

Macroscopic and Microscopic Approaches in Sector Failure Rate Estimation

Alexander V. Kuznetsov and Raman Venkataramani

Seagate Research, Pittsburgh, PA 15222 USA

The sector failure rate (SFR) is extremely small at normal operating conditions of hard disk drives. In practice, it cannot be obtained by counting as that would require prohibitively large simulation times. Therefore, appropriate statistical models characterizing the distribution of error symbols are used in order to estimate the SFR. In this paper, we look at the underlying philosophy of existing estimation methods and classify them into macroscopic and microscopic types. We observe that the microscopic approach is well suited for certain iterative channels.

Index Terms—Error correction coding, error modeling, failure rate estimation, minimum relative entropy.

I. INTRODUCTION

IN data storage systems such as magnetic and optical storage devices, there is a need for extremely high reliability in terms of recovering the stored data. This is achieved by using powerful error correction codes (ECCs) for the recording channel. The simplified block diagram of a typical magnetic recording channel is shown in Fig. 1. The block labeled “channel” represents the combination of physical channel and the inner error correcting codes designed for the particular physical channel. An additional outer ECC is used to correct any residual bit errors resulting from the inner decoder. The purpose of the outer code is to guarantee extremely low error rates. Reed–Solomon (RS) codes are used in today’s recording systems due to their high code rates and their ability to correct burst errors efficiently.

In magnetic and optical recording systems, bits are read and written in blocks or sectors, where a sector can either be a single ECC codeword or contain a small number of such codewords interleaved so that in an event of a burst error, the symbol errors would be distributed evenly among the codewords. Thus, a commonly used figure of merit for a storage device is its sector failure rate (SFR), defined as the rate at which the ECC is unable to decode any part of the sector. Sector failure rates are typically extremely low (10^{-10} or smaller) and cannot be estimated by simulation and counting. Thus, the estimation of sector failure rates is done by first modeling the error statistics of the input to the ECC decoder, and then using the model to estimate the SFR. Because the model is analytically tractable, we can predict even extremely small rates of sector failure.

In this paper, we examine a few existing flavors of SFR estimation based on modeling of error statistics. From a philosophical standpoint, these approaches can be classified as *microscopic* and *macroscopic* methods having their own advantages and disadvantages. In Section III-A, we present a macroscopic approach based on a multinomial model for error events. For this model, we show how to estimate the model parameters from a sample data set and estimate the SFR and confidence bounds. In Section IV, we present a microscopic method which is applicable to channels with iterative decoding. Finally, we present an example of SFR estimation.

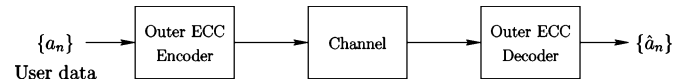


Fig. 1. Block diagram of a channel with an outer error correcting code.

II. DEFINITIONS

We assume that the outer ECC code is a Reed–Solomon (RS) code whose q -bit symbols are elements of $\text{GF}(2^q)$. The RS encoder can produce codewords of length $N \leq 2^q - 1$ symbols. For sufficiently large q , the entire sector can be a single codeword of the RS code. Otherwise, multiple codewords are interleaved in a sector. An RS code can correct as many as T symbol errors in the received codeword and encode $N - 2T$ message symbols in a codeword of length N .

Denote the written and decoded codewords respectively by $\mathbf{x} = (x_1, x_2, \dots, x_N)$ and $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N)$, where $x_i, \tilde{x}_i \in \text{GF}(2^q)$. The *error indicator vector* $\boldsymbol{\xi}$ is a binary sequence defined as $\xi_i = 1$ if $x_i \neq \tilde{x}_i$ and $\xi_i = 0$ otherwise. Denote its weight distribution by

$$r_n = P(w_H(\boldsymbol{\xi}) = n)$$

where $P(\cdot)$ denotes the probability and $w_H(\boldsymbol{\xi}) = \sum_i \xi_i$ is the Hamming weight. Thus, the *word failure rate* (WFR) is

$$\rho_W(T) \stackrel{\text{def}}{=} P(w_H(\boldsymbol{\xi}) > T) = \sum_{n>T} r_n \quad (1)$$

where T is the ECC capability. Assuming that I codewords are interleaved in sector, the *sector failure rate* (SFR) $\rho_S(T)$ is given by the union bound: $\rho_S(T) \leq I\rho_W(T)$. Usually $\rho_S(T) \simeq I\rho_W(T)$ if the word failures are independent and the WFR is small. The quantity (1) is difficult to compute because the error weight distribution of the error vectors is analytically intractable. Furthermore, the typical values for $\rho_W(T)$ are so small that it is infeasible to estimate them by simulation and counting.

