The Role of SP in Data-Storage

Since the early 1990s storage densities of high-end magnetic disk-drive systems have been growing at a compound rate of 60% a year. This rate is equivalent to a doubling of storage density every 18 months. Today's commercial disk-drive products can store more than four billion bits of information in every square inch of disk surface. This phenomenal density is achieved while meeting a stringent reliability constraint—the bit-error rates are typically kept on the order of 10^{-9} before using error-correction coding. The enormous growth in storage capacity has been fueled by the increased size and complexity of applications software and the sheer volume of data generated every day. The storage market is currently growing at an average rate of nearly 100% a year in terms of gigabytes shipped, and the world-wide annual revenue is expected to reach around $90 billion by the year 2000. Some industry analysts forecast that, as access to data becomes more critical in the overall computer system design, demand for computer storage will outstrip unit growth in personal computers, servers, or mainframes.

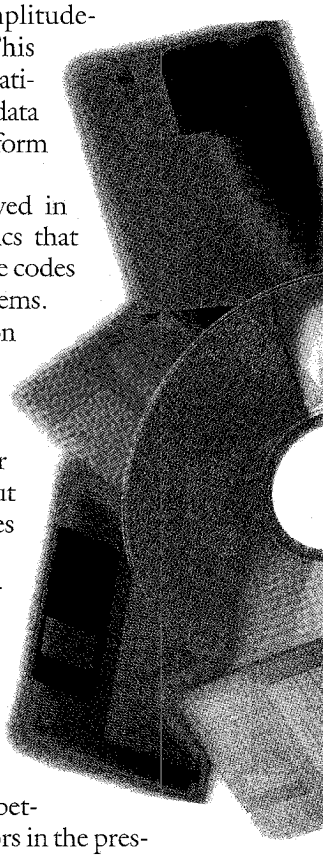
Among the key components in the development of a successful storage system are heads, media, and signal processing. In the past, major breakthroughs in the heads and media technologies have been mainly responsible for the spectacular growth in storage capacity, but signal processing is increasingly recognized as a cost-efficient means of improving density. This article addresses the issues relevant to signal processing. While the focus is on magnetic storage, most of the signal-processing strategies discussed here are also applicable to optical storage.

This article begins with a brief description of the read/write (R/W) process in magnetic storage. The characteristics of noise and nonlinearity present in real magnetic storage channels are also discussed, and a discussion on channel modeling is followed by a description of different modulation codes. Different suboptimal sequence detectors and equalizers are then presented that represent the techniques that are either already in use in commercial systems or being investigated for near-future applications. We then discuss sequence detectors, with in-depth treatment given to those based on tree search, particularly finite or fixed delay tree search (FDTS) detectors [22, 52, 56, 65, 76, 78]. Finally, performance evaluation results are presented under various channel conditions, which should be useful in assessing the potential benefits of different coding and sequence-detection strategies.

Overview

For a given head and media interface, a better signal-processing strategy will result in improved reliability. When the achieved level of reliability exceeds the prescribed requirement, this improvement can be translated into an increase in storage density by allowing more bits to be packed into the unit area of the media surface. Among other important factors considered in choosing the signal-processing strategy for a given storage system are data rates, implementation costs, and power requirements.

The R/W process can be modeled as a communication channel based on the pulse-amplitude-modulation (PAM) method. This model provides a compact mathematical relationship between the stored data bits and the continuous-time waveform produced by the read head.



The modulation codes employed in storage systems have characteristics that are quite different from those of the codes used in other communication systems. In magnetic storage, the modulation codes are used, for example, to facilitate the operation of the simple peak detector, to limit path memory in sequence detection, or to increase distance between output waveforms as seen by certain types of detectors.

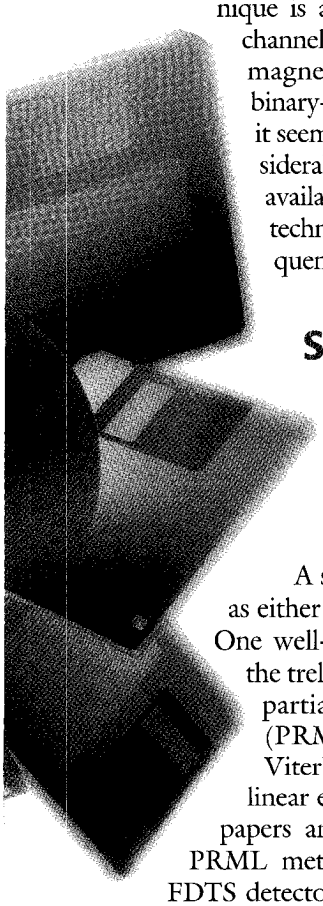
The output from a typical magnetic storage channel is corrupted by severe intersymbol interference (ISI). Sequence detectors make a symbol decision based on observation of signals over many symbol intervals and, as such, their performance is considerably better than symbol-by-symbol detectors in the presence of large signal correlation caused by ISI. The most prominent example of the sequence detector is the maximum-likelihood sequence detector (MLSD), which is typically implemented using a Viterbi detector with a long but finite decision delay [32, 45]. In real applications, however, sequence detectors are invariably used in combination with an equalizer, which reduces the effect of ISI to a certain extent. This combination of equalization and se-

1053-5888/98/$10.00©1998IEEE

Jaekyun Moon

quence detection offers a good compromise between performance and complexity.

Past developments in communication theory suggest that for ISI channels corrupted by Gaussian noise, the channel capacity should be approached through the use of a code with the relatively simple decision-feedback equalizer (DFE) to replace complex sequence detectors [20, 27, 64]. This result implies that no matter how large the ISI is, the detector need not be any more complicated than a simple DFE. Unfortunately, however, this is true only when transmitter precoding can be used to eliminate error propagation [33, 72]. The transmitter precoding technique is applicable only to multilevel input channels and cannot be used for modern magnetic storage channels, which are binary-input channels. At the present time, it seems that the only way to utilize a considerable fraction of the channel capacity available from the given heads and media technologies is to rely on some type of sequence detection.



Strategies for Increasing Density and Improving Reliability in Magnetic and Optical Recording

A sequence detector can be described as either a trellis or a tree-search algorithm. One well-known example of application of the trellis search in magnetic storage is the partial-response maximum likelihood (PRML) technique, which combines Viterbi detection with partial response linear equalization. There exist numerous papers and tutorial articles that deal with PRML methods [21, 30, 43, 68, 81]. The FDTS detectors are closely related to the delay-constrained optimum detector, which makes a symbol decision with a minimum probability of error given the explicit decision delay constraint [2].

An FDTS detector can be implemented in many different ways. One interesting method is to formulate the given FDTS detector in multidimensional signal space. The geometric interpretation of an FDTS detector provides useful insights into interesting implementation options. The signal-space formulation of an FDTS detector is particularly useful in minimizing implementation complexity in the presence of modulation-code constraints. This point will be elaborated later using some important practical examples of the code constraints and tree-depth parameters.

Communication Channel Model for the Read/Write Process

Figure 1 shows a block diagram of general data storage systems. Error-correction coding (ECC) is first applied to information bits to prevent burst errors. In commercial storage devices, the Reed-Solomon codes with certain degrees of interleaving are used almost universally for this purpose. The encoded bits are then subject to another type of coding, namely, modulation coding. As will be discussed later, there are different types of modulation codes depending on the specific need. The modulation encoded bits are the actual bits that are stored in the magnetic media. In this article, these bits will be referred to as data bits or sometimes symbols. The data bit sequence is first converted into a rectangular current waveform and then stored into the medium in the form of a magnetization waveform. This "write" process is accomplished by a small magnetic element called the write head, moving at a constant velocity above the magnetic storage medium. In modern digital recording, the written magnetization takes the form of a rectangular waveform as the medium is saturated either along or opposite to the track direction, At the time of data retrieval, the stored bit sequence is reproduced as the read head senses the magnetic flux (either the flux itself or its gradient depending on the type of read head) emitted from the written magnetization pattern. The signal at the output of the read head appears as a continuous-time-voltage waveform. The amplified version of the read-head-output waveform is then passed through what is generally called the "read channel" in the data-storage community. The read channel consists of some type of bandlimiting filter, a sampler (assuming discrete-time signal processing) driven by a timing circuit, an equalizer, and a symbol de-

Signal processing is increasingly recognized as a cost-efficient means of improving density

tector. The detected symbol or data bit sequence is then applied to a modulation decoder and finally to an error-correction decoder.

Let us now focus on mathematical modeling of the R/W process. More specifically, we shall find an analytical relationship between the stored data bits and the readback waveform available at the output of the write head/medium/read head assembly block in Fig. 1. A communication-type channel model such as this is necessary for design, analysis, and simulation of signal-processing schemes.

There are two different formats for generating the current waveform given a binary data pattern. With the non-return-to-zero-interleaved (NRZI) format, a binary 1 in the data pattern produces a transition and a binary 0 results in the absence of transition. This is shown in Fig. 2(a). With the non-return-to-zero (NRZ) method, the amplitude level in the waveform directly reflects the given binary information, as shown in Fig. 2(b). In the past, data recovery was typically carried out by a simple peak detector that determined whether a transition was present within the given symbol interval. The NRZI recording format is a preferred method with this peak detection, as an error made in a given interval has no effect in the future decisions (i.e., no error propagation). In contrast, serious error propagation will result when the NRZ waveform is reconstructed based on the peak detector outputs. With more sophisticated sequence detectors, however, the NRZI method offers no distinct advantage over the NRZ recording format.

