The Read Channel

Aleksandar Kavčić and Ara Patapoutian

***Abstract*— In this paper, we provide a survey of the novel read channel technologies that found their implementation in products over the past decade, and we outline possible technology directions for the future of read channels. Recently, magnetic recording read channels have undergone several changes. In addition to switching from longitudinal to perpendicular recording channels, detectors tuned to media noise sources are now readily implemented in read channel chips. Powerful numerical techniques have emerged to evaluate the capacity of the magnetic recording channel. Further, improved coding/decoding methods have surfaced both for the incumbent *Reed-Solomon* codes and the promising *low-density parity-check* (LDPC) codes. The paper is a tutorial-like survey of these emerging technologies with the aim to propel the reader to the forefront of research and development in the areas of signal processing and coding for magnetic recording channels.**

I. Introduction

The past decade gave rise to several advances in read channel technologies for magnetic recording. Our goal here is to provide a tutorial-type overview of these advances, and to point to research challenges for several years in advance.

We start the paper by listing several available models for the magnetic recording channels. We emphasize that any advance in designing novel data storage channels begins by picking the proper model for the signal processing task at hand. For this reason, we give a detailed exposition of available channel models in Section II.

In the decade from the late 1990s to the late 2000s we witnessed several changes in the magnetic storage technology, the major event being the shift from longitudinal to perpendicular recording. For this reason, we only consider perpendicular recording channels in this paper. In the field of channel engineering, we highlight four areas where considerable advances have been made in the past decade, and we devote one section to each one of these areas:

- **Section III - Channel detectors:** Novel detectors capable of combatting media noise have been introduced.
- **Section IV - Capacity computation methods:** Powerful methods for numerically evaluating the capacity became available.
- **Section V - Error correction codes:** Powerful iteratively decodable codes are on the verge of being introduced in products, while at the same time more potent decoders of traditional Reed-Solomon codes are being invented.
- **Section VI - Data access techniques:** The constant evolution of servo processing, head fly-height estimation and synchronization algorithms in ever decreasing signal-to-noise ratios (SNRs) enables maintaining accurate data access at higher areal densities.

II. Channel Models

To model the magnetic recording channel accurately and yet in an reasonable time interval is one of the important tasks facing read channel engineers. There are many possible ways to model the channel, but no model will be completely accurate. The choice of the channel model depends on several parameters such as: the desired level of accuracy, the signal processing task at hand, domain of operation (analog or digital), etc. In this section, we outline a few channel models commonly found in the literature (going from the simplest to more complex).

A. Basic notation

As a general rule, stochastic processes will be denoted by uppercase letters, while their realizations will be denoted by lowercase letters. For example, if t denotes time (continuous time), then $R(t)$ denotes a continuous-time stochastic process, while $r(t)$ denotes one of its possible realizations. Similarly, if k denotes discrete time, then X_k denotes a discrete-time stochastic process, while x_k denotes one of its possible realizations. The expected value of a random variable X is denoted as $\mathrm{E}X$ or $\mathrm{E}(X)$. A sequence of discrete-time variables x_i from time index k to time index ℓ is shortly denoted as $x_k^\ell = (x_k, x_{k+1}, \dots, x_{\ell-1}, x_\ell)$.

In digital magnetic recording, bits are written binary signals because the magnetization can point in one of two possible directions. We will denote by $X_k \in \{-1, +1\}$ the *bipolar* random variable that represents the written binary symbol on the magnetic medium at discrete-time k. Its realization is denoted by x_k. The written symbol can equivalently be denoted in *binary* notation as $A_k \in \{0, 1\}$ (whose realization is denoted by a_k), where $X_k = (-1)^{A_k}$. We shall denote a *transition* as $B_k = X_k - X_{k-1}$.

B. A simple PAM channel model

The simplest continuous-time magnetic recording channel model is a pulse amplitude modulated (PAM) channel. We note that such a channel model cannot capture all the subtleties (media noise, nonlinearities, etc.) of a modern recording channel, but is useful, as it easily illustrates the *readback* mechanism. There are two basic ways to model the channel: 1) using the channel's transition response, or 2) using the channel's bit response (symbol response). The former approach emphasizes the channel where, as will see, many noise sources are associated with a transition, whereas the latter approach emphasizes detector design.

The transition response is the response of the read head to a magnetization transition. As already stated earlier, there are three possible transitions: a positive transition (denoted by $B_k = 2$), a negative transition (denoted by $B_k = -2$), and a non-transition (denoted by $B_k = 0$). If we denote by $p(t)$ the transition response, then we can express the channel output as

$$Y(t) = \sum_{k=-\infty}^{\infty} B_k \cdot p(t - kT) + N(t), \tag{1}$$

where $N(t)$ is the channel noise, and T is the symbol interval. A typical assumption for such simple channel models is that the noise is Gaussian and white within the Nyquist band (though we will give more advanced noise models later in the text).

Define the bit response (symbol response) as $h(t) = p(t) - p(t - T)$. Then utilizing the relationship $B_k = X_k - X_{k-1}$, we can rewrite the channel output in terms of the bit response as

$$Y(t) = \sum_{k=-\infty}^{\infty} X_k \cdot h(t-kT) + N(t). \tag{2}$$

Figure 1 illustrates typical transition and bit responses for the perpendicular magnetic recording channel. Here we consider perpendicular recording channels, and we take $PW50$ to be the width of the *bit response* at 50% of the maximal amplitude. Typically, the bit response is longer that the symbol interval T, which leads to *intersymbol interference*. A crude measure of the level of intersymbol interference is the *normalized density* $\mathcal{D}$ which is defined as the ratio of $PW50$ and T, i.e.

$$\mathcal{D} = \frac{PW50}{T}.$$

C. Discrete-time PAM channel model

A discrete-time channel model is obtained by filtering (equalizing) and sampling (at symbol rate) the continuous-time PAM channel model. We will not go into the details of the various available methods for sampling and equalizing; instead we refer the reader to [1], [2], [3], [4], [5], [6], [7], [8].

The discrete-time channel input is denoted by X_k, while the discrete-time channel output is denoted by Y_k. The discrete-time channel response h_k is assumed to be of *finite duration* I, i.e., $h_k = 0$ for $k < 0$ and $k > I$. The channel input-output relationship is given by

$$Y_k = \sum_{m=0}^{I} X_{k-m} \cdot h_m + N_k, \tag{3}$$

where N_k is assumed to be white Gaussian noise whose variance is $\mathrm{E}N_k^2 = \sigma^2$.

The channel response h_k is often given in polynomial form

$$h(D) = \sum_{i=0}^{I} h_i \cdot D^i.$$

The channel polynomial $h(D)$ is called the *partial response polynomial* [3]. In practice, the chosen channel polynomial $h(D)$ depends on many parameters of the real channel, i.e., normalized density, media noise type, media noise percentage, etc. For these reasons, in the literature, widely used "standard" partial response polynomials for perpendicular recording channels have the generic form $(1+D)^I$.

The key characteristics of the channel is measured by two quantities. The first is the *intersymbol interference length* I. The second is the *signal-to-noise ratio* SNR. In general, the SNR is the ratio of the signal power to the noise power. In some cases, especially in the presence of media noise, it is not straightforward to define the SNR because the noise depends on the signal itself; for definitions of SNRs in media noise environments see [9], [10]. For the channel in (3), there is no media noise, and we can define the SNR in dB (decibels) as

$$SNR = 10 \log_{10} \frac{\sum_{i=0}^{I} h_i^2}{\sigma^2}. \tag{4}$$

D. PAM channel model with media noise

Already in the late 1990's most magnetic recording systems started operating in a media noise dominated regime. This implies the magnetic recording medium cannot simply be assumed to be continuous. Instead, it should be recognized that recording medium is composed of individual grains, where each grain is subject to grain-size as well as coercivity variations. The grain quantization phenomena in turn gives rise to several effects, including 1) nonlinear amplitude loss (NLAL), 2) jitter noise and 3) pulse broadening noise.