The traditional approach to estimate failure rates is to use a model for error vector statistics and compute the failure rates analytically. The model parameters are estimated using a sample collection of actual error vectors for the given channel and its inner code. This method requires far fewer number of error vectors than to estimate the failure rates by directly counting failures. In the following sections, we provide models for the

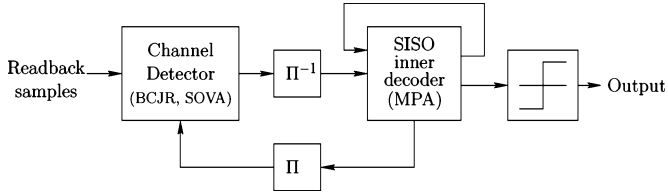


Fig. 2. Simplified illustration of iterative decoding.

error statistics of the error vectors and show how to estimate the word/sector failure rate.

III. MACROSCOPIC AND MICROSCOPIC APPROACHES

The errors at the output of a Viterbi detector are bursty and localized because the ISI in the system is approximately time-limited. For systems that use codes such as run-length limited (RLL) codes and convolutional codes, we observe a similar phenomenon that the errors tend to be bursty and localized. Since a single bit error in a symbol produces a symbol error, we find that a bursty sequence of bit errors usually causes a train of symbol errors. However, for systems that employ iterative codes such as low-density parity check (LDPC) codes or turbo codes, the erroneous bits can be scattered across the entire error sector instead of being bursty. Fig. 2 is an illustration of iterative decoding. Note that the code is applied to a set of input bits and being and then passed through an interleaver Π . This creates a strong long-range correlation of erroneous bits and symbols. The problem is to determine good statistical models for both cases, i.e., for more simple noniterative channels with bursty errors, and also for more complex channels with iterative decoding.

Philosophically, all methods for estimation of the properties of the error indicator vectors may be classified as either *microscopic* or *macroscopic*. We say that a method is microscopic if it attempts to capture the precise correlation among symbol errors in terms of their locations. Likewise, the algorithm to estimate the model parameters uses the actual symbol error locations in the available training data set. In general, it is a difficult task to determine this precise microscopic model for iterative coded channels because they are very complex due to the long-range correlations present in the symbol errors. Furthermore, depending on the exact code and interleaver that is used, each iteratively coded channel has its own characteristic microscopic model. On the other hand, the macroscopic method attempts to model only the weight of the error indicator vectors, paying no attention to the actual locations of the bit or symbol errors. The model estimation also relies only on the weights of the error indicator vectors in the training data, rather than the precise error locations.

The macroscopic method is generally simple to analyze: the main problem is to model the weight distribution function and estimate the model parameters from training data. From experience, the macroscopic approach also tends to work well for simple noniterative codes. However, they tend to be inaccurate for channels with iterative codes and long-range dependence between symbol errors prompting us to resort to a microscopic model. Although a microscopic approach might yield more accurate results, it is often difficult to find an accurate model as it

requires the knowledge of the exact code and interleaver used. In the following sections, we consider two methods for SFR estimation: a macroscopic approach based on the multinomial modeling and a microscopic approach based on modeling the error events with two types of components. These methods are discussed in detail below.

A. Multinomial Model

According to this model (a) an error indicator vector consists of K independent error events, and (b) an error event consists of a collection of up to L ones (representing symbols in error) including the trivial error event with no symbols in error.

