Adaptive Read Thresholds for NAND Flash

Borja Peleato[*], Rajiv Agarwal[†], John Cioffi[†], Minghai Qin[‡], Paul H. Siegel[‡]

[*]Purdue University [†]Stanford University [‡]UCSD

Abstract

A primary source of increased read time on NAND flash comes from the fact that in the presence of noise, the flash medium must be read several times using different read threshold voltages for the decoder to succeed. This paper proposes an algorithm that uses a limited number of re-reads to characterize the noise distribution and recover the stored information. Both hard and soft decoding are considered. For hard decoding, the paper attempts to find a read threshold minimizing bit-error-rate (BER) and derives an expression for the resulting codeword-error-rate. For soft decoding, it shows that minimizing BER and minimizing codeword-error-rate are competing objectives in the presence of a limited number of allowed re-reads, and proposes a trade-off between the two.

The proposed method does not require any prior knowledge about the noise distribution, but can take advantage of such information when it is available. Each read threshold is chosen based on the results of previous reads, following an optimal policy derived through a dynamic programming backward recursion. The method and results are studied from the perspective of an SLC Flash memory with Gaussian noise for each level but the paper explains how the method could be extended to other scenarios.

I. INTRODUCTION

A. Overview

The introduction of Solid State Drives (SSD) based on NAND flash memories has revolutionized mobile, laptop, and enterprise storage by offering random access to the information

with dramatically higher read throughput and power-efficiency than hard disk drives. However, SSD's are considerably more expensive, which poses an obstacle to their widespread use. NAND flash manufacturers have tried to pack more data in the same silicon area by scaling the size of the flash cells and storing more bits in each of them, thus reducing the cost per gigabyte (GB) and making flash more attractive to consumers, but this cell-size shrinkage has come at the cost of reduced performance. As cell-size shrinks to sub-16nm limits, noise can cause the voltage residing on the cell at read time to be significantly different from the voltage that was intended to be stored at the time of write. Even in current state-of-the-art 19nm NAND, noise is significant towards the end of life of the drive. One way to recover host data in the presence of noise is to use advanced signal processing algorithms [1], [2], [3], [4], but excessive re-reads and post-read signal processing could jeopardize the advantages brought by this technology.

Typically, all post-read signal processing algorithms require re-reads using different thresholds, but the default read thresholds, which are good for voltage levels intended during write, are often suboptimal for read-back of host data. Furthermore, the noise in the stored voltages is random and depends on several factors such as time, data, and temperature; so a fixed set of read thresholds will not be optimal throughout the entire life of the drive. Thus, finding optimal read thresholds in a dynamic manner to minimize BER and speed up the post-processing is essential.

The first half of the paper proposes an algorithm for characterizing the distribution of the noise for each nominal voltage level and estimating the read thresholds which minimize BER. It also presents an analytical expression relating the BER found using the proposed methods to the minimum possible BER. Though BER is a useful metric for algebraic error correction codes, the distribution of the number of errors is also important. Some flash memory controllers use a weaker decoder when the number of errors is small and switch to a stronger one when the former fails, both for the same code (e.g. bit-flipping and min-sum for decoding an LDPC code [5]). The average read throughput and total power consumption depends on how frequently each decoder is used. Therefore, the distribution of the number of errors, which is also derived here, is a useful tool to find NAND power consumption.

The second half of the paper modifies the proposed algorithm to address the quality of the soft information generated, instead of just the number of errors. In some cases, the BER is too large for a hard decoder to succeed, even if the read is done at the optimal threshold. It is then necessary to generate soft information by performing multiple reads with different read thresholds. The choice of read thresholds has a direct impact on the quality of the soft information generated, which in turn dictates the number of decoder iterations and the number of re-reads required. The paper models the flash as a discrete memoryless channel with mismatched decoding and attempts to maximize its capacity through dynamic programming.

The overall scheme will work as follows. First, the controller will read with an initial threshold and attempt a hard-decoding of the information. If the noise is weak and the initial threshold was well chosen, this decoding will succeed and no further processing will be needed. Otherwise, when this first decoding fails, the controller will perform additional reads with adaptively chosen thresholds to estimate the mean and/or variance of the voltage values for each level. These estimates will in turn be used to estimate the minimum feasible BER and the corresponding optimal read threshold. The flash controller then decides whether to perform an additional read with that estimated threshold to attempt hard decoding again, or directly attempt a more robust decoding of the information, leveraging the reads already performed to generate soft information.

B. Literature review

Most of the existing literature on optimizing the read thresholds for NAND flash assumes that prior information on the noise is available (e.g., [6], [7], [8], [9], [10]). Some methods, such at the one proposed by Wang et al. in [11], assume complete knowledge of the noise and choose the read thresholds so as to maximize the mutual information between the values written and read, while others attempt to predict the noise from the number of program-erase (PE) cycles and then optimize the read thresholds based on that prediction. An example of the latter was proposed by Cai et al. in [12]. Other references addressing threshold selection and error-correction codes are [13] and [14].

However, in some practical cases there is no prior information available, or the prior information is not accurate enough to build a reliable noise model. In these situations, a common approach is to perform several reads with different thresholds searching for the one that returns an equal number of cells on either side, *i.e.*, the median between the two levels[1]. However, the median threshold is suboptimal in general, as was shown in [1]. In [2] and [15] Zhou et al. proposed encoding the data using balanced, asymmetric, or Berger codes to facilitate the threshold selection. Balanced codes guarantee that all codewords have the same number of ones and zeros, hence narrowing the gap between the median and optimal thresholds. Asymmetric and Berger codes, first described in [16], leverage the known asymmetry of the channel to tolerate suboptimal thresholds. Berger codes are able to detect **any** number of unidirectional errors. In cases of significant leakage, where all the cells reduce their voltage level, it is possible to perform several reads with progressively decreasing thresholds until the Berger code detects a low enough number of errors, and only then attempt decoding to recover the host information.

Researchers have also proposed some innovative data representation schemes with different requirements in terms of read thresholds. For example, rank modulation [17], [18], [19], [20], [21] stores information in the relative voltages between the cells instead of using pre-defined voltage levels. The strategy of writing data represented by rank modulation in parallel to flash memories is studied in [22]. Theoretically, rank modulation does not require actual read thresholds, but just comparisons between the cell voltages. Unfortunately, there are a few technological challenges that need to be overcome before rank modulation becomes practical. Other examples include constrained codes [23], [24]; write-once memories codes [25], [26], [25], [27]; and other rewriting codes [28]. All these codes impose restrictions on the levels that can be used during a specific write operation. Since read thresholds need only separate the levels being used, they can often take advantage of these restrictions.

---

[1]In many cases this threshold is not explicitly identified as the median cell voltage, but only implicitly as the solution of $\frac{t-\mu_1}{\sigma_1} = \frac{t-\mu_2}{\sigma_2}$, where (μ_1, σ_1) and (μ_2, σ_2) are the mean and standard deviation of the level voltages.

The scheme proposed in this paper is similar to those described in [29] and [30] in that it assumes no prior information about the noise or data representation, but it is significantly simpler and more efficient. We propose using a small number of reads chosen by a dynamic program to simultaneously estimate the noise and recover the information, instead of periodically testing multiple thresholds (as in [29]) or running a computationally intensive optimization algorithm to perfect the model (as in [30]). A prior version of this paper was published in [31], but the work presented here has been significantly extended with a bound on the capacity for the soft decoding case and a dynamic programming method for optimizing the read thresholds.

II. SYSTEM MODEL

Cells in a NAND flash are organized in terms of pages, which are the smallest units for write and read operations. Writing the cells in a page is done through a program and verify approach where voltage pulses are sent into the cells until their stored voltage exceeds the desired one. Once a cell has reached its desired voltage, it is inhibited from receiving subsequent pulses and the programming of the other cells in the page continues. However, the inhibition mechanism is non-ideal and future pulses may increase the voltage of the cell [12], creating write noise. The other two main sources of noise are inter-cell interference (ICI), caused by interaction between neighboring cells [32], and charges leaking out of the cells with time and heat [33].

Some attempts have been made to model these sources of noise as a function of time, voltage levels, amplitude of the programming pulses, etc. Unfortunately, the noise is temperature- and page-dependent as well as time- and data-dependent [34]. Since the controller cannot measure those factors, it cannot accurately estimate the noise without performing additional reads. This paper assumes that the overall noise follows a Gaussian distribution for each level, as is common in the literature, but assumes no prior knowledge about their means or variances. Section VI will explain how the same idea can be used when the noise is not Gaussian.

Reading the cells in a page is done by comparing their stored voltage with a threshold voltage t. The read operation returns a binary vector with one bit for each cell. Bits corresponding to

cells with voltage lower than t are 1 and those corresponding to cells with voltage higher than t are 0. However, the aforementioned sources of voltage disturbance can cause some cells to be misclassified, introducing errors in the bit values read. The choice of a read threshold therefore becomes important to minimize the BER in the reads.

In a b-bit MLC flash, each cell stores one of 2^b distinct predefined voltage levels. When each cell stores multiple bits, i.e. $b \geq 2$, the mapping of information bits to voltage levels is done using Gray coding to ensure that only one bit changes between adjacent levels. Since errors almost always happen between adjacent levels, Gray coding minimizes the average BER. Furthermore, each of the b bits is assigned to a different page, as shown in Fig. 1. This is done so as to reduce the number of comparisons required to read a page. For example, the lower page of a TLC ($b = 3$) flash can be read by comparing the cell voltages with a single read threshold located between the fourth and fifth levels, denoted by D in Fig. 1. The first four levels encode a bit value 1 for the lower page, while the last four levels encode a value 0. Unfortunately, reading the middle and upper pages require comparing the cell voltages with more read thresholds: two (B,F) for the middle page and four (A,C,E,G) for the upper page.
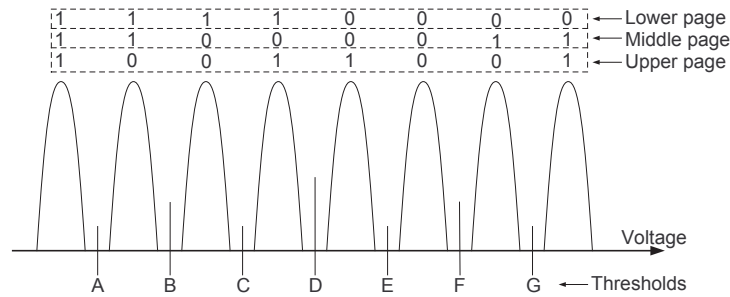

Fig. 1. Typical mapping of bits to pages and voltage levels in a TLC Flash memory.

