three-track-wide media noise estimate $\hat{\mathbf{n}}_m$ back to the 2-D BCJR detector for a second pass in order to obtain a lower detector BER. The BCJR output LLRs $\mathbf{LLR}_b$ from this second pass are scaled (attenuated) and magnitude limited and then proceed through a de-interleaver (denoted as "π^{-1}" in Fig. 9) in order to decorrelate them; then, they are input to the channel decoder. Scaling and magnitude limiting is done because, due to estimation errors in the PR target and the media noise, the LLRs in the vector $\mathbf{LLR}_b$ are typically "overconfident" (larger in magnitude) relative to their true underlying probabilities.

The channel codes used in this article are irregular repeat accumulate (IRA) codes, which are a type of LDPC code with low encoding complexity [24]. As the GSP bits are

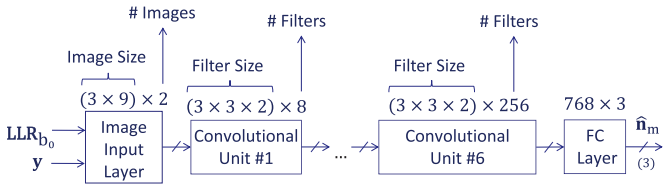

Fig. 10. 3-D CNN architecture of MNP for dataset #1.

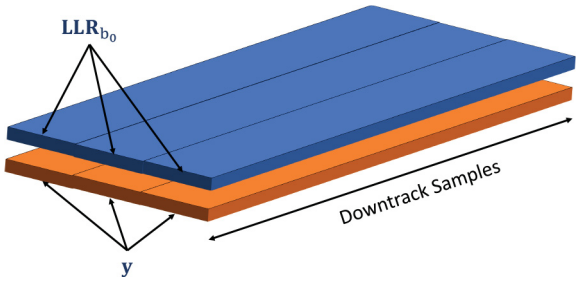

Fig. 11. 3-D input image for the CNN MNP for dataset #1.

randomly distributed, the IRA decoder employs coset decoding. We assume independent per-track encoding and decoding so that different tracks may have different code rates. The decoder output LLRs $\mathbf{LLR}_l$ are scaled and magnitude limited, and then, they are interleaved before being thresholded at zero for the final bit decisions. Instead of making a final decision, the interleaved $\mathbf{LLR}_l$s may be passed to the 2-D BCJR detector for use as extrinsic information in a second turbo iteration; in this case, they are also passed to the CNN MNP to form a second-iteration media noise estimate, which is also passed to the 2-D BCJR detector [21]. The $\mathbf{LLR}_l$ data paths for this optional second iteration are shown as dotted lines in Fig. 9, since in this article, only one iteration of LDPC decoding is done. Additional turbo iterations will be investigated in future work.

B. CNN MNP Architecture

We investigate the architecture of the CNN MNP in the TDMR system. A 3-D CNN for the TDMR system is considered as the architecture. A similar architecture is described in [16]; here, we focus only on the differences in the present article's architecture. Fig. 10 shows the 3-D CNN noise predictor architecture for datasets #1. The 3-D CNN has 21 layers, including one input image layer, six convolutional units, an FC layer, and one output layer. Different from the convolutional units in the CNN equalizer which have five layers, the convolutional units in the CNN noise predictor have three layers, including convolutional layer, batch normalization layer, and activation function. The leaky ReLU is used with the slope of 0.1 as the activation function for the CNN noise predictor. In Fig. 11, stacking of the 2-D input images (with three tracks each) into a 3-D input image is shown for datasets #1. In the designed 3-D CNNs for TDMR, the convolutional layers have filters with the size of $[3 \times 3 \times 2]$.

For dataset #2, in addition to $\mathbf{y}$ and the $\mathbf{LLR}_b$, the even samples $\mathbf{r}^{(2)}$ are passed to the CNN MNP because experiments have shown that this leads to better media noise prediction. The $\mathbf{r}^{(2)}$ are added as a third input to the image input layer in Fig. 10 and as a third layer to the 3-D input image shown in Fig. 11. For the 3-D CNN architecture of dataset #2, we use five (instead of six) convolutional units and the ReLU (instead of leaky ReLU) as the activation function in Fig. 10.