In practice, the nontrivial error events are so infrequent that we assume that they do not overlap. Let the error events in an error vector \mathbf{e} be denoted by $\{E_k : k = 1, \dots, K\}$, and their weights by $w_H(E_k)$. Let p_l for $l = 0, \dots, L$ denote the probability of seeing an error event of weight k symbols. This model results in the *multinomial* distribution for the error vector weight

$$r_n = \sum_{\mathcal{K}_n} \frac{K!}{k_0!k_1! \dots k_L!} \prod_{l=0}^L p_l^{k_l} \quad (2)$$

where k_l is the number of events in the error indicator vector with weight l , and \mathcal{K}_n is the set of all combinations of indices $\{k_l : l = 0, \dots, L\}$ such that their sum is K and the total weight is n

$$\sum_{l=0}^L k_l = K \quad \text{and} \quad \sum_{l=0}^L lk_l = n.$$

This is perhaps the most widely used model for modeling error vector weights [1]–[7]. For example, Feng *et al.* [3] consider byte size symbols and error events with a maximum length of $L = 3$ bytes. The model parameters are estimated using the “common sense” approach based on the frequency of occurrence of each error event. Keirn *et al.* [2] examine both the multinomial and block multinomial models.

The multinomial method is a macroscopic approach, because we model the weight of error vectors rather than the individual error symbol locations. The model parameters are $p_l : l = 0, \dots, L$.

B. Markov Chain Models

Sometimes the symbol errors are modeled as Markov random processes. An example is the Gilbert model [8], [9] or the generalized Gilbert model [10] which can be represented using a state diagram. The model parameters are the unknown transition probabilities. These models are microscopic because they explicitly model the error locations, not the error weight.

In this section, we give an example of a slightly richer model than the Gilbert model. According to this model, given that a symbol error has occurred the probability of seeing another symbol error is q_{m+1} , where m is the run-length of symbol errors leading up to that time. To reduce the model complexity, we let $q_m = q_M$ for all $m \geq M$. This yields the state diagram shown in Fig. 3. When the chain returns to the zero state from the nonzero state m , the generation of an error event $E(m)$ is completed by attaching the final “0” to the run of m consecutive ones. The reduced complexity model is sufficient in many

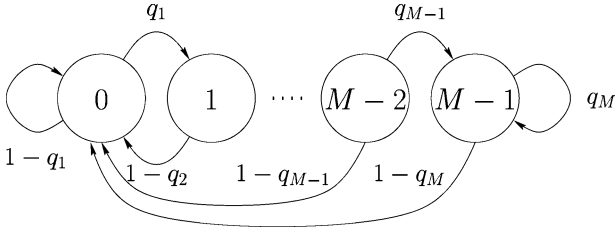


Fig. 3. Model for error event generation.

practical cases. In fact, a model with fewer parameters is preferable because the model parameters can then be estimated more accurately from limited data than a more complex model.

The probability of the m th error event is

$$p_m \stackrel{\text{def}}{=} P(E_m) = (1 - q_{m+1}) \prod_{l=1}^m q_l, \quad m > 0. \quad (3)$$

In the above definition, $q_m = q_M$ for all $m \geq M$. For this model, there is an additional step of finding the modeled weight distribution r_n which is done as follows.

Define the following matrix whose rows and columns indexes are assumed to start from 0:

$$\Pi(x) = \begin{pmatrix} 1 - q_1 & q_1 x & 0 & \dots & 0 \\ 1 - q_2 & 0 & q_2 x & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 - q_{M-1} & 0 & 0 & \dots & q_{M-1} x \\ 1 - q_M & 0 & 0 & \dots & q_M x \end{pmatrix}. \quad (4)$$

The entries of $\Pi(x)$ are monomials in x , and the coefficient of $\Pi_{m,m'}(x)$ is the probability of transition from state m to state m' whereas the presence of an x in a term indicates that the transition induces a symbol error. Note that for a first-order model, i.e., $M = 1$, (4) degenerates to the scalar $\Pi(x) = (1 - q_1) + q_1 x$. Clearly, $\Pi(1)$ is the probability transition matrix for the state diagram in Fig. 3.