In digital communication systems based on PAM, the received waveform is commonly expressed as

$$z(t) = \sum_k x_k s(t - kT) + n(t) \tag{1}$$

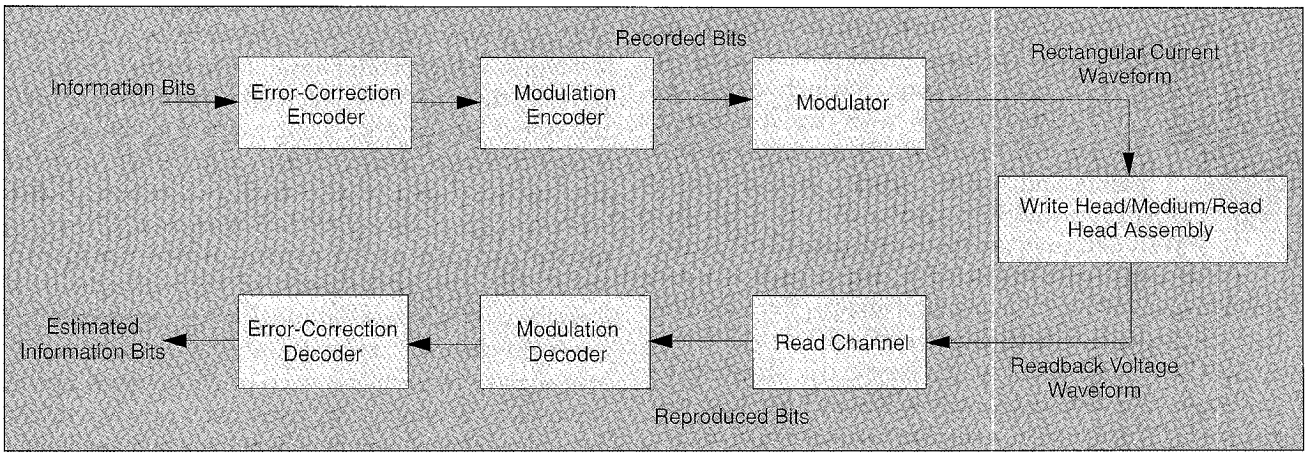
where x_k is the digital data sequence transmitted, $s(t)$ is the channel impulse response, $n(t)$ represents the additive noise observed at the receiver, and T is the symbol period. We now derive an analogous mathematical description for the R/W process of magnetic recording.

The rectangular current waveform generated according to the given binary data pattern (whether NRZI or NRZ) is viewed as the input to the write-head/medium/read-head assembly. The written magnetization waveform is similar to the input current waveform except that the transitions take place over a nonzero period. The voltage waveform generated at the output of the read head during the read process is taken as the output of the same write-head/medium/read-head assembly, and is essentially a differentiated and low-pass-filtered version of the applied waveform. The reproduced waveform is also corrupted by noise. The overall write/read process can be modeled mathematically as:

$$z(t) = \frac{dw(t)}{dt} * h(t) + n(t) \tag{2}$$

where $z(t)$ is the reproduced or readback-voltage waveform; $w(t)$ denotes the rectangular-current waveform; $h(t)$ represents the low-pass filter type of response arising from the frequency-dependent signal loss terms due to the head-medium spacing in the read process, the read head gap effect, the medium thickness effect, and the imperfect writing of magnetic transitions; $n(t)$ is the additive white Gaussian noise due to the read head and electronics; and "*" denotes the convolution. The differentiation occurs under the assumption of an inductive read element, but the modeling results will essentially be the same with the shielded magneto-resistive (MR) read head. Recording systems using thin-film media also exhibit a highly localized, nonadditive type of noise. We shall discuss this type of noise shortly.

Let us for the time being assume that the NRZ recording format is used. Then, the input current waveform, $w(t)$, is generated by multiplying each binary input, x_k, taking the value +1 or −1 by a rectangular pulse of duration T, i.e.,



▲ *1. Block diagram of a data-storage system.*

$$w(t) = \sum_k x_k p(t - kT). \tag{3}$$

Combining Eqs. (2) and (3) and realizing that

$$\frac{dw}{dt} = \sum_k x_k \left[\delta(t - kT) - \delta(t - T - kT)\right], \tag{4}$$

we can express the readback waveform as:

$$z(t) = \sum_k x_k \left[h(t - kT) - h(t - T - kT)\right] + n(t) \tag{5}$$

Comparing Eqs. (1) and (5), we see that the effective impulse response of the magnetic recording channel is $h(t) - h(t - T)$. This is usually called the pulse response since it is the head/medium response to a rectangular current pulse ($h(t)$ is called the step or transition response since it is the readback response to a single current transition). Equation (5) can be alternatively written as

$$z(t) = \sum_k a_k h(t - kT) + n(t) \tag{6}$$

where

$$a_k = x_k - x_{k-1}. \tag{7}$$

The symbol a_k takes values 2, 0, and –2, which represent a positive transition, no transition, and a negative transition, respectively. Note that the polarity must alternate between any two successive nonzero a_k symbols.

The step response is often analytically modeled as the Lorentzian function:

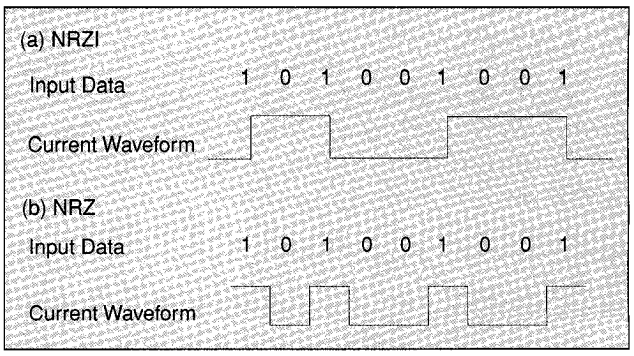
$$h(t) = \frac{1 / w}{1 + (t / w)^2} \tag{8}$$

where w represents the width parameter, which depends on the physical width of the written transition as well as other head/medium properties. While the Lorentzian function enables a convenient mathematical modeling of the step response, its validity is questionable in many real applications. Figure 3 shows a more realistic step response arising from micromagnetic modeling of a thin-film disk and a magneto-resistive (MR) head.

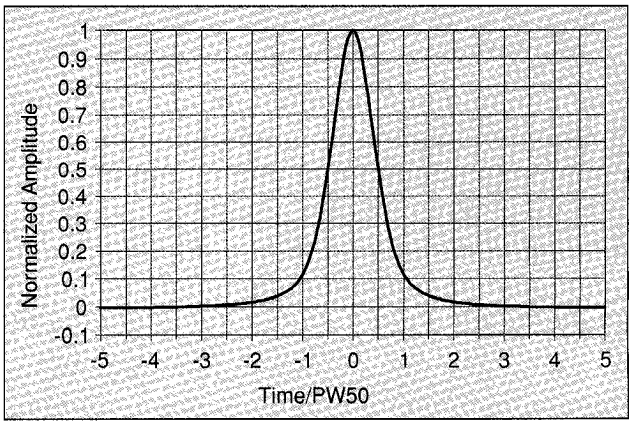
Equation (5) can also be used to describe the input/output relationship of a recording system based on the NRZI format, provided that the input sequence x_k is replaced by the precoded sequence x'_k. The sequence x'_k is obtained by passing x_k through a precoder whose transfer function is $1 / (1 \oplus D)$, where D represents a delay and $\oplus$ denotes a modulo-2 addition. The example shown in Fig. 4 illustrates that the NRZI-recorded waveform of a given bit pattern is indeed identical to the NRZ-recorded waveform of the precoded version of that pattern (to illustrate the modulo-2 operation, binary 1's and 0's are assumed in place of bipolar 1's and –1's). Therefore, when the NRZI format is used, the detector can still be designed and evaluated based on the PAM-like channel description of Eq. (5). In this case, however, the decision is made on the precoded sequence x'_k and the original bit sequence, x_k, is recovered through an inverse precoding operation (i.e., $x_k = x'_k \oplus x'_{k-1}$).

Media Noise

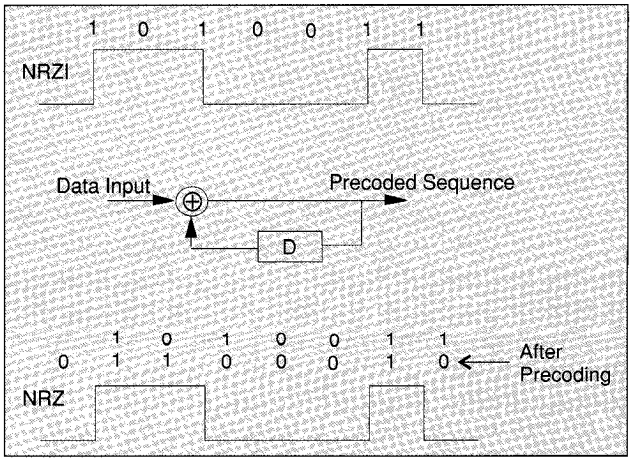
Today's advanced magnetic recording systems such as computer hard disk drives all employ thin-metallic media. In such systems, the step response may change from one transition to the next due to random variations in the geometry of magnetic transitions. This random deviation of the step pulse gives rise to what is known as transition noise [6]. This type of noise depends on the written data



▲ *2. NRZI and NRZ recording formats.*



▲ *3. Typical step response in magnetic recording.*



▲ *4. Relationship between NRZ and NRZI.*

pattern and, strictly speaking, cannot be modeled as additive noise. Transition noise can be a major problem in some high-density drives. One example is IBM's experimental system used in the gigabit-per-square-inch demonstration held in 1989. In that system, transition noise was responsible for more than 90% of the total noise power [36].

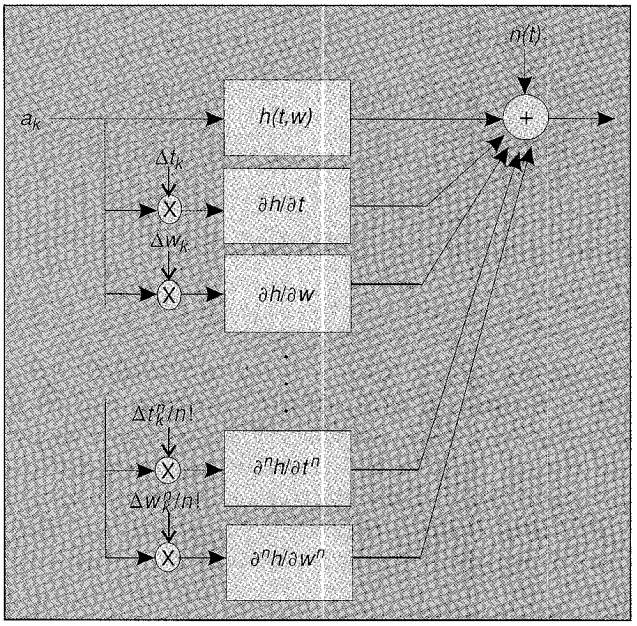
A simple yet fairly general model for transition noise can be obtained by introducing random variations to both width and position parameters of the step response [51, 53]. Let $h(t,w)$ denote the readback response to a noise-free transition located at $t=0$, where w is the width parameter. Also, let $h_k(t)$ denote the transition response for the k-th symbol interval. The subscript k emphasizes the underlying assumption that the step response is generally different from transition to transition. We assume that $h_k(t)$ is uniquely determined by two variables: the position and width parameters. The readback response to a noisy transition in the k-th symbol interval can then be written as

$$h_k(t) = h(t - kT - \Delta t_k,\ w + \Delta w_k) \tag{9}$$

where Δt_k and Δw_k are random parameters representing deviations in the position and width, respectively, from the nominal values. Taking an n-th order Taylor series expansion, the above expression can be approximated as a linear sum of the noise-free response and residual responses due to deviations around the nominal position and width of the pulse:

$$\begin{aligned} h_k \approx\ & h(t - kT, w) + \Delta t_k \frac{\partial h}{\partial t} + \Delta w_k \frac{\partial h}{\partial w} \\ & + \frac{(\Delta t_k)^2}{2!}\frac{\partial^2 h}{\partial t^2} + \frac{(\Delta w_k)^2}{2!}\frac{\partial^2 h}{\partial w^2} + \cdots \\ & + \frac{(\Delta t_k)^n}{n!}\frac{\partial^n h}{\partial t^n} + \frac{(\Delta w_k)^n}{n!}\frac{\partial^n h}{\partial w^n}. \end{aligned} \tag{10}$$

The resulting channel model is depicted in Fig. 5, where a_k represents the sequence indicating the transitions defined in Eq. (7). The multiplicative nature of transition noise is clearly shown in Fig. 5. The partial derivatives weighted by the multiples of the random constants can be viewed as residual responses. The overall channel consists of the main path due to the nominal response and paths due to the residual responses.