VI. 1-D PDNP With LLR Exchange

The 1-D PR target equalizer is used to equalize the per-track ISI, not the cross-track ITI. Because the central track experiences ITI from its two adjacent tracks, we subtract the ITI from both these tracks by passing their estimated soft bits to the central track. Fig. 12 shows the block diagram of the 1-D PDNP system with LLR exchange. The three 1-D BCJR/PDNPs operate on tracks $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$ in parallel. We hypothesize that exchanging the LLRs between the 1-D BCJR/PDNPs of the outer and inner tracks can reduce ITI and improve the quality of the output LLRs for the central track. In this system, the BER and AD are assessed on $\mathbf{a}_2$. The arrows from outer 1-D BCJR/PDNPs to the inner one in Fig. 12 show the LLR exchange from the outer tracks to the central track. Each 1-D BCJR/PDNP could reduce the ITI by assuming some weights for mapping the expected value of coded bits $(\mathbf{u}_1, \mathbf{u}_3)$ to the central track. For each outer tracks' 1-D BCJR/PDNP, the mapping weights $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_3$ are obtained by solving the following least-squared error (LSE) problem:

$$\min_{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_3} \left\| \mathbf{y}_2 - (\mathbf{h}_2 * \mathbf{u}_2 + \boldsymbol{\alpha}_1 * \mathbf{s}_1 + \boldsymbol{\alpha}_3 * \mathbf{s}_3) \right\|^2 \tag{4}$$

where $\mathbf{s}_i, i \in \{1, 3\}$ are the soft bits for the block $\mathbf{b}_i$ of 4 kbytes in track i and $\mathbf{y}_2$ are the equalized waveforms for central track 2 from (1). We pass soft bits, by passing the expected value of each bit to the inner track. The expected value for bits $\mathbf{b}_i$ can be computed from $\mathbf{LLR}_i$ by first converting the LLRs to the probabilities $\mathbf{p}_i$ that $\mathbf{b}_i = 1$

$$\mathbf{p}_i = \exp(\mathbf{LLR}_i)/(\exp(\mathbf{LLR}_i) + 1), \quad i \in \{1, 3\}. \tag{5}$$

Hence, we compute the soft bits $\mathbf{s}_i$ as follows:

$$\begin{aligned} \mathbf{s}_i &= \mathbb{E}[\mathbf{b}_i] = \mathbf{p}_i \times (+1) + \left(1 - \mathbf{p}_i\right) \times (-1) \\ &= (\exp(\mathbf{LLR}_i) - 1)/(\exp(\mathbf{LLR}_i) + 1), \quad i \in \{1, 3\}. \end{aligned} \tag{6}$$

Since for the central track, the PR target includes three taps, we consider three samples in this track and six associated samples in the outer tracks. For each outer track, we map three samples to the inner track. Hence, by solving (4), the corresponding weights $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_3$ with size of $[1 \times 3]$ for the two outer tracks are obtained. For dataset #1b model with joint write- and read-TMR, the computed values of $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_3$ for TP 24 nm dataset are

$$\begin{bmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_3 \end{bmatrix} = \begin{bmatrix} 0.0478 & 0.1624 & 0.0274 \\ 0.0705 & 0.2017 & 0.0522 \end{bmatrix} \tag{7}$$

and for TP 18 nm dataset equal to

$$\begin{bmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_3 \end{bmatrix} = \begin{bmatrix} -0.0049 & 0.1488 & -0.0245 \\ 0.0366 & 0.2153 & 0.0042 \end{bmatrix}. \tag{8}$$

By using these weights to map the outer tracks' signals to the central track, for dataset #1b model in the presence of joint write- and read-TMR, the inner track's MSE [computed via (4)] decreases by 8.85% and 18.60% for TP 24 and

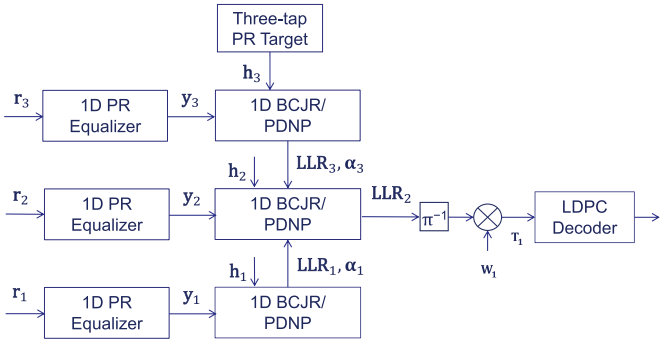

Fig. 12. Block diagram for 1-D PDNP with LLR exchange.