Upper pages take longer to read than lower ones, but the difference is not as large as it might seem. Flash chips generally incorporate dedicated hardware for performing all the comparisons required to read upper pages, without the additional overhead that would arise from issuing multiple independent read requests. The flash controller can then be oblivious to the type of page

being read. Read commands only need to specify the page being read and a scalar parameter representing the desired shift in the read thresholds from their default value. If the page is a lower one, which employs only one threshold, the scalar parameter is understood as the amount by which this threshold needs to be shifted. If the page is an upper one, which employs multiple read thresholds, their shifts are parameterized by the scalar parameter. For example, a parameter value of Δ when reading the middle page in Fig. 1 could shift thresholds B and F by Δ and $-\frac{3}{2}\Delta$ mV, respectively. Then, cells whose voltage falls between the shifted thresholds B and F would be read as 0 and the rest as 1.

After fixing this parametrization, the flash controller views all the pages in an MLC or TLC memory as independent SLC pages with a single read shift parameter that needs to be optimized. In theory, each low level threshold could be independently optimized, but the large number of reads and amount of memory required would render that approach impractical. Hence, most of the paper will assume a SLC architecture for the flash and Section VI will show how the same method and results can be readily extended to memories with more bits per cell.

Figure 2 (a) shows two overlapping Gaussian probability density functions (pdfs), corresponding to the two voltage levels to which cells can be programmed. Since data is generally compressed before being written onto flash, approximately the same number of cells is programmed to each level. The figure also includes three possible read thresholds. Denoting by (μ_1, σ_1) and (μ_2, σ_2) the means and standard deviations of the two Gaussian distributions, the thresholds are: $t_{\text{mean}} = \frac{\mu_1 + \mu_2}{2}$, $t_{\text{median}} = \frac{\mu_1 \sigma_2 + \mu_2 \sigma_1}{\sigma_1 + \sigma_2}$, and $t^\star$, which minimizes BER. If the noise variance was the same for both levels all three thresholds would be equal, but this is not the case in practice. The plot legend provides the BER obtained when reading with each of the three thresholds.

There exist several ways in which the optimal threshold, $t^\star$, can be found. A common approach is to perform several reads by shifting the thresholds in one direction until the decoding succeeds. Once the data has been recovered, it can be compared with the read outputs to find the threshold yielding the lowest BER [29]. However, this method can require a large number of reads if the initial estimate is inaccurate, which reduces read throughput, and additional memory to store and
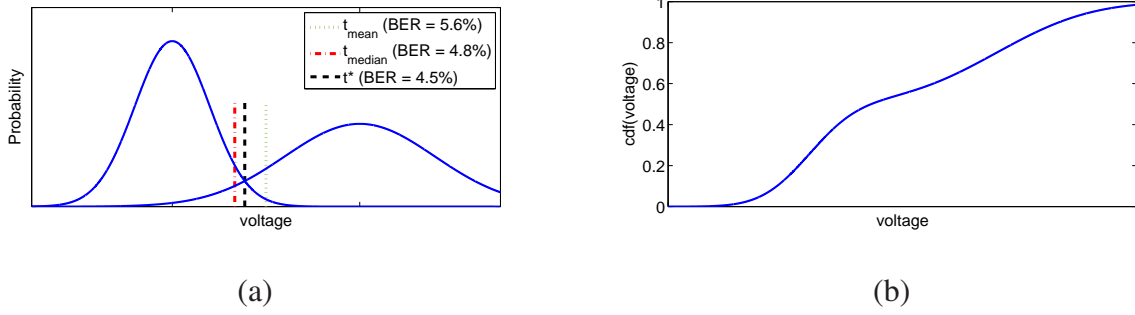
Fig. 2.   (a): Cell voltages pdf in an SLC page, and BER for three different thresholds: $t_{\mathrm{mean}} = (\mu_1 + \mu_2)/2$ is the average of the cell voltages, t_{median} returns the same number of 1s and 0s and $t^\star$ minimizes the BER and is located at the intersection of the two pdfs. (b): cdf corresponding to pdf in (a).

compare the successive reads, which increases cost. The approach taken in this paper consists of estimating (μ_1, σ_1) and (μ_2, σ_2) and deriving $t^\star$ analytically. It will be shown how this can be done with as few reads as possible, thereby reducing read time. Furthermore, the mean and standard deviation estimates can also be used for other tasks, such as generating soft information for LDPC decoding.

A read operation with a threshold voltage t returns a binary vector with a one for each cell whose voltage level is lower than t and zero otherwise. The fraction of ones in the read output is then equal to the probability of a randomly chosen cell having a voltage level below t. Consequently, a read with a threshold voltage t can be used to obtain a sample from the cumulative distribution function (cdf) of cell voltages at t, illustrated in Fig. 2 (b).

The problem is then reduced to estimating the means and variances of a mixture of Gaussians using samples from their joint cdf. These samples will be corrupted by model, read, and quantization noise. Model noise is caused by the deviation of the actual distribution of cell voltages from a Gaussian distribution. Read noise is caused by the intrinsic reading mechanism of the flash, which can read some cells as storing higher or lower voltages than they actually have. Quantization noise is caused by limited computational accuracy and rounding of the Gaussian

cdf[2]. All these sources of noise are collectively referred to as read noise in this paper. It is assumed to be zero mean, but no other restriction is imposed in our derivations.

It is desirable to devote as few reads as possible to the estimation of (μ_1, σ_1) and (μ_2, σ_2). The accuracy of the estimates would improve with the number of reads, but read time would also increase. Since there are four parameters to be estimated, at least four reads will be necessary. Section III describes how the locations of the read thresholds should be chosen in order to achieve accurate estimates and Section IV extends the framework to consider how these reads could be reused to obtain soft information for an LDPC decoder. If the soft information obtained from the first four reads is enough for the LDPC decoding to succeed, no additional reads will be required, thereby reducing the total read time of the flash. Section V proposes a dynamic programming method for optimizing the thresholds for a desired objective. Finally, Section VI explains how to extend the algorithm for MLC or TLC memories, as well as for non-Gaussian noise distributions. Section VII provides simulation results to evaluate the performance of the proposed algorithms and Section VIII concludes the paper.

III. HARD DECODING: MINIMIZING BER

A. Parameter estimation

Let t_i, $i = 1, \ldots, 4$ be four voltage thresholds used for reading a page and let y_i, $i = 1, \ldots, 4$ be the fraction of ones in the output vector for each of the reads, respectively. If (μ_1, σ_1) and (μ_2, σ_2) denote the voltage mean and variance for the cells programmed to the two levels, then

$$y_i = \frac{1}{2}Q\left(\frac{\mu_1 - t_i}{\sigma_1}\right) + \frac{1}{2}Q\left(\frac{\mu_2 - t_i}{\sigma_2}\right) + n_{y_i}, \qquad i = 1, \ldots, 4, \tag{1}$$

where

$$Q(x) = \int_x^\infty (2\pi)^{-\frac{1}{2}} e^{-\frac{t^2}{2}} dt \tag{2}$$

and n_{y_i} denotes the read noise associated to y_i. In theory, it is possible to estimate (μ_1, σ_1) and (μ_2, σ_2) from (t_i, y_i), $i = 1, \ldots, 4$ by solving the system of non-linear equations in Eq. (1),

---

[2]Since the Gaussian cdf has no analytical expression, it is generally quantized and stored as a lookup table

but in practice the computational complexity could be too large for some systems. Another possible approach would be to restrict the estimates to a pre-defined set of values and generate a lookup table for each combination. Finding the table which best fits the samples would require negligible time but the amount of memory required could render this approach impractical for some systems. This section proposes and evaluates a progressive read algorithm that combines these two approaches, providing similar accuracy to the former and requiring only a standard normal ($\mu = 0$, $\sigma = 1$) look-up table.

Progressive Read Algorithm: The key idea is to perform two reads at locations where one of the Q functions is known to be either close to 0 or close to 1. The problem with solving the system in Eq. (1) was that a sum of Q functions cannot be easily inverted. However, once one of the two Q functions is fixed at 0 or 1, the equation can be put in linear form using a standard normal table to invert the other Q function. The system of linear equations can then be solved to estimate the first mean and variance. Once the first mean and variance have been estimated they can be used to evaluate a Q function from each of the two remaining equations in Eq. (1), which can then be solved in a similar way. For example, if t_1 and t_2 are significantly smaller than μ_2, then

$$Q\left(\frac{\mu_2 - t_1}{\sigma_2}\right) \simeq 0 \simeq Q\left(\frac{\mu_2 - t_2}{\sigma_2}\right)$$

and Eq. (1) can be solved for $\widehat{\mu_1}$ and $\widehat{\sigma_1}$ to get

$$\widehat{\sigma_1} = \frac{t_2 - t_1}{Q^{-1}(2y_1) - Q^{-1}(2y_2)} \qquad \widehat{\mu_1} = t_2 + \widehat{\sigma_1}Q^{-1}(2y_2). \tag{3}$$

Substituting these in the equations for the third and fourth reads and solving gives

$$\widehat{\sigma_2} = \frac{t_4 - t_3}{Q^{-1}(2y_3 - q_3) - Q^{-1}(2y_4 - q_4)} \qquad \widehat{\mu_2} = t_4 + \widehat{\sigma_2}Q^{-1}(2y_4 - q_4), \tag{4}$$

where

$$q_3 = Q\left(\frac{\widehat{\mu_1} - t_3}{\widehat{\sigma_1}}\right) \qquad q_4 = Q\left(\frac{\widehat{\mu_1} - t_4}{\widehat{\sigma_1}}\right).$$

It could be argued that, since the pdfs are not known a priori, it is not possible to determine two read locations where one of the Q functions is close to 0 or close to 1. In practice, however,

each read threshold can be chosen based on the result from the previous ones. For example, say the first randomly chosen read location returned $y_1 = 0.6$. This read, if used for estimating the higher level distribution, will be a bad choice because there will be significant overlap from the lower level. Hence, a smart choice would be to obtain two reads for the lower level that are clear of the higher level by reading to the far left of t_1. Once the lower level is canceled, the $y_1 = 0.6$ read can be used in combination with a fourth read to the right of t_1 to estimate the higher level distribution.