▲ 5. Channel model including transition noise sources.

Nonlinearity

So far it has been assumed that the signal portion of the received waveform can be constructed from linear superposition of isolated step responses. In practice, this is true only at low recording densities where magnetic transitions are well separated from one another. As density increases, however, nearby transitions start to interact, resulting in significant nonlinear distortion [61]. An important source of nonlinearity is the shift in transition positions, which occurs as the demagnetizing field of the previous transition influences the head field writing the current transition. Another type of nonlinearity is the broadening of the current transition as the head field gradient is reduced by the demagnetizing field from a previous transition [53]. Thus, a transition is shifted earlier in time and tends to broaden due to the effect of the transition written in the previous symbol interval. Yet another form of nonlinearity, which becomes significant as transitions approach each other even closer, corresponds to partial erasure of adjacent transitions [49]. At the readback level, this appears as a sudden reduction of the signal amplitude. While the position shift can be eliminated to a large extent by precompensating the write current so that the actual written position of a transition coincides with the intended position, the transition broadening and partial erasure effects are difficult to avoid as linear density increases.

Unlike transition noise, nonlinearity is a repeatable phenomenon. The output of a nonlinear magnetic channel can be described as a superposition of pattern-dependent step responses. Let the sequence of transition symbols that contribute to nonlinearity at time k be denoted by the vector $\mathbf{a}_k$. Emphasizing the pattern dependence of the step response, we can write the readback waveform arising from a nonlinear magnetic channel as

$$z(t) = \sum_k a_k h(t - kT, \mathbf{a}_k) + n(t). \tag{11}$$

Expressing the nonlinear distortion as a deviation around the linear step response, we can write

$$h(t) = \sum_k a_k \left[h(t - kT) + \gamma(t - kT,\ \mathbf{a}_k)\right] + n(t) \tag{12}$$

where $\gamma(t - kT,\ \mathbf{a}_k)$ represents the pattern-dependent deviation from the linear response in the k-th symbol interval.

Advanced signal processing has been playing an increasingly important role in improving densities in magnetic storage devices.

The nonlinear distortion can also be described by the Volterra functional series. The Volterra series model constructs the nonlinear portion of the signal as the sum of the outputs of nonlinear "kernels." These nonlinear kernels act like the impulse response of a linear system except that each kernel is driven by a product of the present symbol and some combination of the neighboring symbols contributing to the present nonlinearity. As an example, assume that the nonlinear distortion for the kth symbol interval is affected by three preceding symbols a_k, a_{k-1} and a_{k-2}. Then, the Volterra model describes the nonlinear distortion with three different kernels: two driven by the second-order symbol products, $a_k a_{k-1}$ and $a_k a_{k-2}$, and one driven by the third-order product $a_k a_{k-1} a_{k-2}$. This is depicted in Fig. 6, where $h_1^{(2)}(t)$ and $h_2^{(2)}(t)$ represent the two second-order kernels and $h^{(3)}(t)$ the third-order kernel.

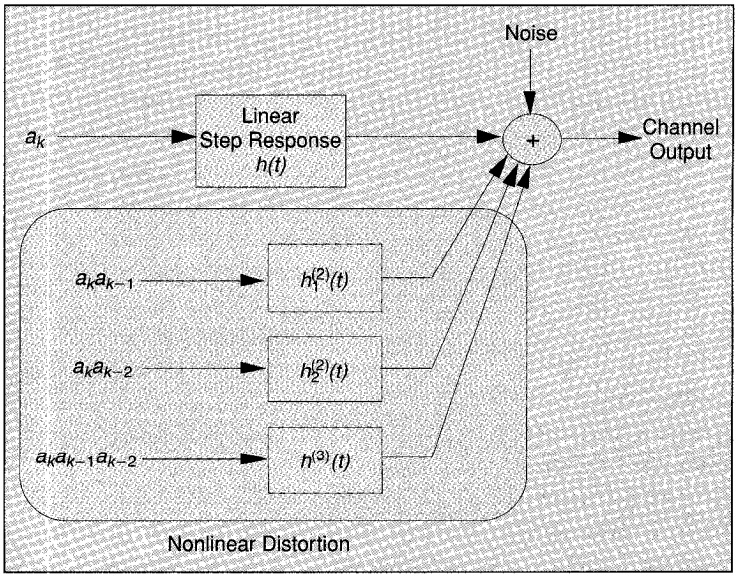
The kernels in Fig. 6 can be estimated by training the assumed structure to measured data. A more systematic approach is also possible to identify them from experimental measurements [35, 46]. Other nonlinearity characterization methods exist that rely more heavily on physical understanding of the write process [3, 61, 83, 85].

Different signal-processing strategies have been developed in the past that are geared specifically toward particular characteristics of the transition noise and nonlinearity. These techniques range from perceptron-like equalizers to sequence detectors with modified metric computation [4, 48, 60, 84]. While we expect to see continued research efforts in this direction, these techniques are mostly exploratory at the present time. The detection techniques reviewed in this article do not attempt a direct optimization against transition noise and nonlinearity, but rather consider them as part of additive random noise. Even when the direct optimization is not a major goal, the transition noise and nonlinearity models discussed above are useful in evaluating performance of various signal-processing ideas under realistic channel conditions.

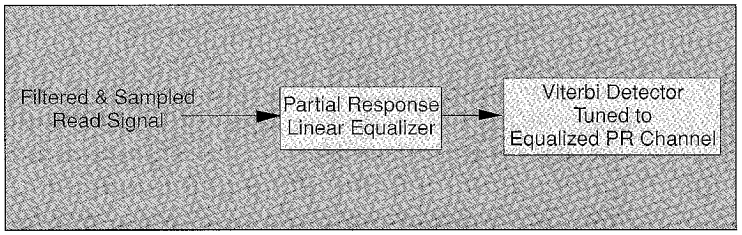
Modulation Coding

With modulation coding a certain constraint is imposed on the input bit sequence so that some desired characteristics are maintained in the stored bit patterns. For this reason modulation coding is also called constrained coding. In digital storage channels, constrained coding is used for a variety of reasons. A prime example is the minimum-run-length constraint, also called the d-constraint, which forces run lengths of like symbols to be at least $(d+1)$ and thereby separates magnetic transitions by at least $(d+1)$ symbol intervals (according to the NRZ recording convention) [67]. The d-constraint was originally introduced to magnetic recording in order to minimize the pulse overlapping to facilitate the operation of simple peak detection. But, as the linear density requirement becomes more stringent, there seems to be a more fundamental reason to employ the d-constraint; for the same user density, the d-constraint provides increased distance between adjacent transitions, which helps to reduce the effects of severe nonlinearities arising from closely spaced transitions. In practice, transition spacing is also upper-bounded by $(k+1)$ symbol intervals so that nonzero signal levels arise frequently enough for a proper operation of the timing circuit. This constraint is called the "k-constraint." Run-length-limited coding with both lower and upper limits on run lengths of like symbols is referred to as "(d,k)" coding.

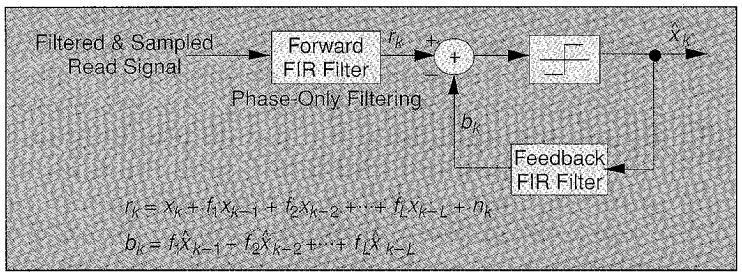
One disadvantage of the d-constraint is that the required code rate is considerably lower than the codes based only on the k-constraint. This means that for the same user data rate, the symbol rate must be higher. For example, the rate 2/3 $(d=1,k)$ code runs at a clock rate 50% higher than that of the rate 8/9 code with the k-constraint only; this puts the (d,k) code at a considerable disadvantage for storage systems that are limited by the clock speed. However, for systems that are not limited by the processing speed of the circuit and for which density is the most important concern, the (d,k) code seems to be the better choice. This is especially true when the ratio of the width of the read pulse to the bit cell length is large (i.e., severe ISI), a situation in which the $d=1$ code has been shown to exhibit a minimum-distance increasing property, resulting in a performance improvement upon



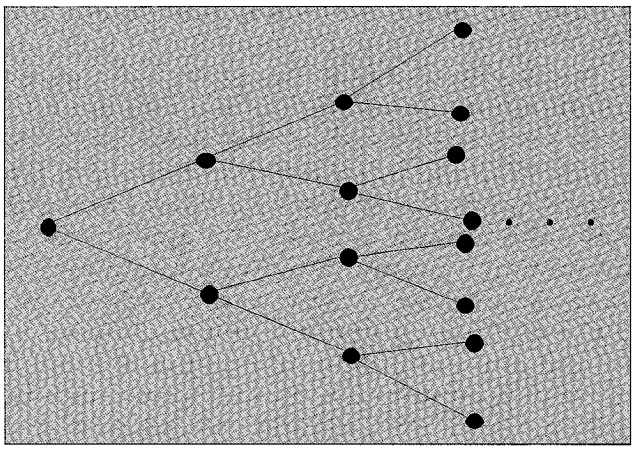
▲ *6. Volterra series model for nonlinearity.*



▲ 7. *Partial-response maximum-likelihood technique.*



▲ 8. *Decision-feedback equalization.*



▲ 9. *Binary tree.*

the $d=0$ codes despite a larger rate loss [38]. The d-constraint can also provide robustness against channel parameter fluctuations [37, 54].