18 nm datasets, respectively. This MSE reduction leads to the AD improvement for the central track. This LLR exchange is more beneficial at lower TP (i.e., TP 18 nm).

After passing the outer tracks' LLRs to the 1-D PDNP/BCJR of the inner track, we use the iterative row-column soft decision feedback algorithm (IRCSDFA) for the bit detection in the inner track's 1-D BCJR/PDNP as follows [25]. Given input vector $\mathbf{u}_{2,k} = [u_{2,k}, u_{2,k-1}, u_{2,k-2}]$, the current state $S_k = (u_{2,k-1}, u_{2,k-2})$. We denote the current equalized sample as y_{2_k} and branch bits $\mathbf{i} = [i_0, \ldots, i_{n_b-1}], i_m \in \{-1, +1\}$, where n_b is the number of input per trellis stage; n_b is one in our work. The modified conditional channel probability density function (pdf) $p'(\cdot)$ sums over $\boldsymbol{\alpha}_1 * \mathbf{s}_1 + \boldsymbol{\alpha}_3 * \mathbf{s}_3$ associated with the six corresponding outer track bits, for the state transition $s' \to s$

$$p'\left(y_{2_k} | u_k = i_0, S_k = s, S_{k-1} = s'\right) = P\left(y_{2_k} | u_{2,k}, s, s', \mu_k + \mu([\mathbf{s}_1, \mathbf{s}_3])\right) \quad (9)$$

where

$$\mu_k = \mathbf{h}_2 * \mathbf{u}_{2,k} = h_{2,0}u_{2,k} + h_{2,1}u_{2,k-1} + h_{2,2}u_{2,k-2} \quad (10)$$

$$\mu([\mathbf{s}_1, \mathbf{s}_3]) = \sum_{[\mathbf{s}_1, \mathbf{s}_3]} P([\mathbf{s}_1, \mathbf{s}_3]) \times (\boldsymbol{\alpha}_1 * \mathbf{s}_1 + \boldsymbol{\alpha}_3 * \mathbf{s}_3) \quad (11)$$

$\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_3$ are obtained in (4), $[\mathbf{s}_1, \mathbf{s}_3]$ is the vector of six outer track bits that correspond to the inner track bits $\mathbf{u}_k$, and the row probabilities

$$P\left([\mathbf{s}_1, \mathbf{s}_3] = \left[s_{1,0}, s_{1,1}, s_{1,2}, s_{3,0}, s_{3,1}, s_{3,2}\right]\right) = \prod_{i \in \{1,3\}} \prod_{j=0}^{2} P\left(l_{i,j}\right) \quad (12)$$

such that $P(l_{i,j})$ are the bit probabilities derived from the LLRs passed from the outer tracks to the inner track. In this case, the 1-D BCJR/PDNP in the central track can utilize the LLRs from outer tracks to reduce ITI and detect the coded bits with a lower BER.

VII. Simulation Results

This section presents the simulation results for the BCJR-LDPC-CNN turbo-detector on a TDMR channel in the presence of write- and read-TMR. The simulations for no TMR and read-TMR alone systems are performed on dataset #2 with TP 15 nm and for datasets #1a with TPs 18 and 24 nm. For the only write-TMR and joint write- and read-TMR cases, the simulation is conducted on datasets #1b with TPs 18 and 24 nm. We provide the results for both linear and CNN-based PR equalization before the turbo-detector.

The BCJRs initially assume that the media noise is zero and compute an initial set of output LLRs $\mathbf{LLR}_{b_0}$. The CNN MNPs are provided with the LLR values for the dataset #1 model. The LLR probabilities, the estimated bit, and the estimation's reliability can be determined by the LLR values. For the dataset #2 model, the LLR probabilities are passed to the CNN MNP. For this model, in [26], it is experimentally shown that CNN provided with LLR probabilities performs better compared to when it uses the signed LLR values. This might be due to the non-linear scale inherent in the LLRs.