Once the mean and variance of both pdfs have been estimated, it is possible to derive an estimate for the read threshold minimizing the BER. The BER associated to a given read threshold t is given by

$$\mathrm{BER}(t) = \frac{1}{2}\left(Q\left(\frac{\mu_2 - t}{\sigma_2}\right) + 1 - Q\left(\frac{\mu_1 - t}{\sigma_1}\right)\right). \tag{5}$$

Making its derivative equal to zero gives the following equation for the optimal threshold $t^\star$

$$\frac{1}{\sigma_2}\phi\left(\frac{\mu_2 - t^\star}{\sigma_2}\right) = \frac{1}{\sigma_1}\phi\left(\frac{\mu_1 - t^\star}{\sigma_1}\right), \tag{6}$$

where $\phi(x) = (2\pi)^{-(1/2)}e^{-x^2/2}$. The optimal threshold $t^\star$ is located at the point where both Gaussian pdfs intersect. An estimate $\widehat{t^\star}$ for $t^\star$ can be found from the following quadratic equation

$$2\log\left(\frac{\widehat{\sigma_2}}{\widehat{\sigma_1}}\right) = \left(\frac{\widehat{t^\star} - \widehat{\mu_1}}{\widehat{\sigma_1}}\right)^2 - \left(\frac{\widehat{t^\star} - \widehat{\mu_2}}{\widehat{\sigma_2}}\right)^2, \tag{7}$$

which can be shown to be equivalent to solving Eq. (6) with (μ_1, σ_1) and (μ_2, σ_2) replaced by their estimated values.

If some parameters are known, the number of reads can be reduced. For example, if μ_1 is known, the first read can be replaced by $t_1 = \mu_1$, $y_1 = 0.25$ in the above equations. Similarly, if σ_1 is known (t_1, y_1) are not required in Eqs. (3)-(4).

B. Error propagation

This subsection first studies how the choice of read locations affects the accuracy of the estimators $(\widehat{\mu_1}, \widehat{\sigma_1})$, $(\widehat{\mu_2}, \widehat{\sigma_2})$, and correspondingly $\widehat{t^\star}$. Then it analyzes how the accuracy of $\widehat{t^\star}$

translates into BER($\widehat{t^\star}$), and provides some guidelines as to how the read locations should be chosen. Without loss of generality, it will be assumed that (μ_1, σ_1) are estimated first using (t_1, y_1) and (t_2, y_2) according to the Progressive Read Algorithm described in Section III-A, and (μ_2, σ_2) are estimated in the second stage. In this case, Eq. (1) reduces to

$$Q\left(\frac{\mu_1 - t_i}{\sigma_1}\right) = 2y_i - 2n_{y_i}$$

for $i = 1, 2$ and the estimates are given by Eqs. (3).

If the read thresholds are on the tails of the distributions, a small perturbation in the cdf value y could cause a significant change in $Q^{-1}(y)$. This will in turn lead to a significant change in the estimates. Specifically, a first-order Taylor expansion of $Q^{-1}(y + n_y)$ at y can be written as

$$Q^{-1}(y + n_y) = x - \sqrt{2\pi}e^{\frac{x^2}{2}}n_y + O(n_y^2), \tag{8}$$

where $x = Q^{-1}(y)$. Since the exponent of e is always positive, the first-order error term is minimized when $x = 0$, i.e., when the read is performed at the mean. The expressions for $(\widehat{\mu_1}, \widehat{\sigma_1})$ and $(\widehat{\mu_2}, \widehat{\sigma_2})$ as seen in Eqs. (3)-(4) use inverse Q functions, so the estimation error due to read noise will be reduced when the reads are done close to the mean of the Gaussian distributions. The first order Taylor expansion of Eq. (3) at σ_1 is given by

$$\widehat{\sigma_1} = \sigma_1 - \frac{\sigma_1^2}{t_2 - t_1}(n_2 - n_1) + O(n_1^2, n_2^2) \tag{9}$$

where

$$n_1 = 2\sqrt{2\pi}e^{\frac{(t_1 - \mu_1)^2}{2\sigma_1^2}}n_{y_1} + O(n_{y_1}^2) \qquad n_2 = 2\sqrt{2\pi}e^{\frac{(t_2 - \mu_1)^2}{2\sigma_1^2}}n_{y_2} + O(n_{y_2}^2). \tag{10}$$

A similar expansion can be performed for $\widehat{\mu_1}$, obtaining

$$\widehat{\mu_1} = \mu_1 - \sigma_1 \frac{(t_2 - \mu_1)n_1 - (t_1 - \mu_1)n_2}{t_2 - t_1} + O(n_1^2, n_2^2). \tag{11}$$

Two different tendencies can be observed in the above expressions. On one hand, Eqs. (10) suggest that both t_1 and t_2 should be chosen close to μ_1 so as to reduce the magnitude of n_1 and n_2. On the other hand, if t_1 and t_2 are very close together, the denominators in Eq. (9) and (11) can become small, increasing the estimation error.

The error expansions for $\widehat{\mu_2}$, $\widehat{\sigma_2}$ and $\widehat{t^\star}$, are omitted for simplicity, but it can be shown that the dominant terms are linear in n_{y_i}, $i = 1, \ldots, 4$ as long as all n_{y_i} are small enough. The Taylor expansion for $\mathrm{BER}(\widehat{t^\star})$ at $t^\star$ is

$$\mathrm{BER}(\widehat{t^\star}) = \mathrm{BER}(t^\star) + \left(\frac{1}{2\sigma_2} \phi \left(\frac{\mu_2 - t^\star}{\sigma_2} \right) - \frac{1}{2\sigma_1} \phi \left(\frac{\mu_1 - t^\star}{\sigma_1} \right) \right) e_{t^\star} + O(e_{t^\star}^2)$$

$$= \mathrm{BER}(t^\star) + O(e_{t^\star}^2), \tag{12}$$

where $\widehat{t^\star} = t^\star + e_{t^\star}$. The cancellation of the first-order term is justified by Eq. (6). Summarizing, the mean and variance estimation error increases linearly with the read noise, as does the deviation in the estimated optimal read threshold. The increase in BER, on the other hand, is free from linear terms. As long as the read noise is not too large, the resulting $\mathrm{BER}(\widehat{t^\star})$ is close to the minimum possible BER. The numerical simulations in Fig. 3 confirm these results.
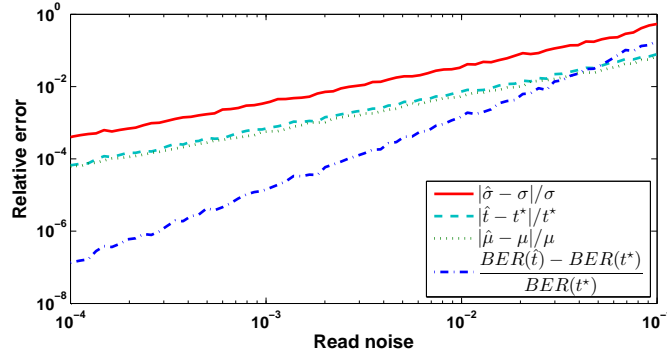

Fig. 3. The relative error in the mean, variance, and threshold estimates increases linearly with the read noise (slope=1), but the relative increase in BER grows quadratically (slope=2) and is negligible for a wide range of read noise amplitudes.

In view of these results, it seems that the read thresholds should be spread out over both pdfs but close to the levels' mean voltages. Choosing the thresholds in this way will reduce the error propagating from the reads to the estimates. However, read thresholds can be chosen sequentially, using the information obtained from each read in selecting subsequent thresholds. Section V proposes a method for finding the optimal read thresholds more precisely.

IV. SOFT DECODING: TRADEOFF BER-LLR

This section considers a new scenario where a layered decoding approach is used for increased error-correction capability. After reading a page, the controller may first attempt to correct any bit errors in the read-back codeword using a hard decoder alone, typically a bit-flipping hard-LDPC decoder [35]. Reading with the threshold $\widehat{t}^\star$ found through Eq. (7) reduces the number of hard errors but there are cases in which even $\text{BER}(\widehat{t}^\star)$ is too high for the hard decoder to succeed. When this happens, the controller will attempt a soft decoding, typically using a min-sum or sum-product soft LDPC decoder.

Soft decoders are more powerful, but also significantly slower and less power efficient than hard decoders. Consequently, invoking soft LDPC decoding too often can significantly impact the controller's average read time. In order to estimate the probability of requiring soft decoding, one must look at the distribution of the number of errors, and not at BER alone. For example, if the number of errors per codeword is uniformly distributed between 40 and 60 and the hard decoder can correct 75 errors, soft decoding will never be needed. However, if the number of errors is uniformly distributed between 0 and 100 (same BER), soft decoding will be required to decode 25% of the reads. Section IV-A addresses this topic.

The error-correction capability of a soft decoder depends heavily on the quality of the soft information at its input. It is always possible to increase such quality by performing additional reads, but this decreases read throughput. Section IV-B shows how the Progressive Read Algorithm from the previous section can be modified to provide high quality soft information.