The PAM model in (1) can be modified to account for NLAL, jitter noise and pulse broadening noise. We introduce the following notation

- $\gamma\left(B_{k-1}^{k+1}\right) \le 1$ is the amplitude loss factor that models NLAL. The value of the factor γ clearly depends on neighboring transitions, i.e., transitions B_{k-1} and B_{k+1}, which is denoted by $\gamma\left(B_{k-1}^{k+1}\right)$.
- J_k is the jitter noise. The mean of J_k is typically 0, and the variance is σ_J^2. J_k need not be Gaussian, although it is often modeled as Gaussian. A more accurate model can consider J_k to be dependent on, say M, neighboring transitions, say B_{k-M}^{k+M}.
- D_k is a random variable that models random pulse broadening. The mean of D_k is typically 0, and the variance is σ_D^2. To get better model accuracy, D_k can be considered to dependent on, say M, neighboring transitions, say B_{k-M}^{k+M}.

If the values of the media noise random variables J_k and D_k are small, using a truncated Taylor series expansion [11], [12], [9], we get the following channel model

$$\begin{aligned} Y(t) \quad \approx \quad & \sum_{k=-\infty}^{\infty} \gamma\left(B_{k-1}^{k+1}\right) \cdot B_k \cdot [p(t-kT) + J_k q(t-kT) \\ & + D_k s(t-kT)] + N(t). \end{aligned} \tag{5}$$

Here $q(t) = p'(t)$ and $s(t) = t \cdot p'(t)$, where $p'(t) = \mathrm{d}p(t)/\mathrm{d}t$. Figure 2 pictorially shows the model in (5).

E. Discrete-time channel with autoregressive media noise

The model in (5) gives a very good physical explanation of all the nonlinear and media-noise effects that affect the magnetic recording channel at high densities. However, directly designing a detector for the model presented in (5) is difficult. We next give a discrete-time channel model for which it is possible to derive the optimal channel detector [13], [14]. In this model, the assumption is that the signal can depend nonlinearly on the written symbols and that the noise is colored and dependent on a neighborhood of written symbols.

When describing the autoregressive signal-dependent noise model, it is convenient to assume that the written symbols are given in the binary mode A_k (recall that $X_k = (-1)^{A_k}$). The discrete-time readback signal is given as

$$Y_k = \mathcal{Y}\left(A_{k-I_1}^{k+I_2}\right) + N_k. \tag{6}$$

Here, $\mathcal{Y}\left(A_{k-I_1}^{k+I_2}\right)$ is the noiseless channel output that may depend causally and anticausally on a neighborhood of channel inputs $A_{k-I_1}^{k+I_2} = [A_{k-I_1}, A_{k-I_1+1}, \ldots, A_{k+I_2}]$. The channel noise is N_k and is assumed to be correlated and dependent on a neighborhood of written symbols A_k. The noise N_k is modeled as an autoregressive process, i.e., N_k is obtained by filtering white noise through a bank of infinite impulse response (IIR) filters. Each filter has L coefficients that are dependent on a neighborhood of written symbols $A_{k-D_1}^{k+D_2}$. We denote the filter coefficients by $b_i\left(A_{k-D_1}^{k+D_2}\right)$, where $1 \le i \le L$. Using this notation, we can now write the evolution equation for the autoregressive noise process N_k

$$N_k = \sum_{i=1}^{L} b_i\left(A_{k-D_1}^{k+D_2}\right) \cdot N_{k-i} + \sigma\left(A_{k-D_1}^{k+D_2}\right) \cdot W_k, \quad (7)$$

where W_k is unit-variance white Gaussian noise and $\sigma\left(A_{k-D_1}^{k+D_2}\right)$ is a standard deviation factor that is dependent on a neighborhood of written symbols $A_{k-D_1}^{k+D_2}$. Consequently, N_k is Gaussian noise, which may not be an entirely appropriate model, especially in heavily jitter noise dominated channels because jitter (even if it is Gaussian) does not necessarily translate into Gaussian amplitude noise. Figure 3 illustrates the autoregressive model.

The model (6)-(7) is the most general model for which an optimal finite-state detector can be formulated (for detector descriptions, see next section). For example, if we take $L = 0$, $I_1 = 0$, $I_2 = I$, $D_1 = D_2 = 0$, $\sigma\left(A_{k-D_1}^{k+D_2}\right) = \sigma$, and if we set

$$\mathcal{Y}\left(A_k^{k+I}\right) = \sum_{m=0}^{I} (-1)^{A_{k-m}} \cdot h_m = \sum_{m=0}^{I} X_{k-m} \cdot h_m,$$

we indeed get the simpler partial response model (3) as a special case of the autoregressive channel model (6)-(7).

While the autoregressive model is very general and flexible, its drawback is that the coefficients $\mathcal{Y}\left(A_{k-I_1}^{k+I_2}\right)$, $b_i\left(A_{k-D_1}^{k+D_2}\right)$ and $\sigma\left(A_{k-D_1}^{k+D_2}\right)$ are not easily linked to physical parameters such as jitter noise variance, pulse broadening variance, NLAL. However, despite this difficulty in analytically determining the coefficients of the autoregressive model to head-media parameters, it is actually relatively simple to find the autoregressive coefficients from readback waveforms using on-line adaptive signal processing techniques [14], [15].

F. Other Channel Characteristics

So far, we have modeled the magnetic recording channel as an intersymbol interference channel where, due to the magnetic grain properties, the noise, to a large degree, is data dependent. Next, we will discuss four additional characteristics of this channel to illustrate some of the subtleties involved in modeling such systems.

In general, there exists some freedom in choosing the channel symbol interval T. As an example, one option is to introduce no coding and record each bit with interval T. Another alternative is to encode the data sequence with some designed code rate $r < 1$ and record the resulting sequence with symbol interval rT. Both options will occupy the same disk area. Reducing the symbol interval from T to rT will increase the intersymbol interference and degrade the signal-to-noise ratio, but the additional coding will improve data reliability. To be able to analyze and optimize such tradeoffs, it is important to have models that scale well with varying symbol length T.

In addition to the localized noise distributions discussed so far, a second class of impairments, referred to as *burst errors*, occur in magnetic recording channels. For example, the absence of magnetic grains on some sections on the disk surface, and the presence of bumps in the disk surface which collide with the magnetic transducer (reader), are two instances that cause long sequences (or bursts) of bit errors. Channel models that incorporate both the local and burst errors are referred to as a *mixed channels*.

A third class of impairment is the reduction in signal-to-noise ratio due to mechanical fluctuations. For example, a particular adjacent track may overwrite a portion of the sector under interest, or the spacing between disk surface and reader may increase. Both of these events will reduce the signal-to-noise ratio.

Finally, in a double-layer perpendicular channel, the read signal is associated with a narrow DC notch, i.e., a frequency spectrum with a zero amplitude at zero frequency. This narrow frequency notch translates into a long tail of the pulse response in the time domain. The superposition of these undesirable pulse tails, is known as *baseline wander* [16], and is a source of impairment. Usually, baseline wander is handled by the use of circuitry known as *DC loops* [16] and *DC-free codes* [17].

III. Channel Detectors

The basic task of a channel detector is to *detect* the symbols written on the magnetic medium. We consider discrete-time channel models (3) or (6)-(7) when designing the detector. However, it should be understood that before we can even begin processing the readback signal in discrete-time, the operations of filtering, sampling and equalization are necessary, see [1], [2], [3], [4], [5], [7], [8].

The two basic types of detectors are 1) *sequence* detectors and 2) *symbol* detectors. The output of a sequence detector is the estimate of the binary sequence A_k, denoted by $\hat{A}_k \in \{0, 1\}$. The output of a symbol detector is the posterior probability (or estimated posterior probability) of the symbol A_k being either 1 or 0, after observing the entire sequence of channel outputs $\ldots, Y_{k-1}, Y_k, Y_{k+1}, \ldots$.

A. Sequence detection - the Viterbi detector

The Viterbi detector operates on the notion of a *channel state*. For the model in (3), the channel state at time k is a random vector $S_k = A_{k-I+1}^k$. The task of the Viterbi detector is to determine the most likely state sequence $\ldots, S_{k-1}, S_k, S_{k+1}, \ldots$, after observing the entire sequence of channel outputs $\ldots, Y_{k-1}, Y_k, Y_{k+1}, \ldots$. The output of the Viterbi detector is the sequence of state estimates

$\ldots, \hat{S}_{k-1}, \hat{S}_k, \hat{S}_{k+1}, \ldots$, which uniquely specifies the sequence of estimated binary symbols $\ldots, \hat{A}_{k-1}, \hat{A}_k, \hat{A}_{k+1}, \ldots$.