Define $R(x) = \Pi(x)^N$ whose entries are polynomials in x . The powers of x in $R(x)$ enumerate the symbol errors in a codeword of length N . Thus, the entry $R_{m,m'}(x)$ is a polynomial in x in which the coefficient of x^n is the probability of a run of length N starting at state m and ending at state m' consisting of a total of exactly n symbol errors. Since we are primarily interested in the number of symbol errors and not the final state, we have

$$\pi_0 R(x) u^T = \sum_{n \geq 0} P(w(\boldsymbol{\xi}) = n) x^n \quad (5)$$

where π_0 is a vector containing the initial probabilities of the M states, and $u = (1, 1, \dots)$ is a row vector of length M consisting of ones. For simplicity, we assume that all error vectors represent paths in the state diagram starting at the zeroth state. In practice, this assumption has a very minimal effect on accuracy of our estimates because the model order M is much smaller than the length of a codeword N and the probabilities $\{q_m\}$ are quite small. Thus, we take $\pi_0 = (1, 0, \dots, 0)$. Combining (1)

and (5), we obtain the following expression for the word failure rate:

$$\rho_W(T) = \sum_{n > T} r_n \quad (6)$$

where r_n is the coefficient of x^n in the polynomial $R(x) = \pi_0 \Pi(x)^N u$. Note that the above expression is not computationally very intensive. The quantity $\Pi(x)^N$ can be implemented in roughly $\log_2(N)$ polynomial matrix multiplications. We can also accumulate the coefficients of x^{T+1} and higher powers of x after each multiplication.

C. Estimation of Model Parameters

Having chosen an appropriate model for the symbol errors, its parameters are estimated from a library of actual error vectors for the particular channel and codes used. Let the data set consist of C error indicator vectors $\mathcal{C} = \{\boldsymbol{\xi}_c : c = 1, \dots, C\}$.

First, consider the macroscopic approach. Let \hat{r}_n denote the histogram of the error vector weights in the sample set of error vectors: $\hat{r}_n = C_n/C$, where C_n is the number of error indicator vectors with weight n and $C = \sum_k C_k$ is the total number of vectors in the data set \mathcal{C} . Let $\{r_n(\mathbf{p})\}$ denote the modeled weight distribution for the model with parameters $\mathbf{p} = \{p_0, p_1, \dots, p_L\}$. Our aim is to determine the model parameters for which the modeled weight distribution vector $r(\mathbf{p}) \stackrel{\text{def}}{=} \{r_n(\mathbf{p})\}$ matches the experimental weight distribution vector $\hat{r} \stackrel{\text{def}}{=} \{\hat{r}_n\}$. The optimal choice of the model parameters minimizes a cost function or a distance measure between the two distributions.

Some researchers [11] have considered *ad hoc* cost functions such as the mean squared error between the counted and modeled *weight distributions*

$$\mathbf{p}^* = \arg \min_{\mathbf{p}} \|\hat{r} - r(\mathbf{p})\|^2 \quad (7)$$

or sometimes, the mean squared error between the counted and modeled *word failure rates* (or equivalently, sector failure rates)

$$\mathbf{p}^* = \arg \min_{\mathbf{p}} \sum_{T \geq 0} (\hat{\rho}_W(T) - \rho_W(\mathbf{p}, T))^2 \quad (8)$$

where $\hat{\rho}_W(T)$ and $\rho_W(\mathbf{p}, T)$ are the counted and modeled WFR estimates for an error correcting capability of T symbols.

Although the above cost functions are conceptually simple, they are suboptimal. Indeed, the optimal model parameter is the maximum-likelihood (ML) model estimate from the empirical weight distribution. It is a fairly easy exercise to check that this solution yields the so-called *minimum relative entropy solution*: $\mathbf{p}^* = \arg \min_{\mathbf{p}} C(\mathbf{p})$, where

$$C(\mathbf{p}) = D(\hat{r} \| r(\mathbf{p})) = \sum_n \hat{r}_n \log \left(\frac{r_n(\mathbf{p})}{\hat{r}_n} \right) \quad (9)$$

is the familiar *relative entropy function* or the *Kullback–Leibler distance* between the two weight distributions [12]. All of the above optimizations can be done using a gradient descent algorithm [13]. Compared to (9), the cost functions (7) and (8) tend to place too much importance on matching the distributions for

small values of index n (usually large values of $r_n(\mathbf{p})$) while ignoring the tails where $r_n(\mathbf{p})$ is smaller. The result is that the tail probabilities are estimated poorly.