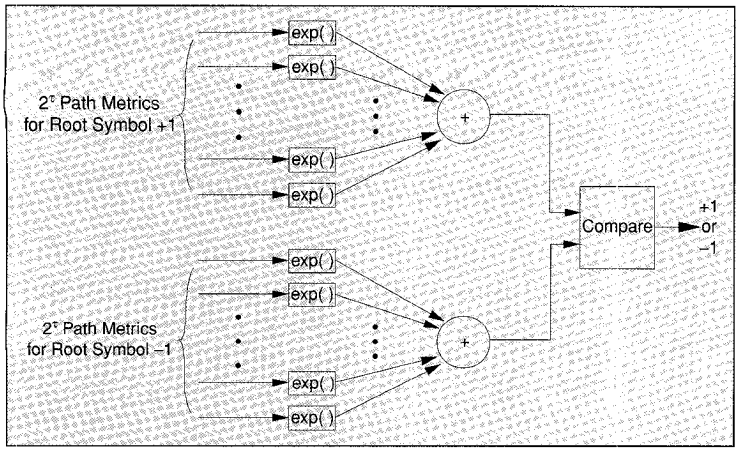
Today's advanced hard disk drives employ the PRML detector in place of the traditional peak detector. The PRML scheme combines partial-response linear equalization with Viterbi detection and requires a different type of code constraint [68, 79]. To reduce complexity and improve speeds, the present PRML system is implemented as two interleaved subchannels, one operating on the odd-indexed bits and the other on the even-indexed bits. The PRML code limits the number of consecutive binary 0's in each of the two interleaved data bit sequences to I. This I constraint is necessary to force a finite decision delay in Viterbi detection. The Viterbi detector keeps track of more than one candidate data bit sequences and makes the decision for a given symbol interval only after all the candidate sequences agree up to that point. Thus, a forced delay is necessary to limit the memory requirement in practice. For timing purposes, a run-length limitation of G consecutive 0's is also imposed on the actual data bit sequence (same as the k-constraint). The code, however, has no minimum run-length constraint, i.e., $d=0$. For these reasons, the PRML code is also referred to as the $(0,G/I)$ code. The typical values for G/I are 4/4 and 3/6, which can be implemented with the rate 8/9. An improved rate of 16/17 is also possible without relaxing the G and I constraints significantly [73].

Constrained coding is also used to directly achieve a coding gain. Results obtained by Karabed and Siegel [39] have shown that a large coding gain is possible by imposing a constraint that forces nulls in the power spectrum of written sequences that match with those in the channel-transfer function. This result is remarkable in the sense that code design, which has traditionally been pursued independently of the channel response, can now take advantage of given spectral properties of the channel. A data-recovery scheme that combines a matched spectral null code with PRML has been designed and implemented in VLSI chips [71]. A general technique is available for designing constrained codes with efficient encoders with sliding block decoders. This technique is based on constructing a constrained graph according to the given code constraint and performing the state-splitting algorithm [47]. This technique can be applied to a very large class of codes, including all practical constrained codes encountered in data storage channels.

Another constrained code that results in a direct coding gain is the recently introduced maximum transition run (MTR) code [58]. The basic idea is to eliminate the critical bit patterns that cause most errors in sequence detectors. More specifically, the MTR code eliminates input patterns that contain j or more consecutive transitions in the corresponding current waveform. As a result, for $j=2$ the NRZ error events of the form $\pm\{2\ -2\ 2\}$ are eliminated. This improves the performance of a broad class of near-optimal sequence detectors substantially at high linear densities. The code possesses the similar distance-gaining property of the $(1,k)$ code operating at a high linear density, but it can be implemented with considerably higher rates. In addition to enhancing distances, the MTR codes are useful in controlling nonlinearities and other write-side anomalies associated with crowded transitions. As a simple example of an MTR code with $j=2$, 16 codewords required to implement the rate 4/5 block code are given in Table 1. The NRZI format is assumed in mapping the codewords to the magnetization wave-

Table 1. Codewords for a rate 4/5 MTR block code with $k=8$.

00001	00110	01100	10010
00010	01000	01101	10100
00100	01001	10000	10101
00101	01010	10001	10110



▲ *10. Decision-making process in FDTS.*

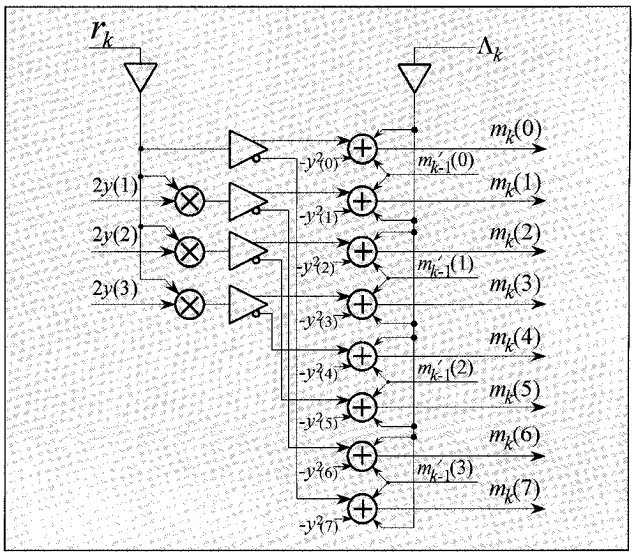
form. It can be seen that consecutive runs of 1's are limited to 2 both within the codewords and at the codeword boundaries. The runs of consecutive 0's are also limited (to 8 in this example) for timing-recovery purposes. Bit-error rate (BER) simulation results indicate that the MTR code yields large performance improvement over existing codes when used with high-order PRML and FDTS detectors. The code rate and/or the k-constraint can be further improved by using a state-dependent encoding strategy [17]. There also exist other types of code constraints that posses similar distance-enhancing properties for high-order PRML systems [40]. A location-dependent MTR code constraint has also been considered that allows certain crowded transition patterns depending on their starting positions [12, 31, 74].

Detection Techniques

Advanced signal processing has been playing an increasingly important role in improving densities in magnetic storage devices. Until recently, simple analog peak detection has been the universal choice for data detection in magnetic data-storage devices. A peak detector operates on the analog readback signal to determine the presence of a pulse within a predetermined observation window that is sliding in time [66]. A readback pulse occurs where there is a transition in the input current waveform, and the written data pattern can be reconstructed by correctly identifying the pulse positions in the readback signal. When the density requirement is moderate, combining a peak detector with an error-correcting code and a (d,k) run-length-limited (RLL) code provides an adequate means to recover data with relatively low implementation cost. However, as linear density increases, overlapping between neighboring pulses becomes severe and the peak detector performance deteriorates rapidly due to large peak shifts and amplitude reductions.

Among notable recent advances in signal processing for magnetic recording is the PRML scheme [21]. The basic idea of the PRML method is illustrated in Fig. 7. The sampled read signal is applied to a linear equalizer, which forces the transfer function of the combined channel filter and equalizer to a $(1-D)(1+D)^n$ polynomial, where D represents a symbol delay and n is a positive integer [70, 75]. The equalizer output signal is then fed to the Viterbi detector tuned to the ISI coefficients corresponding to the $(1-D)(1+D)^n$ target function. While offering a reasonably good match to the natural channel response, this particular family of target functions yields a smaller number of distinct signal levels at the equalizer output than more general target functions with the same number of ISI terms. This, combined with the feature that the signal levels are integer-valued, significantly reduces the number of multipliers required in the implementation of the Viterbi detector. In the absence of noise enhancement and noise correlation, the Viterbi detector will perform maximum-likelihood detection. But, there is an obvious loss of optimality associated with the PR equalizer. A close match between the target function and the natural channel will guarantee that this loss will be small. Equalization is typically done using a finite-impulse-response (FIR) digital filter, but it can also be done in continuous time before sampling occurs. In some commercial disk drives, the equalizers are trained at the factory floor and are "frozen" before shipping. In others, equalizers are designed to operate adaptively in real-time. For more discussion on adaptive equalization for storage channels, the reader should consult other references [10, 19].

Of the $(1-D)(1+D)^n$ PR family, a popular choice of partial response is the class IV partial response (PR4) characterized by the $1-D^2$ transfer function (i.e., $n=1$). The PR4 target provides a reasonably good matching to the natural head/medium response of the magnetic storage channel at low to medium linear densities. PR4 equalization also allows the interleaving of the readback signal into two dicode signals to which a simplified,

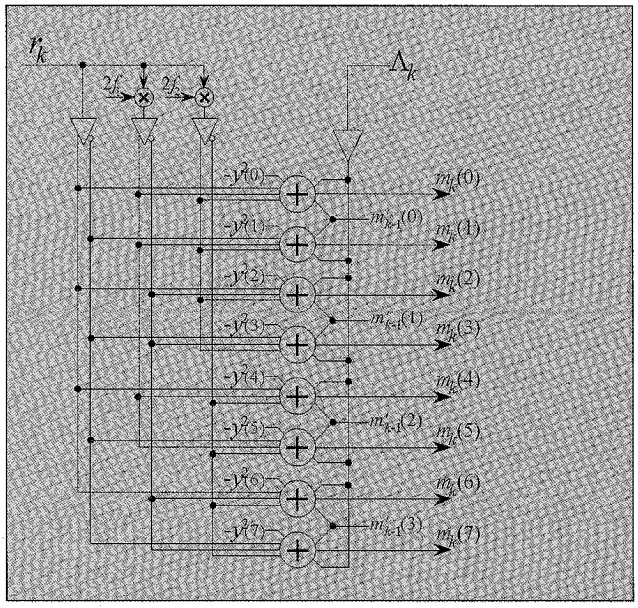


▲ *11. Recursive path metric computation for τ=2 (2y(0) is set to 1).*

Viterbi-equivalent detection algorithm can be applied [28, 81]. Today's PRML systems operate with $(0,G/I)$ codes that do not have the minimum run-length constraint and, at current linear densities, this results in a significant code-rate advantage over the conventional $(d=1,k=7)$ RLL code. Overall, the improved noise immunity and ability to combat ISI more effectively, coupled with the rate advantage of the $d=0$ modulation code, makes the PRML technique considerably more attractive than the conventional peak detection scheme.

The PRML method has also been combined with (1,7) coding. The need for the $d=1$ constraint may arise in some low-end applications to facilitate the write process by placing transitions further apart. With (1,7) coding, however, the symbol density is much higher than the $d=0$ coded systems. As a result, a considerably better matching is achieved with the PR target $(1-D)(1+D)^3$, which is called the E2PR4. Although the number of ISI terms is increased significantly, the complexity of the Viterbi detector is kept reasonable as the $d=1$ constraint eliminates many states and state transitions in the trellis diagram. This particular PRML method has also been employed in commercial VLSI read channel chips [30].