A. Datasets

For the dataset #1 model with 10 nm BL and 18 or 24 nm TP, each block contained $6 \times N_u$ input bits, where $N_u = 32\,768$. For the simulations on datasets #1a with no TMR and read-TMR alone, we used 59, 1, and 40 blocks as the training, validation, and test datasets, respectively, for both the CNN equalizer and the BCJR-LDPC-CNN detector in the CNN-based system. For 1-D PDNP, we used 1 and 40 blocks for the training and testing processes, respectively. For the simulations on datasets #1b with write-TMR alone and with write- and read-TMR, 400, 100, and 200 blocks were used as the training, validation, and test datasets, respectively, for both the CNN equalizer and BCJR-LDPC-CNN detector. For 1-D PDNP, 400 and 100 blocks were used for the training and testing processes, respectively.

For the dataset #2 model with 11 nm BL and 15 nm TP, we used 80 and 20 blocks as the training and test datasets for the CNN-based system and 1-D PDNP.

B. Simulation Parameters

For the 1-D PDNP comparison baseline, the BCJR output LLR multiplicative weight is set to 0.5, and the magnitude limit is set to 100. For the TDMR system of Fig. 9, the corresponding numbers are 0.7 and 10. The IRA decoder performs a minimum of 200 and a maximum of 400 iterations of the LDPC sum-product decoding algorithm. The IRA decoder's output LLRs are weighted by 0.65 and magnitude limited to 100. Achieved AD is computed from the highest code rate that gives a final decoded BER $\leq 10^{-5}$. Different IRA code rates are obtained by puncturing a "base rate" (lowest rate) "base code" in order to realize higher rates, per the realistic puncturing scheme in [23]. AD is computed as

$$\text{Areal density} = \text{achieved-code-rate} / (\text{BL} \times \text{TP}). \quad (13)$$

Tables I and II summarize the results for the TDMR detectors. The raw channel BER for the dataset #2 is reported for the detected bits on the input readings of the three central tracks. For datasets #1, the raw channel BER is the average detected errors on the three central tracks $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$. In addition to the BER, a 95% confidence upper bound on the BER [27] is provided in parentheses in the sixth columns

TABLE I
NO TMR AND READ-TMR SIMULATION RESULTS. NP MEANS NOISE PREDICTION

Method	TP (nm)	Raw Channel BER	AD ($\mathrm{Tb/in^2}$)	Code Rate	Decoded BER	AD Gain/Reduction (%)
1-D PDNP, Lin.-Eq.	15	0.2058	1.724	0.4400	0 (6.67e-7)	0
1-D PDNP, Lin.-Eq., R TMR	15	0.2063	1.713	0.4372	0 (6.67e-7)	0
1-D PDNP, Lin.-Eq., LLR Exch. (cb. 1)	15	0.2058	1.764	0.4501	0 (6.67e-7)	+2.32
1-D PDNP, Lin.-Eq., LLR Exch., R TMR	15	0.2063	1.735	0.4428	0 (6.67e-7)	+1.28 / −1.64
CNN NP, Lin.-Eq. (cb. 2)	15	0.2058	3.890	0.9927	0 (6.67e-7)	0
CNN NP, Lin.-Eq., R TMR	15	0.2063	3.887	0.9918	0 (6.67e-7)	−0.08
CNN NP, Lin.-Eq., R TMR, SNR 20 dB	15	0.2081	3.883	0.9909	0 (6.67e-7)	−0.18
1-D PDNP, Lin.-Eq.	18	0.1108	2.7481	0.7667	0 (7.63e-7)	0
1-D PDNP, Lin.-Eq., R TMR	18	0.1178	2.6357	0.7353	0 (7.63e-7)	0
1-D PDNP, Lin.-Eq., LLR Exch. (cb. 3)	18	0.1108	2.9179	0.8141	0 (7.63e-7)	+6.18
1-D PDNP, Lin.-Eq., LLR Exch., R TMR	18	0.1178	2.7588	0.7697	0 (7.63e-7)	+4.67 / −5.45
CNN NP, Lin.-Eq. (cb. 4)	18	0.1108	3.0046	0.8383	0 (7.63e-7)	0
CNN NP, Lin.-Eq., R TMR	18	0.1178	2.8362	0.7913	0 (7.63e-7)	−5.60
CNN NP, CNN-Eq., R TMR	18	0.1178	2.8591	0.7977	0 (7.63e-7)	−4.84
1-D PDNP, Lin.-Eq.	24	0.0833	2.3474	0.8732	0 (7.63e-7)	0
1-D PDNP, Lin.-Eq., R TMR	24	0.0872	2.2781	0.8474	0 (7.63e-7)	0
1-D PDNP, Lin.-Eq., LLR Exch. (cb. 5)	24	0.0833	2.3922	0.8899	0 (7.63e-7)	+1.91
1-D PDNP, Lin.-Eq., LLR Exch., R TMR	24	0.0872	2.2937	0.8533	0 (7.63e-7)	+0.68 / −4.12
CNN NP, Lin.-Eq. (cb. 6)	24	0.0833	2.3880	0.8883	0 (7.63e-7)	0
CNN NP, Lin.-Eq., R TMR	24	0.0872	2.3016	0.8562	0 (7.63e-7)	−3.62