A. Distribution of the number of errors

Let N be the number of bits in a codeword. Assuming that both levels are equally likely, the probability of error for any given bit, denoted p_e, is given in Eq. (5). Errors can be considered independent, hence the number of them in a codeword follows a binomial distribution with parameters N and p_e. Since N is usually large, it becomes convenient to approximate the binomial

Failure rate	$p_e = 0.008$	$p_e = 0.01$	$p_e = 0.012$
$\alpha = 23$	0.05	0.28	0.62
$\alpha = 25$	0.016	0.15	0.46
$\alpha = 27$	0.004	0.07	0.31

TABLE I

FAILURE RATE FOR A $N = 2048$ BCH CODE AS A FUNCTION OF PROBABILITY OF BIT ERROR p_e AND NUMBER OF CORRECTABLE BIT ERRORS α.

by a Gaussian distribution with mean Np_e and variance $Np_e(1-p_e)$, or by a Poisson distribution with parameter Np_e when Np_e is small.

Under the Gaussian approximation paradigm, a codeword fails to decode with probability $Q\left(\frac{\alpha - Np_e}{\sqrt{Np_e(1-p_e)}}\right)$, where α denotes the number of bit errors that can be corrected. Table I shows that a small change in the value of α may increase significantly the frequency with which a stronger decoder is needed. This has a direct impact on average power consumption of the controller. The distribution of bit errors can thus be used to judiciously obtain a value of α in order to meet a power constraint.

B. Obtaining soft inputs

After performing M reads on a page, each cell can be classified as falling into one of the $M + 1$ intervals between the read thresholds. The problem of reliably storing information on the flash is therefore equivalent to the problem of reliable transmission over a discrete memoryless channel (DMC), such as the one in Fig. 4. Channel inputs represent the levels to which the cells are written, outputs represent read intervals, and channel transition probabilities specify how likely it is for cells programmed to a specific level to be found in each interval at read time.

It is well known that the capacity of a channel is given by the maximum mutual information between the input and output over all input distributions (codebooks) [36]. In practice, however, the code must be chosen at write time when the channel is still unknown, making it impossible to adapt the input distribution to the channel. Although some asymmetric codes have been
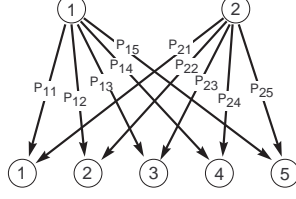
Fig. 4. DMC channel equivalent to Flash read channel with four reads.

proposed (e.g. [15], [24], [37]), channel inputs are equiprobable for most practical codes. The mutual information between the input and the output is then given by

$$I(X;Y) = \frac{1}{2} \sum_{j=1}^{M+1} p_{1j} \log(p_{1j}) + p_{2j} \log(p_{2j}) - (p_{1j} + p_{2j}) \log \left(\frac{p_{1j} + p_{2j}}{2} \right), \tag{13}$$

where p_{ij}, $i = 1, 2$, $j = 1, \ldots, M+1$ are the channel transition probabilities. For Gaussian noise, these transition probabilities can be found as

$$p_{ij} = Q \left(\frac{\mu_i - t_j}{\sigma_i} \right) - Q \left(\frac{\mu_i - t_{j-1}}{\sigma_i} \right), \tag{14}$$

where $t_0 = -\infty$ and $t_{M+1} = \infty$.

The inputs to a soft decoder are given in the form of log-likelihood ratios (LLR). The LLR value associated with a read interval k is defined as $LLR_k = \log(p_{1k}/p_{2k})$. When the mean and variance are known it is possible to obtain good LLR values by reading at the locations that maximize $I(X;Y)$ [11], which tend to be on the so-called uncertainty region, where both pdfs are comparable. However, the mean and variance are generally not known and need to be estimated. Section III provided some guidelines on how to choose read thresholds in order to obtain accurate estimates, but those reads tend to produce poor LLR values. Hence, there are two opposing trends: spreading out the reads over a wide range of voltage values yields more accurate mean and variance estimates but degrades the performance of the soft decoder, while concentrating the reads on the uncertainty region provides better LLR values but might yield inaccurate estimates which in turn undermine the soft decoding.

Some flash manufacturers are already incorporating soft read commands that return 3 or 4 bits of information for each cell, but the thresholds for those reads are often pre-specified and kept constant throughout the lifetime of the device. Furthermore, most controller manufacturers use a pre-defined mapping of read intervals to LLR values regardless of the result of the reads. We propose adjusting the read thresholds and LLR values adaptively to fit our channel estimates.

Our goal is to find the read locations that maximize the probability of successful decoding when levels are equiprobable and the decoding is done based on the estimated transition probabilities. With this goal in mind, Section IV-C will derive a bound for the (symmetric and mismatched) channel capacity in this scenario and Section V will show how to choose the read thresholds so as to maximize this bound. The error-free coding rate specified by the bound will not be achievable in practice due to finite code length, limited computational power, etc., but the BER at the output of a decoder is closely related to the capacity of the channel [38], [39]. The read thresholds that maximize the capacity of the channel are generally the same ones that minimize the BER, in practice.

C. Bound for maximum transmission rate

Shannon's channel coding theorem states that all transmission rates below the channel capacity are achievable when the channel is perfectly known to the decoder; unfortunately this is not the case in practice. The channel transition probabilities can be estimated by substituting the noise means and variances $\widehat{\mu_1}, \widehat{\mu_2}, \widehat{\sigma_1}, \widehat{\sigma_2}$ into Eq. (14) but these estimates, denoted $\widehat{p_{ij}}$, $i = 1, 2$, $j = 1, \ldots, 5$, are inaccurate. The decoder is therefore not perfectly matched to the channel.

The subject of mismatched decoding has been of interest since the 1970's. The most notable early works are by Hui [40] and Csiszár and Körner [41], who provided bounds on the maximum transmission rates under several different conditions. Merhav et al. [42] related those results to the concept of generalized mutual information and, more recently, Scarlett et al. [39] found bounds and error exponents for the finite code length case. It is beyond the scope of this paper to perform a detailed analysis of the mismatched capacity of a DMC channel with symmetric

inputs; the interested reader can refer to the above references as well as [43], [44], [45], and [46]. Instead, we will derive a simplified lower bound for the capacity of this channel in the same scenario that has been considered throughout the paper.

Theorem 1. *The maximum achievable rate of transmission with vanishing probability of error over a Discrete Memoryless Channel with equiprobable binary inputs, output alphabet $\mathcal{Y}$, transition probabilities p_{ij}, $i = 1, 2$, $j = 1, \ldots, |\mathcal{Y}|$, and maximum likelihood decoding according to a different set of transition probabilities $\widehat{p_{ij}}$, $i = 1, 2$, $j = 1, \ldots, |\mathcal{Y}|$ is lower bounded by*

$$C_{P,\hat{P}} = \frac{1}{2} \sum_{j=1}^{|\mathcal{Y}|} p_{1j} \log(\widehat{p_{1j}}) + p_{2j} \log(\widehat{p_{2j}}) - (p_{1j} + p_{2j}) \log\left(\frac{\widehat{p_{1j}} + \widehat{p_{2j}}}{2}\right) \tag{15}$$

Proof: Provided in the Appendix. ∎

It is worth noting that $C_{P,\hat{P}}$ is equal to the mutual information given in Eq. (13) when the estimates are exact, and decreases as the estimates become less accurate. In fact, the probability of reading a given value $y \in \mathcal{Y}$ can be measured directly as the fraction of cells mapped to the corresponding interval, so it is usually the case that $\widehat{p_{1k}} + \widehat{p_{2k}} = p_{1k} + p_{2k}$. The bound then becomes $C_{P,\hat{P}} = I(X;Y) - D(P||\hat{P})$, where $I(X;Y)$ is the symmetric capacity of the channel with matched ML decoding and $D(P||\hat{P})$ is the relative entropy (also known as Kullback-Leibler distance) between the exact and the estimated transition probabilities.

$$D(P||\hat{P}) = \frac{1}{2} \sum_{j=1}^{|\mathcal{Y}|} p_{1j} \log\left(\frac{p_{1j}}{\widehat{p_{1j}}}\right) + p_{2j} \log\left(\frac{p_{2j}}{\widehat{p_{2j}}}\right). \tag{16}$$

In this case $C_{P,\hat{P}}$ is a concave function of the transition probabilities $(p_{ij}, \widehat{p_{ij}})$, $i = 1, 2$, $j = 1, \ldots, |\mathcal{Y}|$, since the relative entropy is convex and the mutual information is concave [36]. The bound attains its maximum when the decoder is matched to the channel (i.e. $p_{ij} = \widehat{p_{ij}} \quad \forall i, j$) and the read thresholds are chosen so as to maximize the mutual information between X and Y, but that solution is not feasible for our problem.

In practice, both the capacity of the underlying channel and the accuracy of the estimates at the decoder depend on the location of the read thresholds and cannot be maximized simultaneously. Finding the read thresholds t_1, t_2, t_3, and t_4 which maximize $C_{P,\hat{P}}$ is not straightforward, but it

can be done numerically. Section V describes a dynamic programming algorithm for choosing each read threshold based on prior information about the noise and the result of previous reads.

V. OPTIMIZING READ THRESHOLDS

In most practical cases, the flash controller has prior information about the voltage distributions, based on the number of PE cycles that the page has endured, its position within the block, etc. This prior information is generally not enough to produce accurate noise estimates, but it can be used to improve the choice of read thresholds. We wish to determine a policy to choose the optimal read thresholds sequentially, given the prior information about the voltage distributions and the results in previous reads.

This section proposes a dynamic programming framework to find the read thresholds that maximize the expected value of a user-defined reward function. If the goal is to minimize the BER at the estimated threshold $\widehat{t}^\star$, as in Section III, an appropriate reward would be $1 - BER(\widehat{t}^\star)$. If the goal is to maximize the channel capacity, the reward could be chosen to be $I(X;Y) - D(P\|\hat{P})$, as shown in Section IV-C.