The Viterbi detector is a dynamic programming algorithm [18]. Its complexity is exponentially proportional to the *size* of the state. For the model in (3), the state size is I, so we can say that the complexity of the Viterbi detector is $O(2^I)$. Our task here is not to explain the details of the Viterbi detector. For that, the reader is referred to [19], [20], [7], [21]. We merely concentrate on the key aspects of the Viterbi algorithm and how they depend on the choice of the channel model.

The key component of the Viterbi algorithm is the computation of the *branch metric*. A *branch* is defined as a pair of states (s_{k-1}, s_k), or equivalently as the sequence of binary symbols that determine these two consecutive states, i.e., a_{k-I}^k in binary notation, or x_{k-I}^k in bipolar notation. The branch metric is defined as $\mathcal{M}(s_{k-1}, s_k) = -\ln p(y_k|s_{k-1}, s_k)$, where $p(y_k|s_{k-1}, s_k)$ is probability density function (pdf) of the channel output given the branch. For the channel model in (3) with *Gaussian* noise, the branch metric is

$$\mathcal{M}(s_{k-1}, s_k) = \frac{1}{2}\ln\left(2\pi\sigma^2\right) + \frac{\left(y_k - \sum\limits_{m=0}^{I} x_{k-m} \cdot h_m\right)^2}{2\sigma^2}. \tag{8}$$

At every time instant k, the branch metric (8) needs to be computed for every one of 2^I pairs of possible states, hence the complexity of the Viterbi detectors is proportional to 2^I.

The Viterbi algorithm is also known as a particular example of a class of algorithms known as *min-sum* algorithms. This means that we sum the branch metrics along the time axis, and in the end we pick the path whose sum of branch metrics is the lowest.

If the channel contains media noise of the type given in the autoregressive model (6)-(7), the Viterbi detector is still the optimal sequence detector [22]. However, the state definition and the branch metric definition need to be altered. For the AR model, the state needs to be defined such that it captures the entire neighborhood of signal and noise dependence on the pattern of written symbols A_k. For the AR media noise model, we define the *state* as

$$S_k = A_{k-\max(D_1, I_1+L)+1}^{k+\max(D_2, I_2)}, \tag{9}$$

and the *branch* as

$$(s_{k-1}, s_k) = a_{k-\max(D_1, I_1+L)}^{k+\max(D_2, I_2)}. \tag{10}$$

Hence, we conclude that there is a total of $2^{\max(D_2,I_2)+\max(D_1,I_1+L)}$ possible states, and $2^{\max(D_2,I_2)+\max(D_1,I_1+L)+1}$ branches. Consequently, for each one of the $2^{\max(D_2,I_2)+\max(D_1,I_1+L)+1}$ branches, the *branch metric* is computed as follows [22]

$$\mathcal{M}(s_{k-1}, s_k) = \frac{1}{2}\ln\left[2\pi \cdot \sigma\left(a_{k-D_1}^{k+D_2}\right)^2\right] + \frac{\left(y_k - \mathcal{Y}\left(a_{k-I_1}^{k+I_2}\right) - \sum\limits_{i=1}^{L} b_i\left(a_{k-D_1}^{k+D_2}\right)\left[y_{k-i} - \mathcal{Y}\left(a_{k-I_1-i}^{k+I_2-i}\right)\right]\right)^2}{2\sigma\left(a_{k-D_1}^{k+D_2}\right)^2}. \tag{11}$$

In the branch metric computation (11), the sum $\sum\limits_{i=1}^{L} b_i\left(a_{k-D_1}^{k+D_2}\right)\left[y_{k-i} - \mathcal{Y}\left(a_{k-I_1-i}^{k+I_2-i}\right)\right]$ is implementable by an FIR filter whose coefficients are $b_i\left(a_{k-D_1}^{k+D_2}\right)$. Obviously, the complexity of the Viterbi detector for the AR model is proportional to $2^{\max(D_2,I_2)+\max(D_1,I_1+L)+1}$. However, the real complexity of the detector is greater. Each filter has L filter coefficient $b_i\left(a_{k-D_1}^{k+D_2}\right)$ and each coefficient depends on a pattern of symbols $a_{k-D_1}^{k+D_2}$. This means that there is a total of $L \cdot 2^{D_1+D_2+1}$ FIR filter coefficients. Even for moderate lengths L, D_1 and D_2, the complexity is too high for implementation in hardware. For this reason, several methods for cutting down the complexity at only a mild performance degradation have emerged.

a) Pattern-dependent post processing: is a technique to strike a trade-off between complexity and performance. The basic idea is to use a simple Viterbi detector (for some value $I \leq I_1 + I_2 + 1$) using branch metrics in (8) to preliminarily identify a few (say only 2) most likely symbol sequences, and then use the more complex branch metric (11) in a post-processing fashion to further narrow down the choice to one sequence [23]. This strategy requires the implementation of only a few (say only 2) FIR filters at any given time instant k.

b) Filter and pattern-truncation: is another technique to lower the complexity of the detector [22], [24]. The idea is to force a shorter filter length L or to force shorter pattern-dependence lengths D_1, D_2, I_1 and I_2. Obviously, this will degrade the performance of the detector, but if done properly, depending on the channel/noise properties, the performance degradation may not be severe and the complexity savings considerable.

B. Symbol detection - the BCJR detector

The BCJR algorithm [25] finds the following posterior probability $\tilde{A}_k = \Pr(A_k = 1|\ldots, y_{k-1}, y_k, y_{k+1}, \ldots)$. Here, we can interpret $\tilde{A}_k$ as the *soft* bit. The closer $\tilde{A}_k$ is to 1, the higher the certainty that A_k indeed equals 1, and vice versa, the closer $\tilde{A}_k$ is to 0, the higher the certainty that $A_k = 0$. The advantages of soft detection are not immediately obvious if the task is simply to determine the written symbol A_k. However, one must keep in mind that the written symbols A_k represent *encoded* user data. The advantages of the soft detector become obvious only if one considers powerful soft decoders of error correction codes.

The computational complexity of BCJR algorithm is roughly two times the Viterbi algorithm because it requires two passes through the readback waveform (one in the forward direction, and the other in the reverse direction)[1]. We note that the implementation of the full-blown BCJR algorithm requires

[1]We note that forward-only implementations of the BCJR detector are also possible [26], but they are generally more complex than forward-backward implementations, and should be used only if a low-delay, high-throughput is required.

a prohibitively large buffer memory, but that this requirement can be relaxed by using appropriate *sliding-window* BCJR algorithms [27].

The general apparatus for building the BCJR detector is roughly the same as for the Viterbi detector: states and branches are defined in the same manner. The BCJR branch metric is easily obtained as the modification of the Viterbi branch metric, i.e., the BCJR branch metric is

$$\mathcal{B}(s_{k-1}, s_k) = \mathrm{e}^{-\mathcal{M}(s_{k-1}, s_k)}. \tag{12}$$

The only substantial difference between the BCJR algorithm and the Viterbi algorithm is that the BCJR algorithm is a *sum-product* algorithm (whereas the Viterbi detector is a min-sum algorithm). Being a sum-product algorithm means that in the BCJR algorithm the branch metrics are multiplied along the time axis, and in the end they are summed.

It is worthy of mentioning that suboptimal soft detection can also be performed by the *soft-output Viterbi algorithm* (SOVA) [28] which uses a min-sum structure to provide an estimate for the posterior symbol probabilities. SOVA typically has implementation advantages over the BCJR algorithm, but there may be a performance loss if a soft error correction decoder uses SOVA instead of BCJR.

When designing soft-decision detectors for the AR model in (6)-(7), detector complexity is always an issue, just as it is for the corresponding Viterbi detector. Hence, suboptimal low-complexity detector designs are typically used for practical implementation. As already mentioned, these suboptimal designs can be based on post-processing strategies [23] and state reduction techniques [22], [24].