We should note that our recommendation to use (9) does not require a specific model. It could be a multinomial model, or any other model with a known distribution $r_n(\mathbf{p})$. In other words, if we have a model for the weight distribution $r_n(\mathbf{p})$, then the correct metric for “fitting” it to the experimental data is the minimal relative entropy (9), not (7) or (8) as used by some. Fitting methods work well for noniterative channels, but again often does not work for iterative channels.

For the microscopic approach, we solve the ML estimation problem using the actual data set \mathcal{C} rather than the weight distributions

$$\mathbf{p}^* = \arg \max_{\mathbf{p}} P(\mathcal{C}; \mathbf{p}) \quad (10)$$

where $P(\mathcal{C}; \mathbf{p})$ is the probability of observing the data set \mathcal{C} assuming the model \mathbf{p} .

As an example, consider the estimation of model parameters for the (microscopic) Markov model shown in Fig. 3. Let α_m be the total number of times that an error indicator vector from the training set traverses the path from state m to state $m + 1$ (for $m \leq M - 2$) or looped-back to state M (if $m = M - 1$). Similarly, let β_m be the number of times that the path from state m to 0 is traversed. Evidently, $\alpha_m + \beta_m$ is the number of times that state m is visited. Then, it is an easy exercise to show that the optimal estimates for \mathbf{q}_m are given by

$$q_{m+1}^* = \frac{\alpha_m}{\alpha_m + \beta_m}, \quad m = 0, \dots, M - 1$$

and \mathbf{p}^* is computed from \mathbf{q}^* using (3). Also, the modeled weight distribution $r_n(\mathbf{p}^*)$ and WFR are found using the algorithm described in Section III-B.

IV. MICROSCOPIC METHOD WITH TWO COMPONENT TYPES

We shall now extend multinomial model to SFR estimation of iterative channels as well. Let $E(m)$ denote an error event (EE) of length m in ξ : it consists of $m \geq 1$ ones followed by a zero, e.g., $E(3) = 1110$. Note that we exclude the single-zero event $E(0) = 0$ in the list of error events. A sequence of $n \geq 1$ consecutive zeros $I(n)$ is called an *error-free run of length n* . Thus, we can always parse an error indicator vector into a unique sequence of interlacing error events E_i and error-free runs I_i as follows:

$$\xi = (E_1, I_1, E_2, I_2, \dots, E_t, I_t).$$

For example, $\xi = (1100100000100111000000)$ is represented as $\xi = (E_1, I_1, E_2, I_2, E_3, I_3, E_4, I_4)$, where $E_1 = 110$, $E_2 = E_3 = 10$, $E_4 = 1110$, $I_1 = 0$, $I_2 = 0000$, $I_3 = 0$, and $I_4 = 00000$.

Main Assumptions: Rather than modeling the symbol errors using a state model as in Fig. 3, we choose to model only the isolated error events and the number of such error events. The events E_i are modeled as independent identically distributed random objects. The number of such error events in a sector

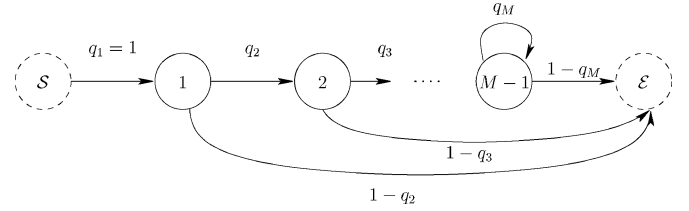


Fig. 4. Model for isolated error event generation, where \mathcal{S} and \mathcal{E} denote the start and end states.

is modeled as a random variable independent of the E_i . Furthermore, assume that each E_i (but not the error-free runs I_i) is described by finite state model with start and end states \mathcal{S} and \mathcal{E} , respectively, as shown in Fig. 4. In other words, conditioned on the first symbol error, the remaining errors are given precisely by the state model in Fig. 3 without the loop-back at the “0” state, i.e., $q_1 = 1$. The estimation of transition probabilities in the trellis from experimental data can be derived in a straightforward way using the ML estimation (10) described in Section III-C, but it is not included in this paper for brevity.