of Tables I and II. In the case of non-zero errors, the BER upper bound p is computed as

$$I_p(x+1, N_{\text{tcb}} - x) = \gamma \tag{14}$$

where I_p indicates the beta distribution with parameters $x+1$ and $N_{\text{tcb}} - x$ for γ-quantile, x is the number of errors, N_{tcb} is the total number of transmitted coded bits, and γ is the confidence threshold, which we set to 0.95. In the case of zero error count, the BER upper bound in (14) simplifies to approximately $3/N_{\text{tcb}}$ [27].

C. Discussion of Simulation Results

Table I summarizes the results for datasets #1a and dataset #2 for the TDMR system with no TMR and read-TMR alone. Table I compares the AD performance of the proposed BCJR-LDPC-CNN detectors with 1-D PDNP. The "1-D PDNP, Lin.-Eq." utilizes the equalized waveforms from a linear minimum MSE equalizer using a 1-D PR target. The 1-D PDNP has 128 trellis states, corresponding to $I = 2$, $L = 4$, and $\Delta = 1$, where I is the ISI channel length, L is the predictor memory, and Δ is the predictor look ahead. The pattern vector length of $I + 1 + L + \Delta = 8$ bits of 1-D PDNP is comparable to the channel inputs $\mathbf{y}$ length of nine sample bits for the CNN. The "CNN NP, Lin.-Eq." employs a linear equalizer and a CNN noise predictor in the turbo-detection system. Also, "CNN NP, CNN-Eq." represents utilizing the CNN-based equalizer and noise predictor. The read-TMR, write-TMR, and joint write- and read-TMR are shown as "R TMR," "W TMR," and "W-R TMR," respectively.

As the reference, we modify the standard 1-D PDNP by soft ITI subtraction, referred to as "1-D PDNP with LLR exchange" (see Section VI). As seen in Table I, passing the LLRs from outer tracks to the inner track increases the code rate for 1-D PDNP on track $\mathbf{a}_2$. For the dataset #2 model with TP 15 nm, the code rate improves by 2.32% and 1.28% with no TMR and only read-TMR compared to the standard 1-D PDNP, respectively. Also, for the dataset #1a model, the soft ITI subtraction increases the code rate by 6.18% and 1.91% for the TP 18 and 24 nm datasets, respectively, without considering any TMR.

In Tables I and II, "cb." stands for comparison baseline used for comparing the AD of other methods. The columns "AD Gain/Reduction" in Tables I and II represent the AD gain or reduction of each method compared to the corresponding reference method. The entities in this column represent the AD gain if their values are positive and represent the AD reduction if their values are negative.