C. Future of channel detectors

Recently, Pighi et.al [12] proposed a detector that is optimized to handle several orthogonal media noise components. The main assumption made is that the noise values are not severe so that a first-order Taylor series expansion, such as the one in (5), models the noise accurately. The resulting detector relies on one matched filter per noise component (resulting in several samples per symbol), and takes a familiar Gaussian branch metric form when the noise is assumed to be Gaussian. However, as we look towards the future, jitter noise will be larger and more dominant, and hence Taylor series expansions may no longer be appropriate approximations. The main challenge for the future then remains the design of a detector for non-Gaussian noise sources, specifically for jitter noise dominated channels.

IV. Channel Capacity

The channel capacity of even the simplest binary intersymbol interference (ISI) channel (3) is still not known in closed form. However, a major achievement over the past decade has been the creation of simulation-based Monte Carlo methods that numerically estimate channel information rates and extremely tightly bound the channel capacity.

The channel capacity C is the upper limit of the number of bits per channel symbol T (bits per channel use) that one can pass through the channel (i.e., store on the magnetic medium) at an arbitrarily low probability of reception (readback) error [29]. Conversely, if one attempts to transmit (write) more bits per channel use than the channel capacity, it is not possible to receive (retrieve) data with an arbitrarily low probability of error. Hence, evaluating the channel capacity is important to channel engineers since it provides an upper bound on the code rate of the *error correction code* that needs to be utilized. Note that in the absence of noise, the channel capacity of a magnetic recording channel is 1 bit/channel-use. Of course, noisy channels will have channel capacities strictly lower than 1 bit/channel-use.

The channel information rate is the difference between the channel output entropy rate $h(Y)$ and the conditional channel output entropy rate $h(Y|X)$, given the channel input X

$$\mathcal{I}(X;Y) = h(Y) - h(Y|X). \tag{13}$$

The entropy rate $h(Y)$ is the limit of the average differential entropy $h(Y_1^n|S_0)$

$$\begin{aligned} h(Y) &= \lim_{n\to\infty} \frac{1}{n} h(Y_1^n|S_0) \\ &= -\lim_{n\to\infty} \frac{1}{n} \mathrm{E} \log_2 p(Y_1^n|S_0), \end{aligned} \tag{14}$$

where S_0 is the initial state. For typical data storage channels, the choice of the initial state does not affect the entropy rate $h(Y)$. Similarly, the conditional entropy rate is

$$\begin{aligned} h(Y|X) &= \lim_{n\to\infty} \frac{1}{n} h(Y_1^n|X_1^n, S_0) \\ &= -\lim_{n\to\infty} \frac{1}{n} \mathrm{E} \log_2 p(Y_1^n|X_1^n, S_0). \end{aligned} \tag{15}$$

In some cases, the conditional entropy $h(Y|X)$ is straight forward to evaluate. For example, for the Gaussian channel in (3), the conditional entropy rate is expressible in closed form $h(Y|X) = \frac{1}{2}\log_2\left(2\pi\mathrm{e}\sigma^2\right)$. In other cases, $h(Y|X)$ must be numerically evaluated.

The information rate $\mathcal{I}(Y;X)$ is dependent on the statistics of the channel input sequence X_k. Perhaps the most useful channel information rate is the rate $\mathcal{I}_{\text{i.u.d.}} \le C$ obtained when the channel input sequence is constrained to be independent and uniformly distributed (i.u.d) random variables, i.e., $\Pr(X_k = -1) = \Pr(X_k = 1) = 1/2$. The rate $\mathcal{I}_{\text{i.u.d.}}$ is also know as the *symmetric information rate* because of the obvious symmetry between symbols 1 and -1.

If the channel input sequence is i.u.d., or memoryless, or generally, if the channel input sequence exhibits *Markov* memory, the information rate can be computed using a Monte Carlo method that was recently devised independently by three groups of authors [30], [31], [32]. The method relies on the BCJR algorithm to compute the following posterior probability density function

$$p(y_1^n|s_n)$$

for some known Markov probability mass function that generates a long sequence of channel inputs $x_1, x_2, \cdots, x_n$. Then by applying the generalized Shannon-McMillan-Briemann theorem [33], the sample entropy rate converges with probability 1 (w.p.1) to the true entropy rate of the channel output $h(Y)$

$$-\frac{1}{n}\sum_{s_n}\log_2 p(y_1^n|s_n) \overset{\text{w.p.1}}{\longrightarrow} h(Y). \tag{16}$$

The key here is to pick n to be very long (say $n = 10^6$) so that the value on the left-hand side is sufficiently close to $h(Y)$. Then, for the channel model in (3), the information rate can be estimated from (13)

$$-\frac{1}{n}\sum_{s_n}\log_2 p(y_1^n|s_n) - \log_2\left(2\pi \mathrm{e}\sigma^2\right) \overset{\text{w.p.1}}{\longrightarrow} \mathcal{I}(X;Y). \tag{17}$$

If the channel input is chosen to be i.u.d., then the quantity in (17) is the estimate of $\mathcal{I}_{\text{i.u.d}}$. The rate $\mathcal{I}_{\text{i.u.d.}}$ is the maximal rate attainable by a random linear code such that the probability of error can be made arbitrarily close to zero [34]. Very often, in magnetic recording channels, the rate $\mathcal{I}_{\text{i.u.d}}$ is considered to be the *practical* channel capacity. The reason for this is that with target code rate r, say $r \approx 0.9$ which is common in magnetic recording channel, bits/channel-use, $\mathcal{I}_{\text{i.u.d}}$ is almost equal to the channel capacity.

Now, if we want to compute the actual channel capacity, we must solve the following optimization problem

$$C = \max \mathcal{I}(X;Y), \tag{18}$$

where the maximization is taken over all possible distributions of stationary input processes X_k. This is a very difficult optimization problem for which no closed-form solution is known. However, we can define the channel capacity if the channel input is a Markov process of order M as

$$C_M = \max \mathcal{I}(X;Y), \tag{19}$$

where the maximization is taken over all stationary Markov processes of order M. Clearly $C_M \leq C_{M+1} \leq C_{M+2} \leq \cdots \leq C$. Recently, it was proved that Markov processes *asymptotically* achieve capacity [35], i.e.

$$\lim_{M\to\infty} C_M = C. \tag{20}$$

This suggests a very simple strategy: numerically compute C_M for large values of M to get ever tightening lower bounds on the capacity C. Indeed this strategy delivers very tight lower bounds on the capacity. The numerical optimization can be conducted using gradient search techniques [31], or an efficient generalization of the Blahut-Arimoto algorithm adapted to Markov channel input processes [36], [37].

Of course, to claim that a lower bound is tight, we must also compute equally tight upper bounds. Fortunately, tight numerical upper-bounding techniques have also been devised in the past several years [38], [39]. The tightest known numerical upper bounds are obtained by computing the *feedback* capacity, or the *delayed feedback* capacity [39]. The feedback capacity is the channel capacity when the receiver sends feedback back to the transmitter. Of course, in data storage channels, the receiver (reader) cannot provide feedback to the transmitter (writer) because the reader cannot send data from the future (time of reading) to the past (time of writing). Nonetheless, the concept of *feedback* can be used to mathematically/numerically obtain upper bounds on the capacity. The more we *delay* the feedback, the tighter the upper bound. Thus, we can consider that the capacity computation problem has been numerically solved.

In many instances, even numerical Monte Carlo computations of the i.u.d. information rate $\mathcal{I}_{\text{i.u.d.}}$ or the channel capacity C (i.e., its lower and upper bounds) are prohibitively complex, especially for the AR channel model whose state size variables L, D_1, D_2, I_1 and I_2 are large. In such cases, we can resort to several techniques to lower the computational cost at the expense of loosening the bounds [40], [41]. Often, however, we can design techniques that only *slightly* loosen the bounds but achieve substantial computational savings. Here we mention two such techniques.

c) Auxiliary channel techniques: rely on establishing an auxiliary channel whose channel state is *much shorter* that the given channel [42]. In that case, we can process the channel outputs of the given channel using the computationally favorable auxiliary channel to obtain bounds on the channel information rate. The bounds will be tight if the auxiliary channel is a "close" approximation of the given channel. For example, if we have a non-finite-state channel $1/(1+0.2D)$, we can "closely" approximate it by an auxiliary finite-state channel $1 - 0.2D + 0.2^2D^2$, and apply auxiliary channel techniques to bound the channel information rate.

d) State reduction techniques: can also be used to obtain upper and lower bounds on the information rate [42]. The idea is to either drop "insignificant" states from the computation algorithms, or to group insignificant states into a super-state. The tightness of the bounds will depend on how many insignificant states are dropped or grouped. An application of this principle to the AR channel model is given in [43].