Let $\mathbf{x} = (\mu_1, \mu_2, \sigma_1, \sigma_2)$ and $\mathbf{r_i} = (t_i, y_i)$, $i = 1, \ldots, 4$ be vector random variables, so as to simplify the notation. If the read noise distribution f_n is known, the prior distribution for $\mathbf{x}$ can be updated based on the result of each read $\mathbf{r_i}$ using Bayes rule and Eq. (1):

$$
\begin{aligned}
f_{\mathbf{x}|\mathbf{r_1},\ldots,\mathbf{r_i}} &= K \cdot f_{y_i|\mathbf{x},t_i} \cdot f_{\mathbf{x}|\mathbf{r_1},\ldots,\mathbf{r_{i-1}}} \\
&= K \cdot f_n \left(y_i - \frac{1}{2} \left(Q \left(\frac{\mu_1 - t_i}{\sigma_1} \right) + Q \left(\frac{\mu_2 - t_i}{\sigma_2} \right) \right) \right) \cdot f_{\mathbf{x}|\mathbf{r_1},\ldots,\mathbf{r_{i-1}}},
\end{aligned}
\tag{17}
$$

where K is a normalization constant. Furthermore, let $R(\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_3}, \mathbf{r_4})$ denote the expected reward associated with the reads $\mathbf{r_1}, \ldots, \mathbf{r_4}$, after updating the prior $f_{\mathbf{x}}$ accordingly. In the following, we will use R to denote this function, omitting the arguments for the sake of simplicity.

Choosing the fourth read threshold t_4 after the first three reads $\mathbf{r_1}, \ldots, \mathbf{r_3}$ is relatively straightforward: t_4 should be chosen so as to maximize the expected reward, given the results of the

previous three reads. Formally,

$$t_4^\star = \arg\max_{t_4} E\left\{R|\mathbf{r_1}, \ldots, \mathbf{r_3}, t_4\right\}, \tag{18}$$

where the expectation is taken with respect to $(y_4, \mathbf{x})$ by factoring their joint distribution in a similar way to Eq. (17): $f_{y_4, \mathbf{x}|\mathbf{r_1}, \ldots, \mathbf{r_3}} = f_{y_4|\mathbf{x}, t_4} \cdot f_{\mathbf{x}|\mathbf{r_1}, \ldots, \mathbf{r_3}}$.

This defines a policy π for the fourth read, and a value V_3 for each possible state after the first three reads:

$$\pi_4(\mathbf{r_1}, \ldots, \mathbf{r_3}) = t_4^\star \tag{19}$$

$$V_3(\mathbf{r_1}, \ldots, \mathbf{r_3}) = E\left\{R|\mathbf{r_1}, \ldots, \mathbf{r_3}, t_4^\star\right\}. \tag{20}$$

In practice, the read thresholds t_i and samples y_i can only take a finite number of values, hence the number of feasible arguments in these functions (states) is also finite. This number can be fairly large, but it is only necessary to find the value for a small number of them, those which have non-negligible probability according to the prior $f_\mathbf{x}$ and value significantly larger than 0. For example, states are invariant to permutation of the reads so they can always be reordered such that $t_1 < t_2 < t_3$. Then, states which do not fulfill $y_1 < y_2 < y_3$ can be ignored. If the number of states after discarding meaningless ones is still too large, it is also possible to use approximations for the policy and value functions [47], [48].

Equations (19) and (20) assign a value and a fourth read threshold to each meaningful state after three reads. The same idea, using a backward recursion, can be used to decide the third read threshold and assign a value to each state after two reads:

$$\pi_3(\mathbf{r_1}, \mathbf{r_2}) = \arg\max_{t_3} E\left\{V_3(\mathbf{r_1}, \ldots, \mathbf{r_3})|\mathbf{r_1}, \mathbf{r_2}, t_3\right\} \tag{21}$$

$$V_2(\mathbf{r_1}, \mathbf{r_2}) = \max_{t_3} E\left\{V_3(\mathbf{r_1}, \ldots, \mathbf{r_3})|\mathbf{r_1}, \mathbf{r_2}, t_3\right\}, \tag{22}$$

where the expectation is taken with respect to $(y_3, \mathbf{x})$. Similarly, for the second read threshold

$$\pi_2(\mathbf{r_1}) = \arg\max_{t_2} E\left\{V_2(\mathbf{r_1}, \mathbf{r_2})|\mathbf{r_1}, t_2\right\} \tag{23}$$

$$V_1(\mathbf{r_1}) = \max_{t_2} E\left\{V_2(\mathbf{r_1}, \mathbf{r_2})|\mathbf{r_1}, t_2\right\}, \tag{24}$$

where the expectation is taken with respect to $(y_2, \mathbf{x})$. Finally, the optimal value for the first read threshold is

$$t_1^\star = \arg \max_{t_1} E \left\{ V_1(t_1, y_1) | t_1 \right\}.$$

These policies can be computed offline and then programmed in the memory controller. Typical controllers have multiple modes tailored towards different conditions in terms of number of PE cycles, whether an upper or lower page is being read, etc. Each of these modes would have its own prior distributions for $(\mu_1, \mu_2, \sigma_1, \sigma_2)$, and would result in a different policy determining where to perform each read based on the previous results. Each policy can be stored as a partition of the feasible reads, and value functions can be discarded, so memory requirements are very reasonable. Section VII presents an example illustrating this scheme.

Just like in Section III-A, the number of reads can be reduced if some of the noise parameters are known or enough prior information is available. The same backward recursion could be used to optimize the choice of thresholds, but with fewer steps.

VI. EXTENSIONS

Most of the paper has assumed that cells can only store two voltage levels, with their voltages following Gaussian distributions. This framework was chosen because it is the most widely used in the literature, but the method described can easily be extended to memories with more than two levels and non-Gaussian noise distributions.

Section II explained how each wordline in a MLC (two bits per cell, four levels) or TLC (three bits per cell, eight levels) memory is usually divided into two or three pages which are read independently as if the memory was SLC. In that case, the proposed method can be applied without any modifications. However, if the controller is capable of simultaneously processing more than two levels per cell, it is possible to accelerate the noise estimation by reducing the number of reads. MLC and TLC memories generally have dedicated hardware that performs multiple reads in the ranges required to read the upper pages and returns a single binary value. For example, reading the upper page of a TLC memory with the structure illustrated in

Fig. 1 requires four reads with thresholds (A, C, E, G) but cells between A and C would be indistinguishable from cells between E and G; all of them would be read as 0. However, one additional read of the lower page (D threshold) would allow the controller to tell them apart.

Performing four reads $(t_1, \ldots, t_4)$ on the upper page of a TLC memory would entail comparing the cell voltages against 16 different thresholds but obtaining only four bits of information for each cell. The means and variances in Eqs. (3)-(4) would correspond to mixtures of all the levels storing the same bit value, assumed to be approximately Gaussian. The same process would then be repeated for the middle and lower page. A better approach, albeit more computationally intensive, would be to combine reads from all three pages and estimate each level independently. Performing one single read of the lower page (threshold D), two of the middle page (each involving two comparisons, with thresholds B and F) and three of the upper page (each involving four comparisons, with thresholds A, C, E, G) would theoretically provide more than enough data to estimate the noise in all eight Gaussian levels. A similar process can be used for MLC memories performing, for example, two reads of the lower page and three of the upper page.

Hence, five page reads are enough to estimate the noise mean and variance in all 4 levels of an MLC memory and 6 page reads are enough for the 8 levels in a TLC memory. Other choices for the pages to be read are also possible, but it is useful to consider that lower pages have smaller probabilities of error, so they often can be successfully decoded with fewer reads. Additional reads could provide more precise estimates and better LLR values for LDPC decoding.

There are papers suggesting that a Gaussian noise model might not be accurate for some memories [49]. The proposed scheme can also be extended to other noise distributions, as long as they can be characterized by a small number of parameters. Instead of the Q-function in Eq. (2), the estimation should use the cumulative density function (cdf) for the corresponding noise distribution. For example, if the voltage distributions followed a Laplace instead of Gaussian distribution, Eq. (1) would become

$$y_i = \frac{1}{2} - \frac{1}{4}e^{-\frac{t_i - \mu_1}{b_1}} + \frac{1}{2}e^{-\frac{t_i - \mu_2}{b_2}} + n_{y_i}, \qquad (25)$$

for $\mu_1 \leq t_i \leq \mu_2$ and the estimator $\widehat{b_1}$ of b_1 would become

$$\widehat{b_1} = \frac{t_2 - t_1}{\log(1 - 2y_1) - \log(1 - 2y_2)} \tag{26}$$

when t_1, t_2 are significantly smaller than μ_2. Similar formulas can be found to estimate the other parameters.

VII. NUMERICAL RESULTS

This section presents simulation results evaluating the performance of the dynamic programming algorithm proposed in Section V. Two scenarios will be considered, corresponding to a fresh page with $\text{BER}(t^\star) = 0.0015$ and a worn-out page with $\text{BER}(t^\star) = 0.025$. The mean voltage values for each level will be the same in both scenarios, but the standard deviations will differ. Specifically, $\mu_1 = 1$ and $\mu_2 = 2$ for both pages, but the fresh page will be modeled using $\sigma_1 = 0.12$ and $\sigma_2 = 0.22$, while the worn page will be modeled using $\sigma_1 = 0.18$ and $\sigma_2 = 0.32$. These values, however, are unknown to the controller. The only information that it can use to choose the read locations are uniform prior distributions on μ_1, μ_2, σ_1, and σ_2, identical for both the fresh and the worn-out pages. Specifically, μ_1 is known to be in the interval $(0.75, 1.25)$, μ_2 in $(1.8, 2.1)$, σ_1 in $(0.1, 0.24)$ and σ_2 in $(0.2, 0.36)$.

For each scenario, three different strategies for selecting the read thresholds were evaluated. The first strategy, S_1, tries to obtain accurate noise estimates by spreading out the reads. The second strategy, S_2, concentrates all of them on the uncertainty region, attempting to attain highly informative LLR values. Finally, the third strategy, S_3, follows the optimal policy obtained by the dynamic programming recursion proposed in Section V, with $C_{P,\hat{P}}$ as reward function. The three strategies are illustrated in Fig. 5 and the results are summarized in Table II, but before proceeding to their analysis we describe the process employed to obtain S_3.