Another technique of interest is the DFE [5, 7]. The DFE does not force the readback response to a predetermined target and, thus, suffers from smaller equalization loss than PRML techniques at high linear densities. Figure 8 illustrates the basic DFE operation. Proper forward filtering is critical to the success of a DFE read channel. The forward filter in a DFE performs a minimum phase filtering on the read signal. This turns the ISI in the forward filter output, r_k, causal with a tendency to emphasize near ISI terms more than the distant ones. In the feedback loop, the latest decisions are used to construct b_k, a replica of the ISI contribution seen at the forward path. If the past decisions are correct, entire ISI will be canceled



▲ *12. Recursive path metric computation based on the expanded metric ($2f_0$ is set to 1).*

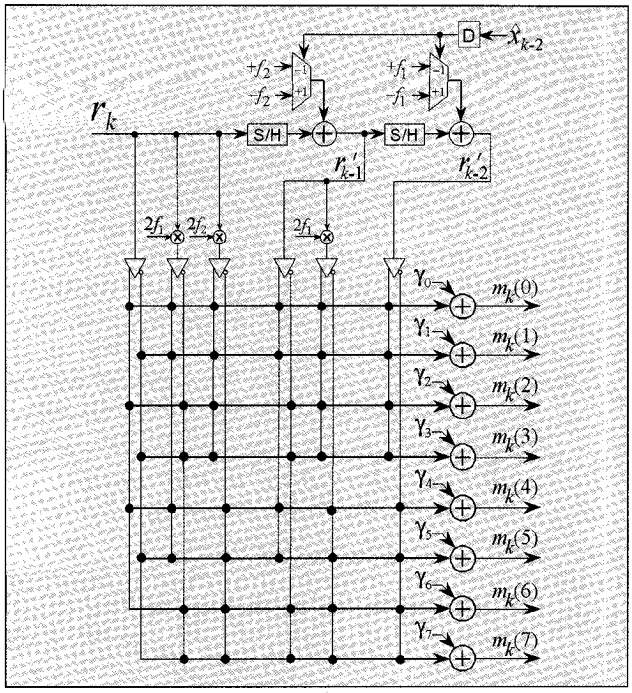
Detection strategies based on two parallel DFE structures offer significant performance advantage over the conventional DFE.

and decision can be made using a simple threshold detector. The feedback filter can be implemented either by an FIR filter or a random access memory (RAM). When the feedback filter is implemented using a RAM, the DFE also can effectively counter the causal part of nonlinear distortion [29], provided that the nonlinear effects can be predicted using either a parameterized nonlinearity model or by direct training on test data. Much efforts are now being directed to efficient circuit implementation of DFEs. Prototype DFE channel chips operating at high speeds have been developed.

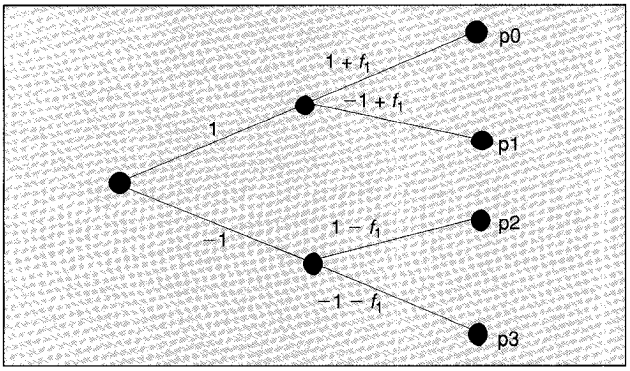
Detection strategies based on two parallel DFE structures offer significant performance advantage over the conventional DFE [9, 23]. The dual DFE (DDFE) described in [9] is particularly attractive from the implementation standpoint. The two DFEs that operate in parallel are identical except that the associated threshold levels are biased toward a positive and a negative value, respectively. In this way, an ambiguity zone is created in between the two threshold levels. When the threshold detector input falls in this region, the two detectors generate opposite decision bits, signaling the beginning of an "erasure period." During the erasure period, the two threshold levels are set at zero to maximize the signal margin. The decision is made after a fixed delay by favoring one DFE over the other, based on the threshold errors (difference between the threshold detector input and the decision bit) accumulated over the erasure period. As the erasure period is forced to an end, the feedback shift register contents of the favored DFE are copied into the other equalizer and the threshold levels are reset at the original biased values. Analysis in [9] shows that the DDFE can offer up to a 1.94 dB SNR gain over the conventional DFE.

A modified DFE structure also exists that can counter the effect of nonlinearity arising from the present and future symbols as well [86]. The basic idea is to dynamically adjust the decision threshold in anticipation of nonlinear distortion contributed by the current and future transitions. Since a significant portion of nonlinear distortion comes from the present and future transitions, this modification results in a visible improvement on the eye opening of the signal at the threshold detector input.

Among the approaches that employ the minimum run-length codes is the (1,7) maximum-likelihood technique, which provides an efficient near-optimal sequence detection of (1,7) run-length-coded data [62, 63]. The Viterbi detector can also be used in conjunction with a higher-order partial-response target that of-



▲ *13. Nonrecursive path metric computation.*



▲ *14. A τ=1 tree labeled with expected signal values.*

fers a better matching to the unconditioned channel spectrum than the PR4 target. Depending on the linear density, the performance improvement over the existing PRML technique based on the PR4 target can be substantial, but the required complexity is expected to be much higher as well.

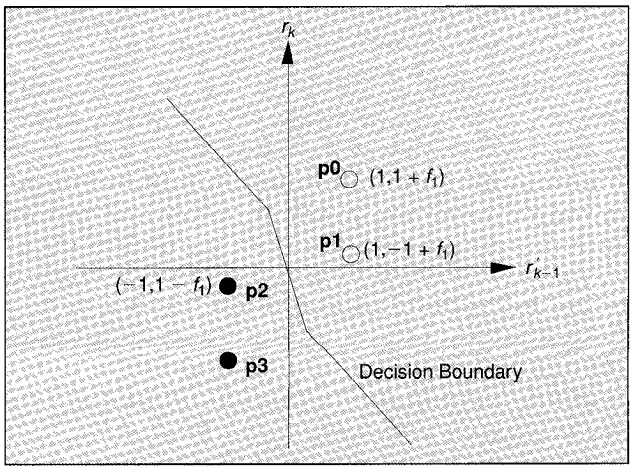
An interesting technique has been proposed to offset the increased complexity requirement associated with the extended PR4 (EPR4) target characterized by the transfer function $(1-D)(1+D)^2$ [82]. Let PR4ML and EPR4ML refer to the PRML methods based on the PR4 and EPR4 targets, respectively. The first step in this approach is to generate an EPR4 error sample sequence by passing a PR4 error sequence through the $(1+D)$ digital filter. The PR4 error sequence here is defined as the difference between the captured PR4-equalized read signal and the noiseless three-level PR4 samples, constructed from the PR4ML decision bits. This tentative EPR4ML error sequence is then compared with the known output error patterns that are most common to the EPR4ML channel. If the tentative error sequence is sufficiently close to any of the dominant EPR4 error patterns, then the PR4ML decision bits are corrected accordingly. The resulting performance approaches EPR4ML, whereas the architecture is based on the existing, much simpler PR4ML and some add-on circuitry. PRML performance can also be improved by incorporating a noise predictor in the branch metric computation of the Viterbi detector [25]. This technique also attempts to make use of existing PRML architectures.

Another example of a detection scheme that provides efficient symbol recovery is the fixed-delay tree search with decision feedback (FDTS/DF) [52]. While the FDTS/DF is essentially a finite-depth tree search utilizing past decisions to remove excessive ISI terms, it is most effective when applied to constrained channels such as minimum-RLL channels and maximum-transition-run channels. The remaining portion of this article will focus on the FDTS/DF and related techniques. It will be shown that the signal-space formulation of the tree-search algorithm and constrained signals leads to highly efficient implementation options.

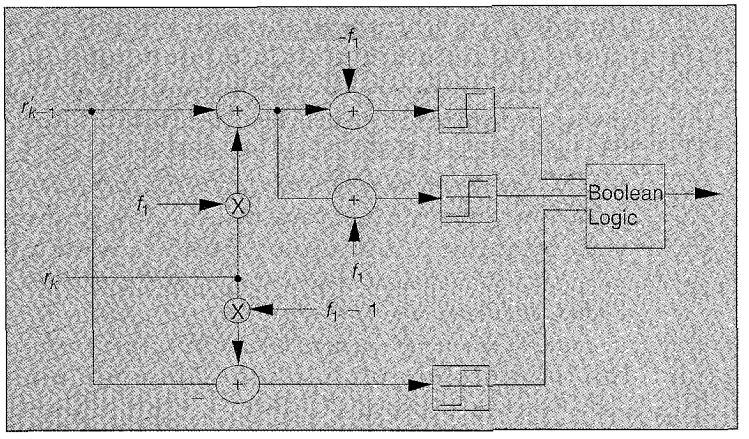
Fixed-Delay Tree Search

The FDTS/DF employs a DFE-like minimum-phase filtering in the forward section to create a causal channel response. The feedback filter driven by past decisions then cancels some of the past ISI terms. This operation is the same as the DFE except that not all the ISI terms are canceled by the feedback filter; in this approach, the τ most recent ISI terms are allowed to enter the decision element, which in turn utilizes the signal energy contained in the allowed ISI terms in making an improved decision. Assuming past decisions are correct, the observation sample available at the input of the decision element at time k is given by

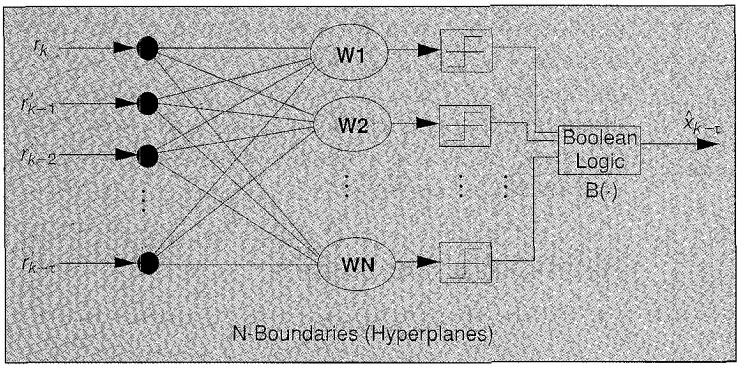
$$r_k = x_k + f_1 x_{k-1} + f_2 x_{k-2} + \cdots + f_\tau x_{k-\tau} + n_k \qquad (13)$$



▲ *15. Signal-space representation of FDTS with τ=1.*



▲ 16. Implementation of signal-space FDTS.