V. Reed Solomon and LDPC Codes

Present disk drives specify their mean time between failures at around 10^{15} bits. Since drives inherently cannot store and retrieve magnetized patterns at such high reliability, they depend on powerful *error correcting codes* to deliver this requirement.

Consider an (n, k) *error correcting code* with block length n symbols, k message (user) symbols, $n - k$ parity symbols, and an associated code rate of $r = k/n$. Also, consider a magnetic recording channel, with symbol interval T that is associated with a channel capacity C . Then, to attain very low failure rates, any *error correcting code* has to be have a code rate $r < C$. The closer r is to C the better the code, since less parity bits and therefore more message bits can be stored on a disk surface.

There are two practical obstacles in attaining channel capacity with an *error correcting code*. First, to achieve channel capacity, n must get very large ($n \to \infty$) which is not feasible in a disk drive where the block length n is fixed and predetermined. Also, the complexity of an optimal decoder increases exponentially with block length n. In general, more complex decoders will need more silicon area and increase the cost of a disk drive.

Therefore, the design of an desirable *error correcting code* is a tradeoff between maximizing user density, and minimizing decoder complexity under the constraints of a fixed block length n and a given reliability specification.

In this section we will discuss two prominent families of error correcting codes, *Reed-Solomon codes* and *low density parity-check* (LDPC) codes. Coincidentally, both of these codes were published in 1960 [44], [45]. In some architectures, these codes compete with each other while in others they cooperate. In Figure 4, the decoder strategies are given for a Reed-Solomon decoder, an LDPC decoder and a two-level decoder with an inner LDPC code and an outer Reed-Solomon code.

In the disk drive industry, where areal densities have increased exponentially over several decades, and where usually no single technology prevails for more than a few years, Reed-Solomon codes have survived for more than a quarter of a century! However, in the last decade, LDPC codes have started to attract attention [46] [47]. In this section, we review the strengths and the weaknesses of both codes and discuss possible code configurations that a disk drive may use in the future.

A. Reed-Solomon codes

An $(n, k) = (2^m - 1, k)$ *Reed-Solomon code*, with symbol length $m = \log_2(n + 1)$, block length of n symbols ($n \cdot m$ bits), and k information symbols ($k \cdot m$ bits), has a *minimum distance* of $d = n - k + 1$ symbols. Hence, such a code can correct an arbitrary number of t symbol errors, as long as

$$t \leq \left\lfloor \frac{n-k+1}{2} \right\rfloor . \qquad (21)$$

The *Singleton bound* [48] states that no (n, k) code can guarantee to correct more symbol errors than the right hand side of (21). Codes, like the *Reed-Solomon code*, that achieve this bound, are called *maximum distance separable* (MDS) codes. In the decade following the appearance of *Reed-Solomon codes* [44], efficient algebraic decoding algorithms were developed to decode *Reed-Solomon codes* [49].

As was mentioned in Section II, the magnetic recording channel is a *mixed channel*, where both local errors and burst errors can coexist. In such channels, *Reed-Solomon codes* have shown to be very robust since they are based on symbols rather than bits.

In summary, *Reed-Solomon codes* (1) are symbol based codes with a robust performance in a mixed channel, (2) are one of the few existing MDS codes, and (3) algebraic decoding algorithms exist to decode them efficiently.

Unfortunately, the elegant and efficient algebraic decoders are also the source of some major limitations. First, symbol based decoders are not powerful in the presence of bit errors. From (21), we need two redundant symbols to correct one erroneous bit. For example, if the symbol size is $m = 12$, then 24 redundant bits need to be consumed in order to correct a single bit error. Second, the algebraic decoders are *bounded distance* decoders which means the decoder does not even attempt to correct blocks that suffer more than t errors. Finally, being algebraic in nature, these decoders are not designed to handle the soft detector outputs $\{\tilde{A}_k\}$.

The past decade gave rise to several improved decoders of Reed-Solomon codes. We classify them into three groups: 1) algebraic bivariate-polynomial interpolation-based decoders, 2) iterative decoders, and 3) ordered statistics decoders.

The Guruswami-Sudan decoder [50], [51] was the first decoder to demonstrate decoding capabilities beyond the half-minimum-distance decoding radius. It used a novel bivariate polynomial interpolation technique to achieve this. Later, Koetter and Vardy [52] adapted this technique to perform *soft* algebraic decoding. The bivariate polynomial interpolation technique can also be used to perform low complexity Chase (LCC) decoding [53], or bit-level generalized minimum distance (BGMD) decoding [54]. The complexity of bivariate polynomial interpolation-based algebraic decoders can vary from very low (e.g., LCC for moderately low Chase parameters) to extremely high (e.g., Koetter-Vardy used with high *multiplicity*), but the gains remain modest when compared to the Berleykamp-Massey algebraic decoder [49].

Better gains can be achieved using iterative decoding techniques. The first such decoder, named adaptive belief propagation (ABP), was proposed by Jiang and Narayanan [55], [56]. Its drawback is the complexity of the Gaussian elimination that is required in each iteration. Subsequently, Bellorado et. al [57] eliminated the need for Gaussian eliminations, thus drastically reducing the complexity.

Nearly optimal performance can be achieved by the enhanced box and match algorithm (eBMA) proposed by Jin and Fossorier [58]. The eBMA method is the latest step in the evolution of ordered statistics decoders (OSDs) [59], [60], and presently it is the best-performing known decoder of Reed-Solomon codes. Its disadvantage is its very high complexity which allows software implementations but prohibits hardware implementations using technologies available today.

B. LDPC Codes

As discussed earlier, decoder complexity, along with the block length constraint, is the major obstacle to achieving channel capacity. The codes invented by Gallager [45] were specifically designed to be decoded with reasonable complexity. These codes are known as LDPC codes and the associated decoders are known as *iterative decoders*. *Iterative decoders* address all three limitations of the algebraic *Reed Solomon* decoder [49] and at large block length n can practically achieve the rate $r \approx \mathcal{I}_{\text{i.u.d}}$ [61], [62]. However, as we will see shortly iterative decoders have their own limitations.

Any (n, k)-linear error correcting code can be described by an $(n - k) \times n$ matrix known as the *parity-check matrix*. An LDPC code can be associated with a *sparse* (or low-density) *parity check matrix*. The sparseness in a parity-check matrix allows iterative decoder algorithms to become more effective and at the same time reduces decoder complexity. *Iterative decoders* locally process the soft detector output $\tilde{A}_k$. In other words, at any given iteration, each of the $(n-k)$ parity constraints operate independent of each other. This is unlike an optimal *maximum likelihood* decoder that simultaneously satisfies all parity constraints. An iterative decoder performs multiple iterations until all the parity constraints are satisfied. These iterations, within the LDPC decoder, are referred to as *local* iterations. In a magnetic recording channel, we also perform *global* iterations, where information is transferred between the LDPC decoder and the soft-output detector, see Figure 4(b).

The reduction in complexity of an *iterative decoder* comes at the expense of some performance loss when compared to the maximum likelihood decoder, especially at small block lengths. Furthermore, an iterative decoder may not converge into a valid codeword even if we iterate indefinitely. At high SNR, this phenomenon usually involves only a few bits in the block. The bits that do not converge are referred to as *trapping sets* [63]. As a result, at the high SNR region, the SNR versus codeword failure rate curve flattens. This flattening of the SNR/codeword failure rate curve is known as the *error floor*. There is a particular concern in the magnetic recording channel, where, as mentioned earlier, the mean time between failures is very large at around 10^{15} bits. Lowering the error floor is a difficult problem, since software simulators can not generate enough error statistics at the SNR region of interest [64].

Despite significant effort consumed in generating structured and therefore lower complexity LDPC codes [65], [66], [67], *iterative decoders* use up significant silicon area, especially when compared to algebraic *Reed-Solomon* decoders. This is due to the logic needed to perform the parallelized repetitive operations, and to the large buffer requirements to hold all the intermediary soft values.