The dynamic programming scheme assumed that read thresholds were restricted to move in steps of 0.04, and quantized all cdf measurements also in steps of 0.04 (making the noise n_y from Eq. (1) uniform between -0.02 and 0.02). Starting from these assumptions, Eqs. (19) and (20)

were used to find the optimal policy π_4 and expected value V_3 for all meaningful combinations of $(t_1, y_1, t_2, y_2, t_3, y_3)$, which were in the order of 10^6 (very reasonable for offline computations). The value function V_3 was then used in the backward recursion to find the policies and values for the first three reads as explained in Section V. The optimal location for the first read, in terms of maximum expected value for $I(X; Y) - D(P \| \hat{P})$ after all four reads, was found to be $t_1^\star = 1.07$. This read resulted in $y_1 = 0.36$ for the fresh page and $y_1 = 0.33$ for the worn page. The policy π_2 dictated that $t_2 = 0.83$ for $y_1 \in (0.34, 0.38)$, and $t_2 = 1.63$ for $y_1 \in (0.3, 0.34)$, so those were the next reads in each case. The third and fourth read thresholds t_3 and t_4 were chosen similarly according to the corresponding policies.

Finally, as depicted in Fig. 5, the read thresholds were

- S_1: $\mathbf{t} = (0.85,\ 1.15,\ 1.75,\ 2.125)$.

- S_2: $\mathbf{t} = (1.2,\ 1.35,\ 1.45,\ 1.6)$.

- S_3 (fresh page): $\mathbf{t} = (1.07,\ 0.83,\ 1.79,\ 1.31)$ resulting in $\mathbf{y} = (0.36,\ 0.04,\ 0.58,\ 0.496)$, respectively.

- S_3 (worn page): $\mathbf{t} = (1.07,\ 1.63,\ 1.19,\ 1.43)$ resulting in $\mathbf{y} = (0.33,\ 0.56,\ 0.43,\ 0.51)$, respectively.

For the fresh page, the policy dictates that the first three reads should be performed well outside of the uncertainty region, so as to obtain good estimates of the means and variances. Then, the fourth read is performed as close as possible to the BER-minimizing threshold. Since the overlap between both levels is very small, soft decoding would barely provide any gain over hard decoding. Picking the first three reads for noise characterization regardless of their value towards building LLRs seems indeed to be the best strategy. For the worn-out page, the policy attempts to achieve a trade-off by combining two reads away from the uncertainty region, good for parameter estimation, with another two inside it to improve the quality of the LLR values used for soft decoding.

Table II shows the relative error in our estimates and sector failure rates averaged over 5000 simulation instances, with read noise n_{y_i}, $i = 1, \ldots, 4$ uniformly distributed between -0.02 and
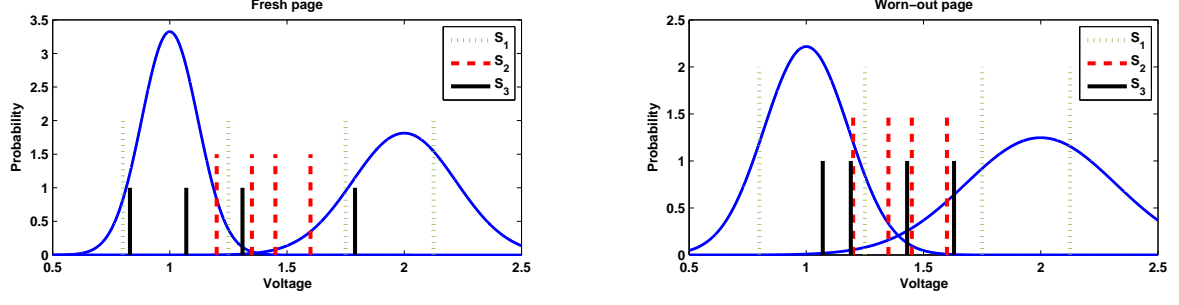
Fig. 5.   Read thresholds for strategies S_1, S_2 and S_3 for a fresh and a worn-out page.

FRESH PAGE	S_1	S_2	S_3
$|\hat{\mu} - \mu|/\mu$	0.004	0.182	0.012
$|\hat{\sigma} - \sigma|/\sigma$	0.03	0.91	0.12
$|\widehat{t^\star} - t^\star|/t^\star$	0.01	0.07	0.02
$|\mathrm{BER}(\widehat{t^\star}) - \mathrm{BER}(t^\star)|/\mathrm{BER}(t^\star)$	0.1	1.4	0.11
LDPC fail rate	1	0.15	0
Genie LDPC fail rate	1	0	0

OLD PAGE	S_1	S_2	S_3
$|\hat{\mu} - \mu|/\mu$	0.005	0.053	0.021
$|\hat{\sigma} - \sigma|/\sigma$	0.03	0.27	0.13
$|\widehat{t^\star} - t^\star|/t^\star$	0.006	0.015	0.011
$|\mathrm{BER}(\widehat{t^\star}) - \mathrm{BER}(t^\star)|/\mathrm{BER}(t^\star)$	0.003	0.009	0.007
LDPC fail rate	1	0.19	0.05
Genie LDPC fail rate	1	0	0.01

TABLE II

TRADE-OFF BETWEEN BER AND LDPC FAILURE RATE.

0.02. The first three rows show the relative estimation error of the mean, variance, and optimal threshold. It can be observed that S_1 provides the lowest estimation error, while S_2 produces clearly wrong estimates. The estimates provided by S_3 are noisier than those provided by S_1, but are still acceptable. The relative increase in BER when reading at $\widehat{t^\star}$ instead of at $t^\star$ is shown in the fourth row of each table. It is worth noting that the $\mathrm{BER}(\widehat{t^\star})$ does not increase significantly, even with inaccurate mean and variance estimates. This validates the derivation in Section III-B.

Finally, the last two rows on each table show the failure rate after 20 iterations of a min-sum LDPC decoder for two different methods of obtaining soft information. The LDPC code had 18% redundancy and codeword length equal to 35072 bits. The fifth row corresponds to LLR values obtained using the mean and variance estimates from the Progressive Read Algorithm and the last row, labeled "Genie LDPC", corresponds to using the actual values instead of the estimated ones. It can be observed that strategy S_1, which provided very accurate estimates, always fails in the LDPC decoding. This is due to the wide range of cell voltages that fall

between the middle two reads, being assigned an LLR value close to 0. The fact that the "Genie LDPC" performs better with S_2 than with S_3 shows that the read locations chosen by the former are better. However, S_3 provides lower failure rates in the more realistic case where the means and variances need to be estimated using the same reads used to produce the soft information.

In summary, S_3 was found to be best from an LDPC code point of view and S_1 from a pure BER-minimizing perspective. S_2 as proposed in [11] is worse in both cases unless the voltage distributions are known. When more than four reads are allowed, all three schemes perform similarly. After the first four reads, all the strategies have relatively good estimates for the optimal threshold. Subsequent reads are located close to the optimal threshold, achieving small BER. Decoding failure rates are then limited by the channel capacity, rather than by the location of the reads.

VIII. CONCLUSION

NAND flash controllers often require several re-reads using different read thresholds to recover host data in the presence of noise. In most cases, the controller tries to guess the noise distribution based on the number of PE cycles and picks the read thresholds based on that guess. However, unexpected events such as excessive leakage or charge trapping can make those thresholds suboptimal. This paper proposed algorithms to reduce the total read time and sector failure rate by using a limited number of re-reads to estimate the noise and improve the read thresholds.

The overall scheme will work as follows. First, the controller will generally have a prior estimation of what a good read threshold might be. It will read at that threshold and attempt a hard-decoding of the information. If the noise is weak and the initial threshold was well chosen, this decoding will succeed and no further processing will be needed. In cases when this first decoding fails, the controller will perform additional reads to estimate the mean and/or variance of the voltage values for each level. These estimates will in turn be used to estimate the minimum achievable BER and the corresponding optimal read threshold. The flash controller then decides whether to perform an additional read with this threshold to attempt hard decoding again, or

directly attempt a more robust decoding of the information, for example LDPC, leveraging the reads already performed to generate the soft information.

The paper proposes using a dynamic programming backward recursion to find a policy for progressively picking the read thresholds based on the prior information available and the results from previous reads. This scheme will allow us to find the thresholds that optimize an arbitrary objective. Controllers using hard decoding only (e.g., BCH) may wish to find the read threshold providing minimum BER, while those employing soft decoding (e.g., LDPC) will prefer to maximize the capacity of the resulting channel. The paper provides an approximation for the (symmetric and mismatched) capacity of the channel and presents simulations to illustrate the performance of the proposed scheme in such scenarios.

References

[1] B. Peleato and R. Agarwal, "Maximizing MLC NAND lifetime and reliability in the presence of write noise," in IEEE Int. Conf. on Communications (ICC). IEEE, 2012, pp. 3752–3756.

[2] H. Zhou, A. Jiang, and J. Bruck, "Error-correcting schemes with dynamic thresholds in nonvolatile memories," in IEEE Int. Symp. on Information Theory (ISIT). IEEE, 2011, pp. 2143–2147.

[3] B. Peleato, R. Agarwal, and J. Cioffi, "Probabilistic graphical model for flash memory programming," in IEEE Statistical Signal Processing Workshop (SSP). IEEE, 2012, pp. 788–791.

[4] M. Asadi, X. Huang, A. Kavcic, and N. P. Santhanam, "Optimal detector for multilevel NAND flash memory channels with intercell interference," IEEE J. Sel. Areas Commun., vol. 32, no. 5, pp. 825–835, May 2014.

[5] M. Anholt, N. Sommer, R. Dar, U. Perlmutter, and T. Inbar, "Dual ECC decoder," Apr. 23 2013, US Patent 8,429,498.