Remark 1: We reiterate that the Markov model is used only to model these nontrivial error events. The error-free runs are not given by the Markov model, and we do not try to fit them to the Markov model for its parameter estimation. Instead, the number of error events in a codeword is modeled separately as an independent entity. The PDF of the error-free runs is used indirectly as described below. Thus, the algorithm is to parse the experimentally observed error indicator vectors into nontrivial error events E_i (bursts of ones), and error-free runs I_j , and extract the transition probabilities from the data $\{E_1, E_2, \dots, E_t\}$ alone.

A. SFR Estimation Using Microscopic Method

By definition, $\rho_W(T) = P(w_H(\xi) > T)$ is the word failure rate, where T is the maximum number of errors that the ECC can correct. Given the number of error events k in the sector, we can also define the *conditional word failure rate* as

$$\rho_W(T, k) = P\left(\sum_{i=1}^k w_H(E_i) > T \mid \xi \text{ contains } k \text{ EEs}\right). \quad (11)$$

Here, we should note that if $E_i = E(m)$, then $w_H(E_i) = m$ according to our definition of an error event of length m . The probability $p(m)$ that the error event E_i has the Hamming weight m is calculated using (3) and estimated transition probabilities of the Markov chain q_j . In practice, given the transition probabilities of the Markov chain q_j and the number of error events k , the conditional word failure rate $\rho_W(T, k)$ can be easily calculated recursively using a computer. As we already mentioned before, the number of error events k in a codeword is a random variable. Therefore, its PDF $Q(k)$ must be estimated from statistics obtained from real or simulated data. Given $Q(k)$, the word failure rate is calculated as follows:

$$\rho_W(T) = \sum_{k=0}^{\infty} Q(k) \rho_W(T, k). \quad (12)$$

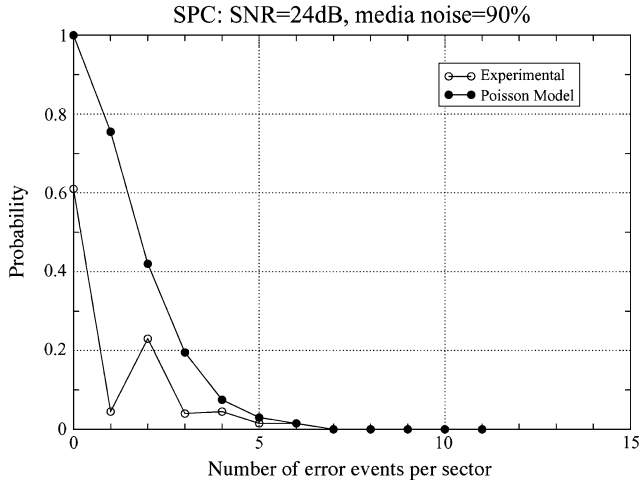


Fig. 5. Empirical distribution $Q(k)$ for an iterative SPC based channel.

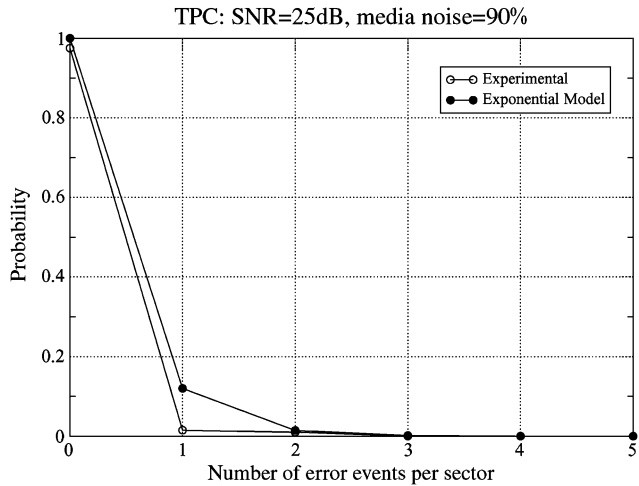


Fig. 6. Empirical distribution $Q(k)$ for an iterative TPC-based channel.