Furthermore, LDPC codes do not have the nice MDS properties of *Reed-Solomon codes*. Hence, in the presence of a burst error LDPC codes may become more vulnerable than *Reed-Solomon codes* [68].

In short, iterative decoders, by overcoming the limitations of the algebraic *Reed-Solomon* decoders, have delivered significant performance gain. However, such gains have in turn created new challenges, including (1) decoder complexity, (2) error-floors and (3) robustness in the presence of burst channels.

Before ending this discussion, we will mention a third option where both LDPC and Reed-Solomon codes are used together, as shown in Figure 4(c) [69]. This is a compromise solution, that delivers some of the gain of the LDPC codes without the need to address the major LDPC challenges discussed above.

In summary, we have identified three very active research areas in coding theory, along with their promises and challenges. The first is to improve the *Reed-Solomon* decoder, the second is the replacement of *Reed-Solomon* with LDPC codes, and finally, the third is to use both codes.

VI. DATA ACCESS

Information is recorded on a disk using a *writer*, while information is retrieved from a disk using a *reader*. A writer is a transducer that transforms an electric signal into magnetic fields which in turn magnetize grains on the disk in a certain direction. Similarly, a reader senses the specific direction of magnetization from grains and generates an associated electric signal. A *head* denotes either a writer or a reader. A head operates while a disk is rotating around a spindle. In this way, a data stream is recorded in concentric circles around a disk.

To be able to successfully interact with the disk surface, the drive should be able to position the head to the desired coordinates. More specifically, using *cylindrical coordinates*, associated with every bit, there is an ideal coordinate pair (ρ_0, ϕ_0) on a disk. Furthermore, if the desired head-disk spacing is z_0, then the head's desired coordinates become (ρ_0, ϕ_0, z_0). *Servo* algorithms are responsible in bringing the head to the desired radial location ρ_0. *Synchronization* algorithms are responsible for bringing the head to the desired angle ϕ_0. Finally, *fly-height control* algorithms are responsible for keeping the head at some constant clearance z_0 from the disk. We should clarify that a disk drive adjusts the head to modify ρ and z but variations in ϕ are obtained by the rotating disk.

To arrive at any desired location (ρ_0, ϕ_0, z_0), the current coordinates of the head need to be determined. Here, we limit ourselves to techniques that use the same reader, that senses data information, to also sense magnetic fields that provide coordinate information. A disk drive applies three technical areas of *signal design* [70], *estimation theory* [71] and *control theory* [72] to access the desired bit location.

The recorded pattern design that reveals positional information is referred to as *signal design*. At the output of a reader, a signal design is transformed to a noisy signal, which in turn is used to obtain an *estimate* of the reader coordinates. Furthermore, the region on the disk that reveals positional information, should be minimized since this area comes at the expense of data storage. Usually, established *estimation theory* algorithms can be used to obtain head position (ρ, ϕ, z) from the retrieved noisy signal. Finally, using *control theory*, the appropriate force is applied to the head, along the ρ and z directions, to arrive at (ρ_0, ϕ_0, z_0).

In the current literature, estimating and controlling each of the coordinates comes under different areas of research. Here, we take a unified approach, and summarize this whole technical field in TABLE I, where to access any of the three coordinates, in general we need to apply three technical fields.

A. Servo

From TABLE I, the radial position is estimated using two different signal designs [70]. The *Gray code* pattern encodes the track number which is a coarse estimate of ρ, whereas the *servo burst* pattern encodes the offtrack information, i.e. estimates ρ with higher resolution. A servo burst signal can encode radial information in the phase, frequency or amplitude content of a periodic pattern. For further discussion on the topic, the interested reader is referred to [73]. In general, the control algorithms are different when the head is seeking a new track than when the head is already on track. Again, we refer the interested reader to [72] for detailed discussions on control theory in general and to the control of the radial distance of disk drives, in particular.

TABLE I
THREE TECHNICAL AREAS FOR EACH OF THE THREE COORDINATES

	signal design	*estimation*	*control*
ρ	Gray code,	detector,	seek,
	servo burst	matched filters	track
ϕ	global clock,	PLL	-
	frame synchronization,	detector	-
	acquisition field	PLL	-
	data	PLL /detector	-
z	fly-height pattern	harmonic sensor	-

B. Fly-height control

The motivation to keep the fly-height at a desired spacing z_0 is to find a compromise between retaining a strong magnetic field interaction between head and disk on one side and head integrity on the other. At large fly heights, the magnetic field generated by the writer becomes less effective and the reader resolution decreases. On the other hand, at too small head-disk spacing, the integrity of the head is compromised. The magnetic spacing loss can be represented by a low-pass filter [74],

$$H(\omega) = e^{-\frac{\Delta z|\omega|}{v}} \tag{22}$$

where $z = z_0 + \Delta z$, v is the head/surface velocity and ω is the angular frequency of the periodic waveform. A *harmonic sensor* can estimate the fly-height information by exploiting the fact that the filter (22), which is applied to the read-back signal, attenuates the fundamental and the harmonics of the received signal with different weights [75].

C. Synchronization

As shown in TABLE I, up to four different strategies can be applied to estimate the angular position ϕ of the head. It is extremely difficult to record bits at an exact coordinate ϕ_0 on a disk. The uncertainty is, in general, larger than the bit interval itself. This is partly due to the fact that a disk drive can not read while writing. Furthermore, due to temperature variations the reader to writer spacing dynamically changes. As a consequence, additional synchronization fields known as *acquisition field* and *frame synchronization field* [76] are recorded as a preamble to a block of data bits. Both of these methods are borrowed from data communications [77].

In an acquisition field, a periodic pattern, also known as preamble pattern $s(t)$ is recorded, where the period is chosen to maximize the mean square of the signal slope, or $E[s'(t)^2]$. Ignoring data dependent noise, the higher $E[s'(t)^2]$ is, the more accurately we can estimate the phase of $s(t)$. However, since $s(t)$ is periodic, we can estimate the phase of $s(t)$ only within a period. This ambiguity can be removed by *frame synchronization*, which in contrast to a preamble, is a non-periodic pattern with good autocorrelation properties. For a recent development on placement of preambles, see [78].

A crucial and evolving area of research is timing extraction from the data field itself. Comparison between a communications and a magnetic recording channel in this area is informative. A magnetic recording channel can be modeled as a communications channel where (1) no carrier synchronization is needed, (2) the alphabet size is restricted to be binary, (3) no over-sampling is performed, due to the high data rate requirements, (4) the channel can be described as an intersymbol interference channel, (5) *decision directed* synchronization algorithms are used where detector decisions influence phase error estimates, and finally (6) *Modulation codes* are used to improve the timing content in the data.

Recently several novel synchronization techniques, on the data field, were proposed [79], [80], [81], [82], [83] . Some of these techniques are evolutionary changes, see [79], [82], where *Kalman filtering* technique on the noise model (6)-(7) is applied to improve synchronization performance. Other approaches are more revolutionary. For example in [80], [81] powerful error correcting codes with iteratively decoders are used to aid the synchronization process. Finally, in [83], [81] a joint trellis is constructed that models both the intersymbol interference and the quantized phase of the waveform. This joint trellis then is processed by a Viterbi or BCJR detector. The challenge of course is to deliver these performance gains at minimum cost. Synchronization remains an active research topic since the ever decreasing SNR values will present a challenge to the existing synchronization methods and require the implementation of advanced synchronization techniques.

We end this section by mentioning that the synchronization algorithms discussed so far handle each block independently, where for each block, first a phase and frequency estimates from the acquisition field is obtained, then from a frame synchronization pattern the start of the block is identified. Finally, the timing variations are tracked from the data field itself. In, [84] synchronization fields are recorded independent of the data blocks. Hence, a timing synchronizer could retain a frequency lock amongst all sectors. With such a global architecture, the frequency uncertainty associated with each block is reduced.

VII. CONCLUSION

Our concise summary of what we believe constitutes the state of the art in data storage channels is as follows.