▲ 17. Implementation of a decision boundary consisting of N hyperplanes.

where f_k represents the causal channel response seen at the output of the forward equalizer. The first coefficient, f_0, is assumed to be 1 with no loss of generality. In order for the feedback cancellation scheme to work, the FDTS detector must make a decision with a fixed symbol delay of τ. The FDTS algorithm can also be viewed as the delay-constrained optimal detector (DCOD) originally proposed by Abend and Fritchman [2], provided that the most recent past decision is correct [56]. But, in general, the FDTS results in a significant reduction in the processing and storage requirements compared to the DCOD. The performance analysis is also straightforward by computing the minimum distance between any two look-ahead paths that diverge at the root [56].

Figure 9 shows a typical binary tree, where two branches coming out of each node correspond to two possible binary input data symbols. Each possible input sequence corresponds to a path through the tree defined by a sequence of branches. Associated with each branch is the noiseless output signal, y, which is a weighted combination of the preceding τ input symbols. For each tree branch the branch metric is defined to be $-(r - y)^2$.

The FDTS detector can be decomposed into the following steps. First, the detector looks ahead (from a given node) τ levels into the tree by computing $2^{\tau+1}$ accumulated path metrics for $2^{\tau+1}$ possible paths. These accumulated metrics associated with each possible look-ahead path are then used to decide whether the symbol at the root of the tree should be +1 or −1 based on some decision rule. The determined symbol is then released as the final decision for that cycle. Finally, the detector discards all the paths that do not share the selected root symbol (i.e., either upper half or lower half of the look-ahead paths whose root symbol is not consistent with the released symbol will be discarded). In the next symbol interval, a new observation sample is received and the surviving paths are extended one level further. The procedure is repeated. Note that because the path metrics are stored in a fixed order, there is no need to maintain survivor paths; the path history is implied in the ordered list of the metrics.

The typical decision strategy is to choose the largest accumulated metric and release the root symbol associated with the corresponding path as the final decision [65]. Another possible strategy is to weigh the metrics with the inverse of the noise power, pass them through a parallel exponential processor, and sum the outputs according to the root symbol, as shown in Fig. 10. The symbol decision is made by comparing the block-wise sums. When the noise is Gaussian, this strategy follows directly from the DCOD assuming that the decision made in the previous interval is correct [56]. If the noise is not Gaussian, the results will be essentially the same as those based on the largest metric path under reasonable high SNRs. This is because even small differences between individual metrics are greatly exaggerated by the exponential functions and the largest metric will always predominate other metrics at the output of the exponential processors. While the performances of two decision strategies are similar, the decision rule utilizing exponential functions may offer considerable advantage in terms of implementation complexity and processing requirements. The implementation of the exponential characteristic is straightforward using bipolar transistor characteristics and high-speed implementation is possible using current-mode processing [14], whereas the maximum-finding operation is often a costly and time-consuming process.

For magnetic channels with constrained codes, the above decision strategy is modified by identifying the look-ahead paths that violate the code constraint and blocking the associated metrics in the summing operation. As pointed out in [52], the FDTS/DF techniques are most useful for constrained channels because of the favorable performance/complexity tradeoff options that they can offer.

Path Metric Computation

As reviewed above, the FDTS algorithm can be divided into two major steps: metric computation and decision ruling. The decision-making process can be implemented

based on the idea depicted in Fig. 10, utilizing the exponential *I-V* characteristic of a *pn*-junction of a bipolar transistor [14].

In the following, we focus on the metric computation process. The branch metric λ_k is expressed as

$$\lambda_k = -(r_k - y_k)^2 = -r_k^2 + 2r_k y_k - y_k^2. \quad (14)$$

Figure 11 depicts a straightforward way of recursively computing the path metric for the case of $\tau=2$. The path metric for the ith path, $m_k(i)$, is obtained by adding the latest branch metric $2r_k \cdot y_k(i) - [y_k(i)]^2$ (ignoring the term $-r_k^2$, which is common to all branches) to an appropriate path metric carried over from the last cycle. Due to the symmetric nature of a tree, the expected signal values for the bottom half of the paths are equal to the negative of those in the top paths, i.e.,

$$y_k(j) = -y_k(2^{\tau+1} - 1 - j) \quad \text{for} \quad 0 \le j \le 2^{\tau+1}, \quad (15)$$

This property is used in Fig. 11 to halve the number of multipliers necessary in metric computation. The quantity Λ_k represents a constant offset that is used to prevent the accumulated metric values from overflowing.

The cross product term $2r_k \cdot y_k$ can be expressed as

$$2r_k \cdot y_k = 2r_k(x_k + f_1 x_{k-1} + f_2 x_{k-2}). \quad (16)$$

Realizing that x_k's are either +1 or −1, Eq. (16) can be rewritten as

$$2r_k \cdot y_k = \pm 2r_k \pm 2r_k f_1 \pm 2r_k f_2 \quad (17)$$

where the eight different combinations (for $\tau=2$) of the + and — signs correspond to the eight look-ahead paths [16]. An alternative implementation strategy for recursive path metric computation based on Eq. (16) is shown in Fig. 12. Clearly, with this method, the number of multipliers necessary increases only linearly, whereas the number of the inputs to each adder is still an exponential function of τ. Thus, this implementation might be suitable for current-mode processing where the multi-input add operation is relatively easy to realize.

The potential latency problem associated with recursive metric computation can be mitigated by computing the branch metrics of all branches for a given look-ahead path, not just the latest branch, and adding them anew each cycle. With $\tau=2$, the path metric at time k is given by

$$m_k = -(r_k - y_k)^2 - (r_{k-1} - y_{k-1})^2 - (r_{k-2} - y_{k-2})^2 \quad (18)$$

where the expected signals y_{k-1} and y_{k-2} depend on past decisions $\hat{x}_{k-3}$ and $\hat{x}_{k-4}$, i.e.,

$$y_{k-1} = x_{k-1} + f_1 x_{k-2} + f_2 \hat{x}_{k-3} \quad (19a)$$

$$y_{k-2} = x_{k-2} + f_1 \hat{x}_{k-3} + f_2 \hat{x}_{k-4} \quad (19b)$$

The role of advanced detection schemes becomes more critical as the demand for higher user density continues.

It is sometimes more convenient to express the path metric in terms of the expected signals that have no dependency on past decisions. We define

$$y'_{k-1} = x_{k-1} + f_1 x_{k-2} \quad (20a)$$

$$y'_{k-2} = x_{k-2}. \quad (20b)$$

Then, the path metric can be rewritten as

$$m_k = -(r_k - y_k)^2 - (r'_{k-1} - y'_{k-1})^2 - (r'_{k-2} - y'_{k-2})^2 \quad (21)$$

assuming the terms depending on the past decisions have also been removed from the observation signals, i.e.,

$$r'_{k-1} = r_{k-1} - f_2 \hat{x}_{k-3} \quad (22a)$$

$$r'_{k-2} = r_{k-2} - (f_1 \hat{x}_{k-3} + f_2 \hat{x}_{k-4}). \quad (22b)$$

After expanding the square terms in Eq. (21), dropping the terms that are common to all paths and expressing the cross products between the observation and the expected signal in the form of Eq. (17), we obtain the structure shown in Fig. 13. Notice that there are only small increases in the number of multipliers and in the number of inputs to each add element, compared to the recursive architecture of Fig. 12. The advantage of the nonrecursive architecture of Fig. 13 is that the number of the sample/hold (S/H) components, which can be a major source of power consumption in analog circuit implementation, grows only linearly as a function of τ. Also, the architecture of Fig. 13 is more suitable for various pipelining and parallel-processing strategies necessary for high-speed implementation.

Signal-Space Formulation of FDTS

In essence, the FDTS algorithm finds the look-ahead path that is closest to the observation sample sequence in the Euclidean sense and identifies the binary decision symbol associated with that path. This problem can be viewed as finding the nearest neighbor in the $(\tau+1)$-dimensional signal space [18, 41]. This geometric interpretation of the FDTS algorithm leads to some interesting implementation options. As will be explored later, this is true especially when constrained coding is imposed on the input data sequence.

Let us take the $\tau=1$ example. The two consecutive observation samples available at the FDTS detector input,

$r_k = x_k + f_1 x_{k-1} + n_k$ and $r_{k-1} = x_{k-1} + f_1 x_{k-2} + n_{k-1}$, are to be used to make a decision on x_{k-1}. After removing the terms depending on the past decision, the observation samples become $r_k = x_k + f_1 x_{k-1} + n_k$ and $r'_{k-1} = x_{k-1} + n_{k-1}$. Figure 14 shows a $\tau=1$ tree with the branches labeled with the expected signal values, $y_k = x_k + f_1 x_{k-1}$ and $y'_{k-1} = x_{k-1}$. The ith look-ahead path is denoted by pi.

In Fig. 15, the look-ahead paths are represented as vectors or points in the 2-dimensional signal space. If the observation point specified by r_k and r'_{k-1} has as the nearest signal p0 or p1 (p2 or p3), then the decision is made in favor of +1 (–1). This process is summarized by the decision boundary made up of three straight lines perpendicularly bisecting the signal pairs associated with different binary symbols; if the observation falls in the right (left)-hand side of the boundary, the decision should be +1 (–1). The decision rule can be stated as

$$\text{Decide } x_{k-1} = 1 \text{ if}$$
$$\left(f_1 r_k + r'_{k-1} - f_1 \geq 0\right) \textit{ or}$$
$$\left\{\left(f_1 r_k + r'_{k-1} + f_1 \geq 0\right) \textit{ and } \left(r_k - f_1 r_k - r'_{k-1} \geq 0\right)\right\} \quad (23)$$

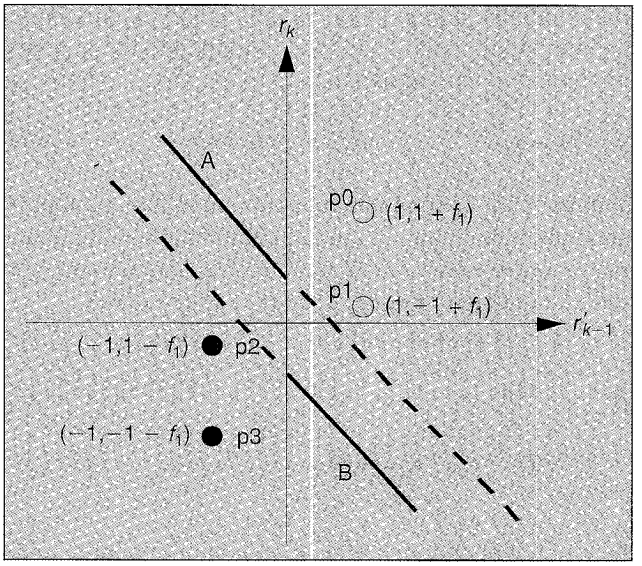
and $x_{k-1} = -1$ otherwise.