[6] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error-correction codes in NAND flash memory," IEEE Trans. Circuits Syst. I: Reg. Papers, vol. 58, no. 2, pp. 429–439, Nov. 2011.

[7] F. Sala, R. Gabrys, and L. Dolecek, "Dynamic threshold schemes for multi-level non-volatile memories," IEEE Trans. Commun., vol. 61, no. 7, pp. 2624–2634, Jul. 2013.

[8] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Error patterns in MLC NAND flash memory: Measurement, characterization, and analysis," in Proc. Conf. Design, Automation and Test in Europe. IEEE, Mar. 2012, pp. 521–526.

[9] Q. Li, A. Jiang, and E. F. Haratsch, "Noise modeling and capacity analysis for NAND flash memories," in IEEE Int. Symp. on Information Theory (ISIT). IEEE, Jul. 2014, pp. 2262–2266.

[10] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Threshold voltage distribution in MLC NAND flash memory: Characterization, analysis, and modeling," in Proc. Conf. Design, Automation and Test in Europe. EDA Consortium, Mar. 2013, pp. 1285–1290.

[11] J. Wang, T. Courtade, H. Shankar, and R. Wesel, "Soft information for LDPC decoding in flash: Mutual information optimized quantization," in IEEE Global Communications Conf. (GLOBECOM), 2011, pp. 5–9.

[12] Y. Cai, O. Mutlu, E. F. Haratsch, and K. Mai, "Program interference in MLC NAND flash memory: Characterization, modeling, and mitigation," in IEEE Int. Conf. on Computer Design (ICCD), 2013, pp. 123–130.

[13] R. Gabrys, F. Sala, and L. Dolecek, "Coding for unreliable flash memory cells," IEEE Commun. Lett., vol. 18, no. 9, pp. 1491–1494, Jul. 2014.

[14] R. Gabrys, E. Yaakobi, and L. Dolecek, "Graded bit-error-correcting codes with applications to flash memory," IEEE Trans. Inf. Theory, vol. 59, no. 4, pp. 2315–2327, Apr. 2013.

[15] H. Zhou, A. Jiang, and J. Bruck, "Non-uniform codes for asymmetric errors," in IEEE Int. Symp. on Information Theory (ISIT).   IEEE, 2011.

[16] J. Berger, "A note on error detection codes for asymmetric channels," Inform. and Control, vol. 4, no. 1, pp. 68–73, 1961.

[17] A. Jiang, M. Schwartz, and J. Bruck, "Error-correcting codes for rank modulation," in IEEE Int. Symp. on Information Theory (ISIT).   IEEE, 2008, pp. 1736–1740.

[18] A. Jiang, R. Mateescu, M. Schwartz, and J. Bruck, "Rank modulation for flash memories," IEEE Trans. on Inf. Theory, vol. 55, no. 6, pp. 2659–2673, June 2009.

[19] E. En Gad, A. Jiang, and J. Bruck, "Compressed encoding for rank modulation," in IEEE Int. Symp. on Information Theory Proc. (ISIT).   IEEE, Aug. 2011, pp. 884–888.

[20] Q. Li, "Compressed rank modulation," in 50th Annu. Allerton Conf. on Commun, Control, and Computing (Allerton), Oct. 2012, pp. 185–192.

[21] E. E. Gad, E. Yaakobi, A. Jiang, and J. Bruck, "Rank-modulation rewriting codes for flash memories," in Proc. IEEE Int. Symp. on Information Theory (ISIT).   IEEE, Jul. 2013, pp. 704–708.

[22] M. Qin, A. Jiang, and P. H. Siegel, "Parallel programming of rank modulation," in Proc. IEEE Int. Symp. on Information Theory (ISIT).   IEEE, Jul. 2013, pp. 719–723.

[23] M. Qin, E. Yaakobi, and P. H. Siegel, "Constrained codes that mitigate inter-cell interference in read/write cycles for flash memories," IEEE J. Sel. Areas Commun., vol. 32, no. 5, pp. 836–846, May 2014.

[24] S. Kayser and P. H. Siegel, "Constructions for constant-weight ici-free codes," in IEEE Int. Symp. on Information Theory (ISIT).   IEEE, Jul. 2014, pp. 1431–1435.

[25] R. Gabrys and L. Dolecek, "Constructions of nonbinary WOM codes for multilevel flash memories," IEEE Trans. Inf. Theory, vol. 61, no. 4, pp. 1905–1919, Apr. 2015.

[26] E. Yaakobi, P. H. Siegel, A. Vardy, and J. K. Wolf, "Multiple error-correcting WOM-codes," IEEE Trans. on Inf. Theory, vol. 58, no. 4, pp. 2220–2230, Apr. 2012.

[27] A. Bhatia, M. Qin, A. R. Iyengar, B. M. Kurkoski, and P. H. Siegel, "Lattice-based WOM codes for multilevel flash memories," IEEE J. Sel. Areas Commun., vol. 32, no. 5, pp. 933–945, May 2014.

[28] Q. Li and A. Jiang, "Polar codes are optimal for write-efficient memories." in 51th Annu. Allerton Conf. on Communication, Control, and Computing (Allerton), 2013, pp. 660–667.

[29] N. Papandreou, T. Parnell, H. Pozidis, T. Mittelholzer, E. Eleftheriou, C. Camp, T. Griffin, G. Tressler, and A. Walls, "Using adaptive read voltage thresholds to enhance the reliability of MLC NAND flash memory systems," in Proc. 24th Great Lakes Symp. on VLSI.   ACM, 2014, pp. 151–156.

[30] D.-H. Lee and W. Sung, "Estimation of NAND flash memory threshold voltage distribution for optimum soft-decision error correction," IEEE Trans. Signal Process., vol. 61, no. 2, pp. 440–449, Jan. 2013.

[31] B. Peleato, R. Agarwal, J. Cioffi, M. Qin, and P. H. Siegel, "Towards minimizing read time for NAND flash," in IEEE Global Communications Conf. (GLOBECOM).   IEEE, 2012, pp. 3219–3224.

[32] G. Dong, S. Li, and T. Zhang, "Using data postcompensation and predistortion to tolerate cell-to-cell interference in MLC NAND flash memory," IEEE Trans. Circuits Syst. I: Reg. Papers, vol. 57, no. 10, pp. 2718–2728, Oct. 2010.

[33] A. Torsi, Y. Zhao, H. Liu, T. Tanzawa, A. Goda, P. Kalavade, and K. Parat, "A program disturb model and channel leakage current study for sub-20nm NAND flash cells," IEEE Trans. Electron Devices, vol. 58, no. 1, pp. 11–16, Jan. 2011.

[34] E. Yaakobi, J. Ma, L. Grupp, P. Siegel, S. Swanson, and J. Wolf, "Error characterization and coding schemes for flash memories," in GLOBECOM Workshops (GC Wkshps).   IEEE, 2010, pp. 1856–1860.

[35] D. Nguyen, B. Vasic, and M. Marcellin, "Two-bit bit flipping decoding of LDPC codes," in IEEE Int. Symp. on Information Theory (ISIT).   IEEE, 2011, pp. 1995–1999.

[36] T. M. Cover and J. A. Thomas, Elements of Information Theory.   John Wiley & Sons, 2012.

[37] A. Berman and Y. Birk, "Constrained flash memory programming," in IEEE Int. Symp. on Information Theory (ISIT), 2011, pp. 2128–2132.

[38] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," IEEE Trans. Inf. Theory, vol. 56, no. 5, pp. 2307–2359, May 2010.

[39] J. Scarlett, A. Martinez, and A. Guillén i Fàbregas, "Mismatched decoding: Finite-length bounds, error exponents and approximations," submitted for publication. [Online: http://arxiv. org/abs/1303.6166], 2013.

[40] J. Y. N. Hui, "Fundamental issues of multiple accessing," Ph.D. dissertation, Mass. Inst. Technol., 1983.

[41] I. Csiszár and J. Körner, "Graph decomposition: A new key to coding theorems," IEEE Trans. Inf. Theory, vol. 27, no. 1, pp. 5–12, Jan. 1981.

[42] N. Merhav, G. Kaplan, A. Lapidoth, and S. Shamai Shitz, "On information rates for mismatched decoders," IEEE Trans. Inf. Theory, vol. 40, no. 6, pp. 1953–1967, Nov. 1994.

[43] A. Lapidoth, P. Narayan et al., "Reliable communication under channel uncertainty," IEEE Trans. Inf. Theory, vol. 44, no. 6, pp. 2148–2177, Oct. 1998.

[44] V. B. Balakirsky, "A converse coding theorem for mismatched decoding at the output of binary-input memoryless channels," IEEE Trans. Inf. Theory, vol. 41, no. 6, pp. 1889–1902, Nov. 1995.

[45] M. Alsan and E. Telatar, "Polarization as a novel architecture to boost the classical mismatched capacity of B-DMCs," arXiv preprint arXiv:1401.6097, 2014.

[46] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," IEEE Trans. Inf. Theory, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[47] D. P. Bertsekas, <u>Dynamic Programming and Optimal Control</u>. Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.

[48] Y. Wang, B. O'Donoghue, and S. Boyd, "Approximate dynamic programming via iterated Bellman inequalities," <u>Int. Journal of Robust and Nonlinear Control</u>, 2014.

[49] T. Parnell, N. Papandreou, T. Mittelholzer, and H. Pozidis, "Modelling of the threshold voltage distributions of sub-20nm NAND flash memory," in <u>IEEE Global Communications Conf. (GLOBECOM)</u>. IEEE, 2014, pp. 2351–2356.

IX. APPENDIX

Proof: (Theorem 1) The proof is very similar to that for Shannon's Channel Coding Theorem, but a few changes will be introduced to account for the mismatched decoder. Let $X \in \{1, 2\}^n$ denote the channel input and $Y \in \mathcal{Y}^n$ the channel output, with X_i and Y_i denoting their respective components for $i = 1, \ldots, n$. Throughout the proof, $\hat{P}(A)$ will denote the estimate for the probability of an event A obtained using the transition probabilities $\widehat{p_{ij}}$, $i = 1, 2$, $j = 1, \ldots, |\mathcal{Y}|$, to differentiate it from the exact probability $P(A)$ obtained using transition probabilities p_{ij}, $i = 1, 2$, $j = 1, \ldots, |\mathcal{Y}|$. The inputs are assumed to be symmetric, so $\hat{P}(X) = P(X)$ and $\hat{P}(X, Y) = \hat{P}(Y|X)P(X)$.