- Accurate, but also efficient channel modeling is an important prerequisite for understanding, and subsequently proposing and successfully evaluating various channels design architectures.
- Detector designs and capacity estimation methods have matured in the last decade. However, if noise statistics evolve to have non-Gaussian distributions, further improvements will be required in both of these areas.
- Compact signal designs, adaptive and more sophisticated data access algorithms (servo algorithms, fly-height adjustment methods and synchronizers) will continue to evolve in order to accommodate ever more ambitious data rates, areal densities and SNR demands.
- The greatest uncertainty and anticipation is in the area of error correcting codes where one or more of the three different architectures reviewed in this paper (or even a yet unknown code/decoder), will survive as the ultimate error correction code for data storage.

REFERENCES

[1] E. A. Lee and D. G. Messerschmitt, *Digital Communications*. Boston: Kluwer, 1994.

[2] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 4th ed., 2000.

[3] H. K. Thapar and A. M. Patel, "A class of partial response systems for increasing storage density in magnetic recording," *IEEE Trans. Magn.*, vol. MAG-23, pp. 3666–3668, Sept. 1987.

[4] D. J. Tyner and J. G. Proakis, "Partial response equalizer performance in digital magnetic recording channels," *IEEE Transactions on Magnetics*, vol. 29, pp. 4194–4208, 1993.

[5] J. M. Cioffi, W. L. Abbott, H. K. Thapar, C. M. Melas, and K. D. Fisher, "Adaptive equalization in magnetic-disk storage channels,," *IEEE Communications Magazine*, pp. 14–29, Feb. 1990.

[6] J. W. Bergmans, *Digital Baseband Transmission and Recording*. Boston: Kluware Academic Publishers, 1996.

[7] J. G. Proakis, "Partial response equalization with application to high density magnetic recording channels," in *Coding and Signal Processing for Magnetic Recording Systems* (B. Vasic and E. M. Kurtas, eds.), ch. 8, CRC Press, 2005.

[8] P. M. Aziz, "Adaptive equalization architectures for partial response channels," in *Coding and Signal Processing for Magnetic Recording Systems* (B. Vasic and E. M. Kurtas, eds.), ch. 28, CRC Press, 2005.

[9] J. Moon, "Modeling the recording channel," in *Coding and Signal Processing for Magnetic Recording Systems* (B. Vasic and E. M. Kurtas, eds.), ch. 5, CRC Press, 2005.

[10] X. Yang and E. Kurtas, "Signal and noise generation for magnetic recording channel simulations," in *Coding and Signal Processing for Magnetic Recording Systems* (B. Vasic and E. M. Kurtas, eds.), ch. 6, CRC Press, 2005.

[11] T. Oenning and J. Moon, "Modeling the Lorentzian magnetic recording channel with transition noise," *IEEE Transactions on Magnetics*, vol. 37, January 2001.

[12] R. Pighi, R. Raheli, and U. Amad, "Multidimensional signal processing and detection for storage systems with data-dependent transition noise," *IEEE Transactions on Magnetics*, vol. 42, pp. 1905–1916, July 2006.

[13] A. Kavčić and A. Patapoutian, "A signal-dependent autoregressive channel model," *IEEE Trans. Magn.*, vol. 35, pp. 2316–2318, September 1999.

[14] J. Stander and A. Patapoitian, "Performance of a signal-dependent autoregressive channel model," *IEEE Trans. Magn.*, vol. 36, pp. ?–?, Sep. 2000. to appear.

[15] J. J. Ashley and H. J. Stockmanns, *Method and Apparatus for Calibrating Data-Dependent Noise Prediction*. Infineon Technologies, AG, May 2005. US Patent no. 6,889,154 B2.

[16] A. Patapoutian, "Baseline wander compensation for the perpendicular magnetic recording channel," *IEEE Transactions on Magnetics*, vol. 40, pp. 235–241, Jan. 2004.

[17] S. Denic and B. Vasic, "Spectrum shaping codes," in *Coding and Signal Processing for Magnetic Recording Systems* (B. Vasic and E. M. Kurtas, eds.), ch. 19, CRC Press, 2005.

[18] D. P. Bertsekas, *Dynamic Programming and Optimal Control (Vol. I, 2nd Edition)*. Belmont, MA: Athena Scientific, 2001.

[19] G. D. Forney Jr., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Transactions on Information Theory*, vol. 18, pp. 363–378, March 1972.

[20] G. D. Forney Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, March 1973.

[21] M. Despotovic and V. Šenk, "Data detection," in *Coding and Signal Processing for Magnetic Recording Systems* (B. Vasic and E. M. Kurtas, eds.), ch. 32, CRC Press, 2005.

[22] A. Kavčić and J. M. F. Moura, "The Viterbi algorithm and Markov noise memory," *IEEE Trans. Inform. Theory*, vol. 46, pp. 291–301, Jan. 2000.

[23] G. Burd and Z. Wu, *Detection in the Presence of Media Noise*. Marvell International Ltd., August 2005. US Patent no. 6,931,585 B1.

[24] J. Moon and J. Park, "Pattern-dependent noise prediction in signal-dependent noise," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 730–43, April 2001.

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

[26] X. Ma and A. Kavčić, "Path partitions and forward-only trellis algorithms," *IEEE Trans. Inform. Theory*, pp. 38–52, January 2003.

[27] S. Benedetto, D. D., M. G., and P. F., "Algorithm for continuous decoding of turbo codes," *Electronics Letters*, vol. 32, pp. 314–315, February 1996.

[28] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. IEEE Global Telecommunications Conference*, (Dallas, TX), pp. 1680–1686, November 1989.

[29] C. E. Shannon, "A mathematical theory of communications," *Bell Systems Technical Journal*, vol. 27, pp. 379–423 (part I) and 623–656 (part II), 1948.

[30] D. Arnold and H.-A. Loeliger, "On the information rate of binary-input channels with memory," in *Proceedings IEEE International Conference on Communications 2001*, (Helsinki, Finland), pp. 2692–2695, June 2001.

[31] H. D. Pfister, J. B. Soriaga, and P. H. Siegel, "On the achievable information rates of finite state ISI channels," in *Proceedings IEEE Global Communications Conference 2001*, (San Antonio, Texas), November 2001.

[32] V. Sharma and S. K. Singh, "Entropy and channel capacity in the regenerative setup with applications to Markov channels," in *Proc. IEEE ISIT 2001*, (Washington, DC), p. 283, June 2001.

[33] A. R. Barron, "The strong ergodic theorem for densities: Generalized Shannon-McMillan-Breiman theorem," *Ann. Probab.*, vol. 13, pp. 1292–1303, November 1985.

[34] A. Kavčić, X. Ma, and M. Mitzenmacher, "Binary intersymbol interference channels: Gallager codes, density evolution and code performance bounds," *IEEE Transactions on Information Theory*, vol. 49, pp. 1636–1652, July 2003.

[35] J. Chen and P. H. Siegel, "Markov processes asymptotically achieve the capacity of finite-state intersymbol interference channels," in *Proceedings of IEEE International Symposium on Information Theory*, (Chicago, IL), June 2004.

[36] A. Kavčić, "On the capacity of Markov sources over noisy channels," in *Proceedings IEEE Global Communications Conference 2001*, (San Antonio, Texas), November 2001. available at http://hrl.harvard.edu/˜kavcic/publications.html.

[37] P. Vontobel, "A generalized Blahut-Arimoto algorithm," in *Proceedings of IEEE International Symposium on Information Theory*, (Chicago, IL), p. 53, June 2004.

[38] P. Vontobel and D. M. Arnold, "An upper bound on the capacity of channels with memory and contraint input," in *IEEE Information Theory Workshop*, (Cairns, Australia), September 2001.

[39] S. Yang, A. Kavčić, and S. Tatikonda, "The feedback capacity of finite-state machine channels," *IEEE Trans. Inform. Theory*, vol. 51, pp. 799–810, March 2005.

[40] Z. Zhang, T. M. Duman, and E. M. Kurtas, "Information theory of magnetic recording channels," in *Coding and Signal Processing for Magnetic Recording Systems* (B. Vasic and E. M. Kurtas, eds.), ch. 12, CRC Press, 2005.