Each of the three inequalities involved in Eq. (23) indicates the condition whether the observation point is in the right-hand or left-hand side of the corresponding straight line bisector. Figure 16 shows a block diagram implementation of this decision rule, where the Boolean logic represents the two-level AND-OR operation corresponding to Eq. (23).

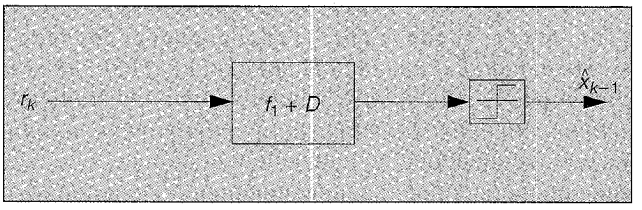
In the above example, the decision rule is found by simple inspection of the expected signal points in the 2-dimensional space. However, as the dimensionality increases, this approach is no longer feasible. Given that the equation of a hyperplane is described as a known linear combination of τ observation variables (plus a constant offset term) and that the decision boundary is made up of some of these hyperplanes, a general signal-space detector takes the form shown in Fig. 17.

Looking at the structure of Fig. 17, a natural question arises: how can we find the minimum set of linear weights, w's (including the constant offset terms), and the logic rule that would yield the optimal decision boundary? Fortunately, there exists a systematic way to find an exact answer to this question. This method is based on the concept of the Voronoi diagram (VoD) and Delaunay tessellation (DT) developed in computational geometry. In a general multidimensional space, there are signal pairs that do not contribute to the construction of the decision boundary. For example, the line bisecting the signal pairs p0 and p3 in Fig. 15 does not play a role in the decision process. The DT provides a systematic way of identifying all signal pairs whose bisecting planes do not affect the decision boundary in the multidimensional signal space. As a consequence, the minimum set of hyperplanes and associated logic combination that realize the optimal decision boundary can be found analytically. For more detailed information, the reader should consult other references [13, 59].

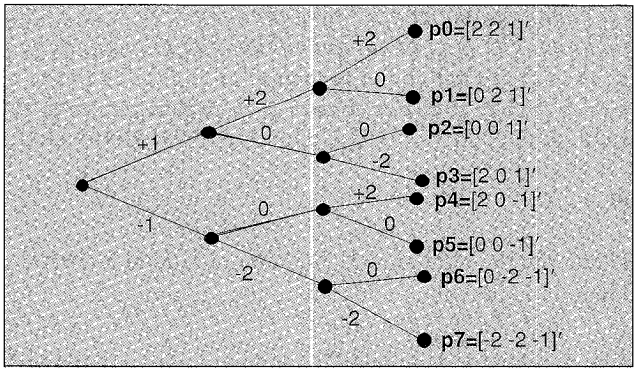
When the dimensionality is low enough (e.g., τ less than 4), a direct search process combined with Monte Carlo BER simulations can also be used to find the appropriate weights and the logic rule [16]. In this approach, to determine which hyperplanes should be ignored in constructing the decision boundary, a ranked-order distance list of the bisecting hyperplanes is formed in which the



▲ *18. Signal-space decision rule for τ=1 with d=1 coding.*



▲ *19. τ=1 signal-space detector for d=1 code.*



▲ *20. τ=2 look-ahead tree with branches labeled with expected signals.*

hyperplanes are listed in order of increasing distance from the signal pairs that generated the hyperplane. Starting from the bottom of the list, the hyperplanes are removed until the BER performance degrades below the prescribed level. Each time a hyperplane is removed from the list, the logic rule is found anew. An exhaustive search of the space can be used to find the logic rule by processing points on a regular grid through the given set of weights and labeling the resulting region with the binary symbol associated with the closest signal. The ordered distance approach is particularly useful when one is interested in finding a reduced-complexity implementation at the expense of some loss of performance. It is found that using a representative channel response that arises from a high-density disk drive, this approach yields a signal-space detector consisting of 7 hyperplanes that matches the performance of the $\tau=2$ FDTS. For the $\tau=3$ detector, the ranked order distance list starts with 64 hyperplanes, and after the search, the decision boundary can be constructed with 19 hyperplanes without losing performance relative to the original FDTS [16]. The weights of the surviving hyperplanes are often similar, implying that the number of multipliers required to construct the decision boundary can be reduced further. Another systematic method exists for finding reduced-complexity signal-space detectors [44].

The obvious similarity between the structure of Fig. 17 and that of the Adaline or the perceptron suggest that both the weights and the logic rule can be trained using algorithms similar to those used for the neural networks [77]. One may also attempt to introduce a nonlinear element into the structure of Fig. 17 in an effort to create a smoothly shaped decision boundary, which might be more efficient than the piecewise linear boundaries discussed above.

Impact of Constrained Coding

As discussed above, constrained coding is used for a variety of reasons in digital storage channels. In the following we focus on the d-constraint and the MTR constraint. Using specific examples, we show how the signal-space formulation of the FDTS can take advantage of these code constraints and results in highly efficient architectures.

d-constraint

Let us take a specific example of the $d=1$ constraint and the $\tau=1$ FDTS. With the $d=1$ constraint, going from $\tau=0$ to $\tau=1$ provides a very large performance improvement whereas increasing τ beyond 1 yields only marginal performance gain [52]. For this reason, $\tau=1$ is often considered the most attractive choice with the $d=1$ constraint.

With the $d=1$ constraint, the paths that contain two successive symbol changes should be ignored in the decision making process as they violate the code constraint (assuming the NRZ convention). From Fig. 14 we see that path p2 is the illegal path if the previous symbol x_{k-2} was $+1$, whereas path p1 turns out to be the code-violating path if x_{k-2} was -1. Thus, we arrive at two different decision rules according to the previous decision $\hat{x}_{k-2}$. In Fig. 18 line A and line B bisect the signal vector pair p0 and p2 and the pair p1 and p3, respectively. When $\hat{x}_{k-2}=1$, we know that p1 should be discarded, in which case line A is used as the decision boundary. Likewise, line B serves as the decision boundary when p2 is discarded as in the case of $\hat{x}_{k-2}=-1$. Strictly speaking, the decision boundary in either case should also incorporate the line separating the furthest pair, p0 and p3. However, the probability that the given error is caused by the noise moving the furthest signal (p3 in the case of decision boundary A or p0 in the case of B) across the boundary will be asymptotically negligible. Thus, the effective signal margin of this simplified decision rule will be the same as the original FDTS.

The decision rule can be formalized as follows:

For $\hat{x}_{k-2}=1$,

$$\hat{x}_{k-1}=sgn\left(f_1 r_k+r'_{k-1}+f_1\right) \tag{24}$$

For $\hat{x}_{k-2}=-1$,

$$\hat{x}_{k-1}=sgn\left(f_1 r_k+r'_{k-1}-f_1\right) \tag{25}$$

where $sgn(a)=1$ for $a\geq 0$ and -1 otherwise.

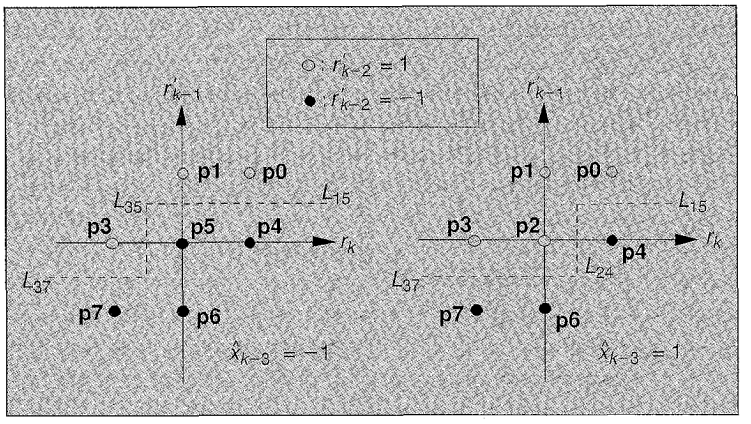
The two separate cases can be incorporated into a single decision rule:

$$\hat{x}_{k-1}=sgn\left(f_1 r_k+r'_{k-1}+f_1\hat{x}_{k-2}\right) \tag{26}$$

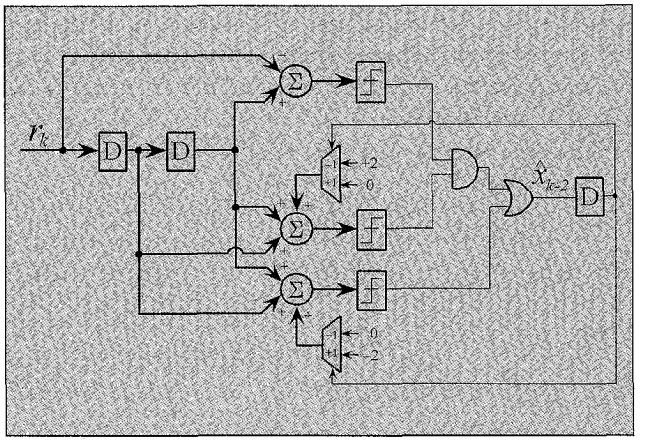
which can be rewritten as

$$\hat{x}_{k-1}=sgn\left(f_1 r_k+r_{k-1}\right) \tag{27}$$

since $r'_{k-1}=r_{k-1}-f_1\hat{x}_{k-2}$. Figure 19 shows the block diagram implementation of this rule. Interestingly, this structure corresponds to a discrete-time matched filter for the truncated channel with the transfer function $1+f_1D$, followed by a threshold detector. A simple signal margin argument shows that this detector indeed achieves asymptotically (i.e., at high SNRs) the optimum performance [57]. It should be pointed out, however, that the matched filter with a threshold detector does not normally provide an optimal detection quality when there is ISI. It is the $d=1$ code constraint that makes this structure optimal even in the presence of significant ISI. The discrete-time matched filter, whose transfer function is given by $D+f_1$, can also be incorporated into the forward and feedback equalizers, as suggested in [42]. This may provide an additional advantage in terms of the processing speed in VLSI implementation. The resulting structure then essentially becomes a DFE, except that the equalizer coefficients are different and that the slicer makes a binary decision on a four-level input signal. This



▲ *21. Cross-sectional views of the 3-d signal space.*



▲ *22. τ=3 signal-space detector with the MTR constraint (assuming f_1=1 and f_2=0).*

structure is known as the multilevel DFE (MDFE). Implementation issues related to the MDFE have been investigated in [11].