B. Estimation of $Q(k)$

Collecting statistics for estimating probabilities q_k and $Q(k)$ requires long simulations or tests of real drives. But even after lengthy testing, we rarely observe error events longer than some small integer J for moderate SNRs. Fig. 5 shows an example of the simulation results for a perpendicular iterative channel using a single parity check (SPC) code. In this case, the maximum number of observed error events in a sector was $J = 12$ (bottom curve). Fig. 6 shows another example of the simulation results in a perpendicular iterative channel using a turbo product code (TPC) where the maximum number of observed error events in a sector was $J = 6$.

Different experiments with SFR estimation show the following.

- 1) We cannot apply (12) with $Q(k) = 0$ for $k > J$ since this leads to an abrupt change of the slope of SFR curves as a function of T at some $T = T_0$ and wrong SFR estimates for $T > T_0$. This is because the tail part of the sequence $Q(k)$ is too small to be accurately estimated by counting, yet crucial for estimation. Therefore, the distribution $Q(k)$ must be extrapolated for $k > J$.

- 2) The distribution $Q(k)$ can have different types of tails for different detectors. For example, we found that iterative channels using TPC and LDPC codes have exponential tails

$$\hat{Q}(k) = e^{-\lambda k}, \quad k > J$$

and iterative channels using SPC have tails with a Poisson distribution

$$\hat{Q}(k) = \frac{\lambda^k}{k!} e^{-\lambda}, \quad k > J.$$

- 3) Having identified the tail type, we can use data fitting techniques such as those described in the last section to choose specific parameters of the distributions. Suboptimal metrics can also be used. For example, we can choose parameter λ that minimizes the following quantity:

$$\sum_{k=k_0}^J \left| \frac{\hat{Q}(k) - Q(k)}{\hat{Q}(k)} \right|^\alpha$$

where k_0 is an integer chosen in the range $0 < k_0 < J$, and $\alpha > 0$ is some constant. A quadratic form $\alpha = 2$ is a convenient choice.

- 4) Therefore, in (12), we use experimental values $Q(k)$ for $k \leq J$ and modeled values $Q(k)$ for $k > J$.

Remark 2: An experimental histogram for the number of non-trivial error events per sector should be used *as is*. It could be fancy for some iterative channels, so do not approximate it. The tail of the histogram will be always missing due to limited simulation time. Do not use zeros for the tail. We can guess the analytical shape of the tail (exponential, Poisson, or something else). For different classes of iterative channels the guess could be different. Choose the parameter(s) of the analytical expression for tail by matching with existing experimental part of tail with analytical part. In our case, this last approach was working well for all types of iterative channels.

C. Simulation Results

In the following example, we evaluate the performance of the *descriptive* and *predictive* capabilities of various SFR estimation methods. The descriptive capability refers to how well the analytic SFR estimate from a data set matches the counted estimate from the same data set. The predictive capability refers to how well the analytic SFR estimate from a data set matches the counted SFR, which uses a much larger data set. Thus, to evaluate the predictive capability of a method, we first estimate the SFR analytically using a limited set of data. We then perform more simulations and use this data to get a more reliable counted estimate, which usually yields more points on the SFR plot. Finally, we compare the analytic estimate and the counted estimates. Another way of viewing this is that we collect a lot of data for the counting estimation but only a part of it is used in the analytic estimation.

Consider a TPC system operating at 24 dB. Each sector consists of $N = 500$ symbols (including the ECC parity symbols) which are 10 bits wide. The full data set contains 79711 sectors and the partial data set contains 10% of this data. Suppose we use an RS code with correction capability of T symbol errors,

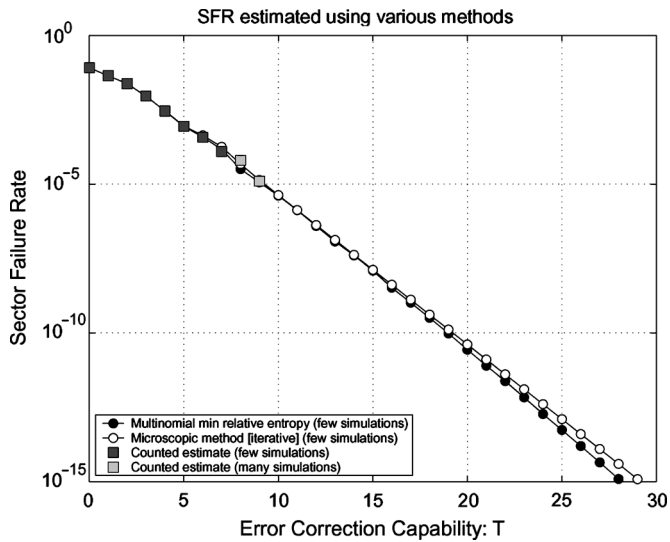


Fig. 7. SFR estimate for the TPC system at 24 dB.