TABLE II
WRITE-TMR AND JOINT WRITE- AND READ-TMR SIMULATION RESULTS

Method	TP (nm)	Raw Channel BER	AD ($\mathrm{Tb/in^2}$)	Code Rate	Decoded BER	AD Gain/Reduction (%)
1-D PDNP, Lin.-Eq., LLR Exch. (cb. 3)	18	0.1108	2.9179	0.8141	0 (7.63e-7)	0
1-D PDNP, Lin.-Eq., LLR Exch., W TMR	18	0.1261	2.5688	0.7167	0 (1.53e-7)	−11.96
1-D PDNP, Lin.-Eq., LLR Exch., W-R TMR	18	0.1304	2.4789	0.6916	0 (1.53e-7)	−15.04
CNN NP, Lin.-Eq. (cb. 4)	18	0.1108	3.0046	0.8383	0 (7.63e-7)	0
CNN NP, Lin.-Eq., W TMR	18	0.1261	2.7588	0.7697	0 (1.53e-7)	−8.18
CNN NP, Lin.-Eq., W-R TMR	18	0.1304	2.6654	0.7436	0 (1.53e-7)	−11.29
CNN NP, CNN-Eq., W TMR	18	0.1261	2.7805	0.7757	0 (1.53e-7)	−7.46
CNN NP, CNN-Eq., W-R TMR	18	0.1304	2.6856	0.7493	0 (1.53e-7)	−10.62
1-D PDNP, Lin.-Eq., LLR Exch. (cb. 5)	24	0.0833	2.3922	0.8899	0 (7.63e-7)	0
1-D PDNP, Lin.-Eq., LLR Exch., W TMR	24	0.0876	2.2440	0.8347	0 (1.53e-7)	−6.20
1-D PDNP, Lin.-Eq., LLR Exch., W-R TMR	24	0.0918	2.1795	0.8108	0 (1.53e-7)	−8.89
CNN NP, Lin.-Eq. (cb. 6)	24	0.0833	2.3880	0.8883	0 (7.63e-7)	0
CNN NP, Lin.-Eq., W TMR	24	0.0876	2.2695	0.8443	0 (1.53e-7)	−4.96
CNN NP, CNN-Eq., W TMR	24	0.0876	2.2793	0.8479	0 (1.53e-7)	−4.55
CNN NP, CNN-Eq., W-R TMR	24	0.0918	2.1947	0.8164	0 (1.53e-7)	−8.09

In most cases in Table I, the CNN-based systems achieve some gain over the PDNP systems. For dataset #1a with TP 18 nm, the CNN-based method gains 2.97% and 3.64% AD over the baseline with no TMR and only read-TMR cases, respectively. For dataset #1a with TP 24 nm, the CNN-based system achieves 0.34% AD gain over the baseline for the cases with read-TMR alone. For dataset #2, the proposed CNN-based method gains 120.52% and 124.03% AD over the baseline with no TMR and only read-TMR. As shown by the column "AD Gain/Reduction," in all cases, the CNN-based system is more robust to read-TMR compared to the PDNP-based systems.

Table II presents the results for datasets #1b for TDMR in the presence of only write-TMR and joint write- and read-TMR. For all cases, the CNN-based system improved the density and robustness to write-TMR and joint write- and read-TMR compared to the PDNP systems. Also, the CNN system with the CNN equalizer improves AD and robustness compared to PDNP. For dataset #1b with TP 18 nm, the CNN-based system achieves 8.24% and 8.34% AD gain over the baseline with only write-TMR and joint write- and read-TMR, respectively. For dataset #1b with TP 24 nm, the CNN-based system gains 1.57% and 0.70% AD over the baseline with only write-TMR and joint write- and read-TMR, respectively.

The base code rate is 0.6030 for all the simulations in Tables I and II. By puncturing the base code, we could achieve the highest code rate for every detector and layer. The ADs for each detector are computed using (13).

In Tables I and II, for each dataset, we consider "1-D PDNP, Lin-Eq., LLR Exch." and "CNN NP, Lin-Eq." with no TMR as the two references. The simulation results show that by using the CNN-based method, the system is more robust to TMR compared to the baseline. Also, by using the CNN-based equalizer, the system achieves higher AD and robustness compared to the linear equalizer. Thus, employing more CNN-based techniques in the turbo-detection leads to higher AD and more robustness to write- and read-TMR.

Both datasets #1 and #2 contain no read-head electronic AWGN, i.e., $n_e(k) = 0$ in (1). The rows without label "SNR 20 dB" report results for this case. The row that includes the label "SNR 20 dB" reports the result when non-zero AWGN n_e at an SNR of 20 dB is added to both readings $\mathbf{r}_i^{(1)}$ and $\mathbf{r}_i^{(2)}, i = 1, 2, 3$, for dataset #2. The SNR for dataset #2 is computed as

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\mathbb{E}\left[\sum_{i=1}^{3}\left(\left\|\mathbf{r}_i^{(1)}\right\|^2 + \left\|\mathbf{r}_i^{(2)}\right\|^2\right)\right]}{\sigma_e^2}\right) \tag{15}$$

where σ_e^2 indicates the AWGN variance. This SNR is computed based on all the coded bits of each track. The simulation result for the added AWGN at 20 dB SNR for dataset #2 with read-TMR is shown in Table I; it shows that for dataset #2 with read-TMR, the CNN-based system with zero electronic noise has an AD gain of a reasonably small amount (0.09%) over the same method with 20 dB SNR for AWGN.