We start by generating 2^{nR} random binary sequences of length n to form a random code $\mathcal{C}$ with rate R and length n. After revealing the code $\mathcal{C}$ to both the sender and the receiver, a codeword $\mathbf{x}$ is chosen at random among those in $\mathcal{C}$ and transmitted. The conditional probability of receiving a sequence $\mathbf{y} \in \mathcal{Y}^n$ given the transmitted codeword $\mathbf{x}$ is given by $P(Y = \mathbf{y}|X = \mathbf{x}) = \prod_{i=1}^{n} p_{x_i y_i}$, where x_i and y_i denote the i-th components of $\mathbf{x}$ and $\mathbf{y}$, respectively.

The receiver then attempts to recover the codeword $\mathbf{x}$ that was sent. However, the decoder does not have access to the exact transition probabilities p_{ij} and must use the estimated probabilities $\widehat{p_{ij}}$ instead. When $p_{ij} = \widehat{p_{ij}} \quad \forall i, j$, the optimal decoding procedure is maximum likelihood decoding (equivalent to maximum *a posteriori* decoding, since inputs are equiprobable). In maximum likelihood decoding, the decoder forms the estimate $\hat{\mathbf{x}} = \arg\max_{\mathbf{x} \in \mathcal{C}} \hat{P}(\mathbf{y}|\mathbf{x})$, where $\hat{P}(Y = \mathbf{y}|X = \mathbf{x}) = \prod_{i=1}^{n} \widehat{p_{x_i y_i}}$ is the estimated likelihood of $\mathbf{x}$, given $\mathbf{y}$ was received.

Denote by $\hat{A}_\epsilon^{(n)}$ the set of length-n sequences $\{(\mathbf{x}, \mathbf{y})\}$ whose estimated empirical entropies

are ϵ-close to the typical estimated entropies:

$$\hat{A}_\epsilon^{(n)} \;=\; \{(\mathbf{x}, \mathbf{y}) \in \{1,2\}^n \times \mathcal{Y}^n : \tag{27}$$

$$\left| -\frac{1}{n} \log P(X = \mathbf{x}) - 1 \right| < \epsilon, \tag{28}$$

$$\left| -\frac{1}{n} \log \hat{P}(Y = \mathbf{y}) - \mu_Y \right| < \epsilon, \tag{29}$$

$$\left| -\frac{1}{n} \log \hat{P}(X = \mathbf{x}, Y = \mathbf{y}) - \mu_{XY} \right| < \epsilon \}, \tag{30}$$

where μ_Y and μ_{XY} represent the expected values of $-\frac{1}{n} \log \hat{P}(Y)$ and $-\frac{1}{n} \log \hat{P}(X, Y)$, respectively, and the logarithms are in base 2. Hence,

$$\mu_Y = -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{|\mathcal{Y}|} P(Y_i = k) \log \hat{P}(Y_i = k) \tag{31}$$

$$= -\sum_{k=1}^{|\mathcal{Y}|} \frac{p_{1k} + p_{2k}}{2} \log \left(\frac{\widehat{p_{1k}} + \widehat{p_{2k}}}{2} \right), \tag{32}$$

$$\mu_{XY} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{b=1}^{2} \sum_{k=1}^{|\mathcal{Y}|} P(X_i = b, Y_i = k) \log \hat{P}(X_i = b, Y_i = k) \tag{33}$$

$$= -\sum_{k=1}^{|\mathcal{Y}|} \left(\frac{p_{1k}}{2} \log \left(\frac{\widehat{p_{1k}}}{2} \right) + \frac{p_{2k}}{2} \log \left(\frac{\widehat{p_{2k}}}{2} \right) \right), \tag{34}$$

where the exact transition probabilities are used as weights in the expectation and the estimated ones are the variable values. Particularly, $(\mathbf{x}, \mathbf{y}) \in \hat{A}_\epsilon^{(n)}$ implies that $\hat{P}(Y = \mathbf{y} | X = \mathbf{x}) > 2^{n(1 - \mu_{XY} - \epsilon)}$ and $\hat{P}(Y = \mathbf{y}) < 2^{-n(\mu_Y - \epsilon)}$. We will say that a sequence $\mathbf{x} \in \{1,2\}^n$ is in $\hat{A}_\epsilon^{(n)}$ if it can be extended to a sequence $(\mathbf{x}, \mathbf{y}) \in \hat{A}_\epsilon^{(n)}$, and similarly for $\mathbf{y} \in \mathcal{Y}^n$.

First we show that with high probability, the transmitted and received sequences $(\mathbf{x}, \mathbf{y})$ are in the $\hat{A}_\epsilon^{(n)}$ set. The weak law of large numbers states that for any given $\epsilon > 0$, there exists n_0, such that for any codeword length $n > n_0$

$$P \left(\left| -\frac{1}{n} \log P(X = \mathbf{x}) - 1 \right| \geq \epsilon \right) < \frac{\epsilon}{3}, \tag{35}$$

$$P \left(\left| -\frac{1}{n} \log \hat{P}(Y = \mathbf{y}) - \mu_Y \right| \geq \epsilon \right) < \frac{\epsilon}{3}, \tag{36}$$

$$P \left(\left| -\frac{1}{n} \log \hat{P}(X = \mathbf{x}, Y = \mathbf{y}) - \mu_{XY} \right| \geq \epsilon \right) < \frac{\epsilon}{3}. \tag{37}$$

Applying the union bound to these events shows that for n large enough, $P \left((\mathbf{x}, \mathbf{y}) \notin \hat{A}_\epsilon^{(n)} \right) < \epsilon$.

When a codeword $\mathbf{x} \in \{1,2\}^n$ is transmitted and $\mathbf{y} \in \mathcal{Y}^n$ is received, an error will occur if there exists another codeword $\mathbf{z} \in \mathcal{C}$ such that $\hat{P}(Y = \mathbf{y}|X = \mathbf{z}) \geq \hat{P}(Y = \mathbf{y}|X = \mathbf{x})$. The estimated likelihood of $\mathbf{x}$ is greater than $2^{n(1-\mu_{XY}-\epsilon)}$ with probability at least $1 - \epsilon$, as was just shown. The other $nR - 1$ codewords in $\mathcal{C}$ are independent from the received sequence. For a given $\mathbf{y} \in \hat{A}_\epsilon^{(n)}$, let $S_\mathbf{y} = \left\{ \mathbf{x} \in \{1,2\}^n : \hat{P}(Y = \mathbf{y}|X = \mathbf{x}) \geq 2^{n(1-\mu_{XY}-\epsilon)} \right\}$ denote the set of input sequences whose estimated likelihood is greater than $2^{n(1-\mu_{XY}-\epsilon)}$. Then

$$1 = \sum_{\mathbf{x} \in \{1,2\}^n} \hat{P}(X = \mathbf{x}|Y = \mathbf{y}) \tag{38}$$

$$> \sum_{\mathbf{x} \in S_\mathbf{y}} \hat{P}(Y = \mathbf{y}|X = \mathbf{x}) \frac{P(X = \mathbf{x})}{\hat{P}(Y = \mathbf{y})} \tag{39}$$

$$> |S_\mathbf{y}| 2^{n(1-\mu_{XY}-\epsilon)} 2^{-n} 2^{n(\mu_Y-\epsilon)} \tag{40}$$

which implies $|S_\mathbf{y}| < 2^{n(\mu_{XY}-\mu_Y+2\epsilon)}$ for all $\mathbf{y} \in \hat{A}_\epsilon^{(n)}$.

If $(\mathbf{x}, \mathbf{y}) \in \hat{A}_\epsilon^{(n)}$, any other codeword causing an error must be in $S_\mathbf{y}$. Let E_i, $i = 1, \ldots, nR-1$ denote the event that the i-th codeword in the codebook $\mathcal{C}$ is in $S_\mathbf{y}$, and F the event that $(\mathbf{x}, \mathbf{y})$ are in $\hat{A}_\epsilon^{(n)}$. The probability of error can be upper bounded by

$$P(\hat{\mathbf{x}} \neq \mathbf{x}) = P(F^c) P(\hat{\mathbf{x}} \neq \mathbf{x}|F^c) + P(F) P(\hat{\mathbf{x}} \neq \mathbf{x}|F) \tag{41}$$

$$\leq \epsilon P(\hat{\mathbf{x}} \neq \mathbf{x}|F^c) + \sum_{i=1}^{2^{nR}-1} P(E_i|F) \tag{42}$$

$$\leq \epsilon + 2^{nR} |S_\mathbf{y}| 2^{-n} \tag{43}$$

$$\leq \epsilon + 2^{n(R+\mu_{XY}-\mu_Y-1+2\epsilon)} \tag{44}$$

Consequently, as long as

$$R < \frac{1}{2} \sum_{k=1}^{|\mathcal{Y}|} \left(p_{1k} \log(\widehat{p_{1k}}) + p_{2k} \log(\widehat{p_{2k}}) \right) - (p_{1k} + p_{2k}) \log \left(\frac{\widehat{p_{1k}} + \widehat{p_{2k}}}{2} \right), \tag{45}$$

for any $\delta > 0$, we can choose ϵ and n_ϵ so that for any $n > n_\epsilon$ the probability of error, averaged over all codewords and over all random codes of length n, is below δ. By choosing a code with average probability of error below δ and discarding the worst half of its codewords, we can construct a code of rate $R - \frac{1}{n}$ and maximal probability of error below 2δ, proving the

achievability of any rate below the bound $C_{P,\hat{P}}$ defined in Eq. (15). This concludes the proof.

■