[41] S. Yang and A. Kavčić, "Capacity of partial response channels," in *Coding and Signal Processing for Magnetic Recording Systems* (B. Vasic and E. M. Kurtas, eds.), ch. 13, CRC Press, 2005.

[42] D. Arnold, H.-A. Loeliger, P. O. Vontobel, A. Kavčić, and W. Zeng, "Simulation-based computation of information rates for channels with memory," *IEEE Transactions on Information Theory*, vol. 52, pp. 3498–3508, Aug. 2006.

[43] R. Pighi, R. Raheli, and F. Cappelletti, "Information rates of multidimensional front-ends for digital storage channels with data-dependent transition noise," *IEEE Transactions on Magnetics*, vol. 42, pp. 2218–2228, September 2006.

[44] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *SIAM Journal of Applied Mathematics*, vol. 8, pp. 300–304, 1960.

[45] G. Gallager, Robert, *Low-Density Parity-Check Codes*. The MIT Press, 1963.

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

[47] W. E. Ryan, "An introduction to ldpc codes," in *Coding and Signal Processing for Magnetic Recording Systems* (B. Vasic and E. M. Kurtas, eds.), ch. 36, CRC Press, 2005.

[48] R. Singleton, "Maximum distance q-nary codes," *IEEE Trans. Inform. Theory*, vol. 10, pp. 116–118, 1964.

[49] E. Berlekamp, *Algebraic Coding Theory*. Mc-Graw Hill, 1968.

[50] M. Sudan, "Decoding of Reed-Solomon codes beyond the error-correction bound," *Journal of Complexity*, vol. 13, pp. 180–193, 1997.

[51] V. Guruswami and M. Sudan, "Improved decoding of reed-solomon codes and algebraic-geometry codes," *IEEE Transactions on Information Theory*, vol. 45, pp. 1757–1767, September 1999.

[52] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Transactions on Information Theory*, vol. 49, pp. 2809–2825, 2003.

[53] J. Bellorado and A. Kavčić, "A low-complexity method for Chase-type decoding of Reed-Solomon codes," in *IEEE Intrnational Symposium on Information Theory*, (Seattle, WA), pp. 2037–2041, July 2006.

[54] J. Jiang and K. R. Narayanan, "Algebraic soft decision decoding of Reed-Solomon codes using bit level soft information," in *Proc. Allerton Conference on Communications and Control*, September 2006.

[55] J. Jiang and K. R. Narayanan, "Iterative soft decoding of Reed-Solomon codes," *IEEE Communications Letters*, vol. 8, pp. 244–246, April 2004.

[56] J. Jiang and K. R. Narayanan, "Iterative soft-input soft-output decoding of ReedSolomon codes by adapting the parity-check matrix," *IEEE Transactions on Information Theory*, vol. 52, pp. 3746–3756, August 2006.

[57] J. Bellorado, A. Kavčić, and L. Ping, "Soft-input, iterative, Reed-Solomon decoding using redundant parity-check equations," in *IEEE Information Theory Workshop*, (Lake Tahoe, CA), September 2007.

[58] W. Jin and M. Fossorier, "Enhanced box and match algorithm for reliability-based soft-decision decoding of linear block codes," in *IEEE Global Communications Conference*, (San Francisco, CA), November 2006.

[59] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Transactions on Information Theory*, vol. 41, pp. 1379–1396, September 1995.

[60] A. Valembois and M. Fossorier, "Box and match techniques applied to soft-decision decoding," *IEEE Transactions on Information Theory*, vol. 50, pp. 796–810, May 2004.

[61] S.-Y. Chung, G. D. Forney, T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters*, vol. 5, pp. 58–60, Feb. 2001.

[62] N. Varnica and A. Kavčić, "Optimized low-density parity-check codes for partial response channels," *IEEE Comm. Letters*, vol. 7, pp. 168–170, 2003.

[63] T. Richardson, "Error-floors of ldpc codes," *Proc. 41st Annual Allerton Conf. Comm. Cont. and Comp.*, pp. 1426–1435, 2003.

[64] L. Sun, H. Song, Z. Keirn, and B. V. K. Vijaya Kumar, "Field programmable gate array (fpga) for iterative code evaluation," *IEEE Transactions on Magnetics*, vol. 42, pp. 226–231, Feb. 2006.

[65] H. Zhong, T. Zhong, and E. Haratsch, "Quasi-cyclic ldpc codes for the magnetic recording channel: Code design and vlsi implementation," *IEEE Trans. Magn.*, vol. 43, pp. 1118–1123, March 2007.

[66] B. Vasic and O. Milenkovic, "Combinatorial construtions of low-density parity-check codes for iterative decoding," *IEEE Trans. Inform. Theory*, vol. 50, pp. 1156–1176, June 2004.

[67] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low density parity check codes: Construction based on finite geometries," in *Proceedings IEEE Global Communications Conference 2001*, (San Francisco, CA), November 2000.

[68] W. Ryan and M. Yang, "Performance of (quasi) cyclic LDPC codes in noise bursts on the EPR4 channel," in *Proceedings IEEE Global Communications Conference 2001*, (San Antonio, Texas), November 2001.

[69] J. Li, K. R. Narayanan, E. M. Kurtas, and T. R. Oenning, "Concatenated single-parity check codes for high-density digital recording systems," in *Coding and Signal Processing for Magnetic Recording Systems* (B. Vasic and E. M. Kurtas, eds.), ch. 37, CRC Press, 2005.

[70] C. L. Weber, *Elements of Detection and Signal Design*. Springer, 1987.

[71] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Prentice-Hall, Inc., 1993.

[72] F. Franklin, D. Powell, and M. Workman, *Digital Control of Dynamical Systems*. Reading, MA: Addison Wesley, 1997.

[73] A. Patapoutian, "Head position estimation," in *Coding and Signal Processing for Magnetic Recording Systems* (B. Vasic and E. M. Kurtas, eds.), ch. 29, CRC Press, 2005.

[74] R. Wallace Jr., "The reproduction of magnetically recorded signals," *Bell Systems Technical Journal*, vol. 30, pp. 1145–1173, 1951.

[75] K. B. Klaassen and K. C. van Peppen, "Slider-disk clearence measurements in magnetic disk drives using the readback transducer," *IEEE Trans. Instrumentation and Measurement*, vol. 43, pp. 121–126, April 1994.

[76] F. Dolivo and W. Schott, "Preamble recognition and synchronization detection in partial-response systems," in *Proc. 25th Asilomar Conf. on Signals, Systems and Computers*, Nov. 1991.

[77] R. A. Scholtz, "Frame synchronization techniques," *IEEE Trans. Commun.*, vol. 28, pp. 1204–1213, August 1980.

[78] A. R. Nayak, J. R. Barry, and S. W. McLaughlin, "Optimal placement of training symbols for frequency acquisition: A Cramer-Rao bound approach," *IEEE Transactions on Magnetics*, vol. 42, pp. 1730–1742, June 2006.

[79] A. Patapoutian, "Data-dependent synchronization in partial response systems," *IEEE Transactions on Signal Processing*, vol. 54, pp. 1494–1503, April 2006.

[80] A. R. Nayak, J. R. Barry, and S. W. McLaughlin, "Joint timing recovery and turbo equalization for coded partial response channels," *IEEE Transactions on Magnetics*, vol. 38, pp. 2295–2297, Sept. 2002.

[81] J. R. Barry, A. Kavčić, S. W. McLaughlin, A. R. Nayak, and W. Zeng, "Iterative timing recovery," *IEEE Signal Processing Magazine*, vol. 21, pp. 89–102, January 2004.

[82] J. Liu, H. Song, and B. V. K. Vijaya Kumar, "Timing acquisition for low-SNR data storage channels," *IEEE Transactions on Magnetics*, vol. 39, pp. 2558–2560, Sept. 2003.

[83] W. Zeng, M. F. Erden, A. Kavcčić, E. M. Kurtas, and R. C. Venkataramani, "Trellis-based baud-rate timing recovery loop for magnetic recording channels," in *IEEE Internatinal Conference on Communications*, (Istanbul, Turkey), pp. 3179–3184, June 2006.

[84] H. Yada, "Clock jitter in a servo-derived clocking scheme for magnetic disk drives," *IEEE Trans. Magn.*, vol. 32, pp. 3283–3290, July 1996.