we require $2T + 1$ parity symbols. Thus, the code rate would be $1 - (2T + 1)/N$. Fig. 7 shows the SFR estimates as a function of T using several methods from the partial data and counted values from the full data set. These methods appear to be consistent with each other and match the extrapolated counted values well.

V. CONCLUSION

We examined several existing approaches to estimate SFR for iterative and noniterative channels. Philosophically, they may be classified as macroscopic and microscopic depending on whether we model the error weights or the actual error locations. For the macroscopic approach, we introduced the well-known minimum relative entropy solution to estimate the model parameters from a sample library of error vectors for any chosen model. These can then be used to analytically compute the word and sector failure rates. The macroscopic approach tends to be easier to analyze and applicable to a wide range of noniterative channels. However, it is inaccurate for iterative channels where the underlying codes create long-range correlations in symbol errors. For these channels, the microscopic

approach works better. We observe through simulations that the microscopic models can be used for certain iterative channels where the multinomial model fails.

ACKNOWLEDGMENT

This work was supported by Seagate Technology.

REFERENCES

- [1] R. Martin, "Error bounds for concatenated Reed-Solomon/convolutional codes," in *Proc. MILCOM'79*, Oct. 1979, vol. 2, pp. 23.5.1–23.5.6.
- [2] Z. A. Keirn, V. Y. Krachkovsky, E. F. Haratsch, and H. Burger, "Use of redundant bits for magnetic recording: Single-parity codes and Reed-Solomon error-correcting code," *IEEE Trans. Magn.*, vol. 40, no. 1, pp. 225–230, Jan. 2004.
- [3] W. Feng, A. Vityaev, G. Burd, and N. Nazari, "On the performance of parity codes in magnetic recording systems," in *Proc. IEEE GLOBECOM*, Nov. 2000, pp. 1877–1881.
- [4] R. Cideciyan, J. Coker, E. Eleftheriou, and R. Galbraith, "Noise predictive maximum likelihood detection combined with parity-based post-processing," *IEEE Trans. Magn.*, vol. 37, no. 2, pp. 714–720, Mar. 2001.
- [5] C. M. Riggle and G. McCarthy, "Design of error correction systems for disk drives," *IEEE Trans. Magn.*, vol. 34, no. 4, pp. 2362–2371, Jul. 1998.
- [6] K. Saeki and Z. Keirn, "Optimal combination of detection and error correction coding for magnetic recording," *IEEE Trans. Magn.*, vol. 37, no. 2, pp. 708–713, Mar. 2001.
- [7] J. Moon, J. Park, and J. Lee, "Cyclic redundancy check code based high-rate error-detection code for perpendicular recording," *IEEE Trans. Magn.*, vol. 42, no. 5, pp. 1626–1628, May 2006.
- [8] J. R. Yee and E. J. Weldon, "Evaluation of the performance of error-correcting codes on a Gilbert channel," *IEEE Trans. Commun.*, vol. 43, no. 8, pp. 2316–2323, Aug. 1995.
- [9] E. N. Gilbert, "Capacity of burst noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253–1266, Sep. 1960.
- [10] J. K. Wolf, "ECC performance of interleaved RS codes with burst errors," *IEEE Trans. Magn.*, vol. 34, no. 1, pp. 75–79, Jan. 1998.
- [11] J. L. Sonntag, "New technique for estimating ECC performance of channels with large blocks," Lucent Technologies, Tech. Rep., 2000.
- [12] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
- [13] D. G. Luenberger, *Optimization by Vector Space Methods*. New York: Wiley, 1997.

Manuscript received May 18, 2007. Corresponding author: A. V. Kuznetsov (e-mail: alexander.v.kuznetsov@seagate.com).