These architectures rely on the CNN noise predictors and equalizers trained with the read- and write-TMR to adapt to the spatially varying TMR. However, these CNNs are static; they are trained once and remain in the system. Future work might consider adapting the network weights of either noise predictor or equalizer to adjust to the read- and write-TMR.

D. Computational Complexity Comparison

The computational complexity (per bit) figures for the TDMR detectors are represented in Table III. The reported

TABLE III
DETECTOR COMPLEXITY PER BIT FOR TDMR

Method	mul/div	add/sub	exp/log
1-D PDNP	16,266	15,626	257
1-D PDNP, LLR Exch.	49,520	47,359	777
CNN NP	7,713,804	5,874,232	343

numbers are computed without considering the equalizer. The 1-D PDNP has the lowest complexity among the evaluated detectors. The 1-D PDNP with LLR exchange detector is more than $3\times$ complex than 1-D PDNP due to the passing soft coded bits from the outer tracks into the central track, which requires 1-D-PDNP detectors on all three tracks. The reported numbers for "1-D PDNP" are for both datasets #1 and #2. The numbers for "1-D PDNP, LLR Exch." and "CNN NP" are reported for dataset #1. The "CNN NP" is the most complex method since it has six convolutional units to process the data and estimate the media noise.

VIII. CONCLUSION

This article presents the BCJR-CNN-LDPC architecture combined in a TDMR turbo-detection system in the presence of write- and read-TMR. In this work, we develop the spatially varying write- and read-TMR models. For the baseline, we modify the standard 1-D PDNP by ITI subtraction from adjacent tracks, called 1-D PDNP with LLR exchange, to improve its performance in the TDMR system. The simulation results show that the proposed turbo-detection system is more robust to TMR compared with 1-D PDNP with LLR exchange and achieves 8.34% and 0.70% AD gain over the baseline in the presence of joint write- and read-TMR for TP 18 and 24 nm datasets #1b, respectively. As future work, we will investigate adapting the network weights for the CNN noise predictor and equalizer to adjust to the read- and write-TMR. Also, the second iteration of LDPC decoding will be investigated to achieve higher AD gain.

ACKNOWLEDGMENT

This work was supported in part by the United States National Science Foundation under Grant CCF-1817083 and in part by the Advanced Storage Research Consortium (ASRC). The authors thank Roger Wood, Rick Galbraith, and Iouri Oboukhov for their advice and suggestions.

REFERENCES

[1] J. Moon and J. Park, "Pattern-dependent noise prediction in signal-dependent noise," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 4, pp. 730–743, Apr. 2001.
[2] E. M. Kurtas, J. Park, X. Yang, W. Radich, and A. Kavcic, "Detection methods for data-dependent noise in storage channels," in *Coding and Signal Processing for Magnetic Recording Systems*, B. Vasic and E. M. Kurtas, Eds. Boca Raton, FL, USA: CRC Press, Nov. 2004.
[3] J. R. Barry et al., "Optimization of bit geometry and multi-reader geometry for two-dimensional magnetic recording," *IEEE Trans. Magn.*, vol. 52, no. 2, pp. 1–7, Feb. 2016.
[4] G. Mathew, E. Hwang, J. Park, G. Garfunkel, and D. Hu, "Capacity advantage of array-reader-based magnetic recording (ARMR) for next generation hard disk drives," *IEEE Trans. Magn.*, vol. 50, no. 3, pp. 155–161, Mar. 2014.
[5] Y. Zheng, G. Mathew, T. Oenning, R. Rauschmayer, B. Wilson, and W. Hanson, "TMR sensitive equalization for electronic servoing in array-reader-based hard disk drives," *IEEE Trans. Magn.*, vol. 52, no. 2, Feb. 2016, Art. no. 3000407.
[6] W. Chang and J. R. Cruz, "Inter-track interference mitigation for bit-patterned magnetic recording," *IEEE Trans. Magn.*, vol. 46, no. 11, pp. 3899–3908, Nov. 2010.
[7] E. Banan Sadeghian and J. R. Barry, "Soft intertrack interference cancellation for two-dimensional magnetic recording," *IEEE Trans. Magn.*, vol. 51, no. 6, pp. 1–9, Jun. 2015.
[8] Y. Qin and J.-G. Zhu, "TDMR with machine learning data detection channel," *IEEE Trans. Magn.*, vol. 58, no. 2, pp. 1–5, Feb. 2022.
[9] Y. Qin et al., "CNN-based machine learning channel on TDMR drive data," *IEEE Trans. Magn.*, vol. 58, no. 4, pp. 1–7, Apr. 2022.
[10] J. Shen and N. Nangare, "Nonlinear equalization for TDMR channels using neural networks," in *Proc. 54th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2020, pp. 1–6.
[11] A. Aboutaleb and N. Nangare, "Reduced complexity neural network equalizers for two-dimensional magnetic recording," in *Proc. IEEE 33rd Magn. Recording Conf. (TMRC)*, Milpitas, CA, USA, Aug. 2022, pp. 1–2.
[12] J. Shen, A. Aboutaleb, K. Sivakumar, B. J. Belzer, K. S. Chan, and A. James, "Deep neural network a posteriori probability detector for two-dimensional magnetic recording," *IEEE Trans. Magn.*, vol. 56, no. 6, pp. 1–12, Jun. 2020.
[13] J. Shen, B. J. Belzer, K. Sivakumar, K. S. Chan, and A. James, "Convolutional neural network based symbol detector for two-dimensional magnetic recording," *IEEE Trans. Magn.*, vol. 57, no. 3, pp. 1–5, Mar. 2021.
[14] A. Aboutaleb et al., "Deep neural network-based detection and partial response equalization for multilayer magnetic recording," *IEEE Trans. Magn.*, vol. 57, no. 3, pp. 1–12, Mar. 2021.
[15] A. Aboutaleb et al., "A perspective on deep neural network-based detection for multilayer magnetic recording," *Appl. Phys. Lett.*, vol. 119, no. 1, pp. 1–12, Jul. 2021.
[16] A. Sayyafan, A. Aboutaleb, B. J. Belzer, K. Sivakumar, S. Greaves, and K. S. Chan, "Turbo-detection for multilayer magnetic recording using deep neural network-based equalizer and media noise predictor," *IEEE Trans. Magn.*, vol. 58, no. 4, pp. 1–11, Apr. 2022.
[17] S. J. Greaves, K. S. Chan, and Y. Kanai, "Areal density capability of dual-structure media for microwave-assisted magnetic recording," *IEEE Trans. Magn.*, vol. 55, no. 12, pp. 1–9, Dec. 2019.
[18] S. J. Greaves, K. S. Chan, and Y. Kanai, "Optimization of dual-structure recording media for microwave-assisted magnetic recording," *IEEE Trans. Magn.*, vol. 55, no. 7, pp. 1–5, Jul. 2019.
[19] K. S. Chan et al., "Channel models and detectors for two-dimensional magnetic recording," *IEEE Trans. Magn.*, vol. 46, no. 3, pp. 804–811, Mar. 2010.
[20] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284–287, Mar. 1974.
[21] A. Sayyafan et al., "Deep neural network media noise predictor turbo-detection system for 1-D and 2-D high-density magnetic recording," *IEEE Trans. Magn.*, vol. 57, no. 3, pp. 1–13, Mar. 2021.
[22] 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. Magn.*, vol. 51, no. 10, pp. 1–15, Oct. 2015.
[23] X. Sun et al., "ISI/ITI turbo equalizer for TDMR using trained local area influence probabilistic model," *IEEE Trans. Magn.*, vol. 55, no. 4, pp. 1–15, Apr. 2019.
[24] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. 2nd Int. Symp. Turbo Codes Rel. Topics*, Brest, France, Sep. 2000, pp. 1–8.
[25] T. Cheng, B. J. Belzer, and K. Sivakumar, "Row-column soft-decision feedback algorithm for two-dimensional intersymbol interference," *IEEE Signal Process. Lett.*, vol. 14, no. 7, pp. 433–436, Jul. 2007.
[26] A. Sayyafan, B. J. Belzer, K. Sivakumar, J. Shen, K. S. Chan, and A. James, "Deep neural network based media noise predictors for use in high-density magnetic recording turbo-detectors," *IEEE Trans. Magn.*, vol. 55, no. 12, pp. 1–6, Dec. 2019.
[27] F. Scholz. (2008). *Confidence Bounds and Intervals for Parameters Relating to the Binomial, Negative Binomial, Poisson and Hypergeometric Distributions With Applications to Rare Events*. [Online]. Available: https://faculty.washington.edu/fscholz/DATAFILES498B2008/ConfidenceBounds.pdf