For this simplified structure, the worst-case signal separation for $x_k = 1$ and $x_k = -1$ at the slicing point is $2(1+f_1^2)$. The performance advantage of the FDTS/DF over a standard DFE, assuming they are both used for the d=1 constrained channel, can easily be understood. Assuming perfect cancellation of the tail of the channel response via the feedback equalizer, the signal separation for the DFE is 2. This means that the FDTS/DF improves upon the DFE by a factor of $1+f_1^2$ in SNR.

MTR Constraint

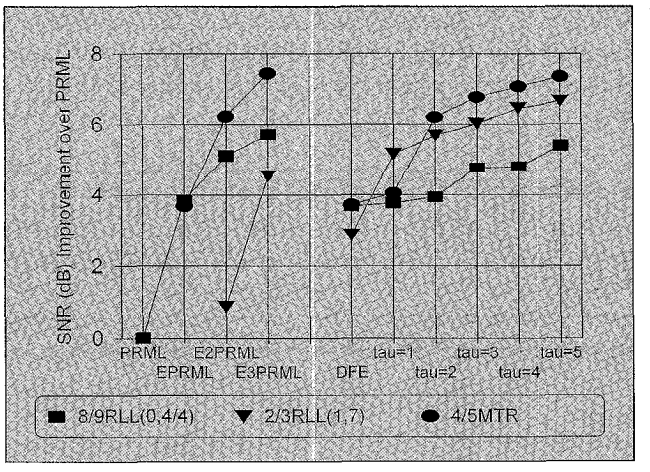
Analysis and simulation results indicate that with the MTR code constraint, the FDTS/DF performance improves by the largest margin when τ increases from 1 to 2 [58]. Figure 20 shows a τ=2 look-ahead tree. The branches are labeled with the corresponding expected signal values, y_k, y'_{k-1}, and y'_{k-2} and the look-ahead paths, denoted by pi's, are expressed as the vectors of the elements y_k, y'_{k-1} and y'_{k-2}. For simplicity, it is assumed that f_1=1 and f_2=0. This assumption makes the placement of the signal points in the 3-dimensional space highly symmetric, which enables the determination of the decision boundary by simple inspection. These assumed coefficient values are also a fairly good approximation to the natural values at high densities. An additional equalization constraint can also be imposed to make this assumption valid without changing the underlying noise statistics significantly. With the additional assumption that $f_0 = 1$, this equalization constraint is referred to as the "110" equalization [15]. It is also straightforward to derive the signal-space detector for arbitrary coefficients f_1 and f_2 using the DT method [59].

The essence of the MTR j=2 code is to prohibit more than two consecutive level changes in the written magnetization pattern. Therefore, depending on the past symbol, x_{k-3}, it is easy to see that either path p2 or path p5 in Fig. 20 should be declared illegal (assuming the NRZ convention) as one of these two paths will always contain three consecutive symbol changes.

Figure 21 shows the cross-sectional views of the 3-dimensional signal space at $r'_{k-2} = 0$. Two cases corresponding to two different previous decisions are shown separately. The black (white) dots represent the paths associated with the decision symbol x_{k-2}=−1 (+1). In each case, the dotted line represents the decision boundary, which is made up of three different planes. In the actual 3-dimensional space, the white dots are placed above the plane of the paper, which corresponds to $r'_{k-2} = 0$, and the black dots below it. Therefore, each plane that makes up the decision boundary is angled at 45 degrees to the plane of the paper. The equation of the plane L_{ij} that bisects the signal vectors $\mathbf{p}_i$ and $\mathbf{p}_j$ is given by

$$L_{ij} = (\mathbf{p}_i - \mathbf{p}_j)'[r_k \; r'_{k-1} \; r'_{k-2}]' - \frac{1}{2}(\mathbf{p}_i + \mathbf{p}_j)'(\mathbf{p}_i - \mathbf{p}_j) \tag{28}$$

where $(\cdot)'$ denotes the vector transpose.



▲ *23. Summary of PRML and FDTS/DF performances with additive noise.*

The equations of the planes necessary to construct the overall decision boundary are given by

$$L_{15} = r_{k-1} + r'_{k-2} - 1 = r_{k-1} + r_{k-2} - 1 - \hat{x}_{k-3} \quad (29a)$$

$$L_{25} = -r_k + r'_{k-2} - 1 = -r_k + r_{k-2} - 1 - \hat{x}_{k-3} \quad (29b)$$

$$L_{37} = r_{k-1} + r'_{k-2} + 1 = r_{k-1} + r_{k-2} + 1 - \hat{x}_{k-3} \quad (29c)$$

$$L_{24} = -r_k + r'_{k-2} + 1 = -r_k + r_{k-2} + 1 - \hat{x}_{k-3} \quad (29d)$$

$L_{ij} \geq 0$ means that the observation vector is closer to $\mathbf{p}_i$ than $\mathbf{p}_j$. Inspecting Fig. 21 and using Eq. (29), we arrive at the following decision rule:

For $\hat{x}_{k-3} = 1$,

$$\begin{aligned}\hat{x}_{k-2} &= \mathrm{sgn}'(L_{15}) \oplus \mathrm{sgn}'(L_{35}) \otimes \mathrm{sgn}'(L_{37}) \\ &= \mathrm{sgn}'(r_{k-1} + r_{k-2}) \oplus \mathrm{sgn}'(-r_k + r_{k-2}) \\ &\quad \otimes \mathrm{sgn}'(r_{k-1} + r_{k-2} + 2)\end{aligned} \quad (30)$$

For $\hat{x}_{k-3} = -1$,

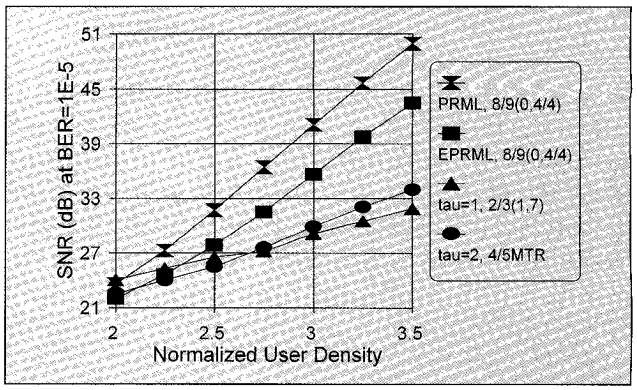
$$\begin{aligned}\hat{x}_{k-2} &= sgn'(L_{15}) \oplus sgn'(L_{24}) \otimes sgn'(L_{37}) \\ &= sgn'(r_{k-1} + r_{k-2}) \oplus sgn'(-r_k + r_{k-2} + 2) \\ &\quad \otimes sgn'(r_{k-1} + r_{k-2})\end{aligned} \quad (31)$$

where $sgn'(a) = 1$ for $a \geq 0$ and $sgn'(a) = 0$ otherwise, and $\oplus$ and $\otimes$ represent modulo-2 addition and multiplication, respectively. (Here, the decision symbol $\hat{x}_{k-2}$ is expressed as binary 1 or 0 to facilitate logic operation, but should be converted to bipolar 1 or −1 to maintain consistency.) The decision rules of Eqs. (30) and (31) can be combined into a single equation:

$$\begin{aligned}\hat{x}_{k-2} &= sgn'(r_{k-1} + r_{k-2} - 1 - \hat{x}_{k-3}) \\ &\quad \oplus sgn'(-r_k + r_{k-2} + 1 + \hat{x}_{k-3}) \\ &\quad \otimes sgn'(r_{k-1} + r_{k-2} + 1 - \hat{x}_{k-3})\end{aligned} \quad (32)$$

whose block diagram implementation is shown in Fig. 22.

The above two examples illustrate the signal-space formulation of the FDTS and the efficient implementation strategies taking into account the $d=1$ and MTR constraint. Although these examples deal with some specific cases of tree depths, they represent important practical situations. The idea presented through the above examples can also be extended to more general coding and tree-depth parameters. For large tree depths, however, methods such as the VoD/DT technique [59] or the systematic partitioning scheme of [44] should be used to construct the decision boundary as the geometric visualization becomes difficult.



▲ *24. SNR versus linear user density.*

Performance Comparison

The BER performances of different code/detector combinations were simulated using random information bit sequences and noise. Theoretical analysis based on some distance argument is possible, but Monte-Carlo simulation provides an accurate picture, especially in the presence of residual ISI and error propagation. In our performance evaluation, the BER was first obtained as a function of readback SNR for each technique. Then, the readback SNR required to achieve a target BER of 10^{-5} was recorded for each code/detector combination. The readback SNR is defined as

$$SNR = \frac{peak\ amplitude\ of\ isolated\ step\ response}{RMS\ noise\ in \left(0 - \frac{1}{PW50}\right)\ frequency\ band} \quad (33)$$

where $PW50$ denotes the width of the isolated step response at the 50% height. The required readback SNRs were then used as a basis of performance comparison. The simulations were run at different user densities. The user density, D_u, is defined as

$$D_u = \frac{PW50}{user\ bit\ period} \quad (34)$$

A modeled MR response was used to construct the readback signals. Both PRML and FDTS/DF techniques were simulated. Large enough FIR filters were used for equalization for both techniques so that the performances were not degraded by imperfect equalization. In all cases, the equalizer taps were obtained based on the minimum squared error criterion.

Figure 23 summarizes the simulation results taken at $D_u=2.5$ with additive Gaussian noise. The density $D_u=2.5$ is considered a high density (current high-end products operate at around $D_u=2$ –2.5). PRML and FDTS/DF techniques are compared and the performance is plotted in terms of the SNR improvement over the PR4ML, which is considered today's industry standard. The PR targets take the form $(1-D)(1+D)^n$ with n corresponding to 1, 2, 3, or 4 (denoted by PR4ML,

EPRML, E2PRML, or E3PRML, respectively). The FDTS/DF schemes are shown for different τ values ($\tau=0$ corresponds to DFE). For each detector, three different codes—the rate 8/9 (0,4/4) code, the rate 2/3 (1,7) code, and the rate 4/5 MTR $j=2$ code—are considered (the I constraint plays no role in high-order PRMLs, which are not interleavable, and can simply be ignored). All techniques shown yield performance improvements over the PR4ML. The E3PRML and the $\tau=5$ FDTS/DF, both with the MTR code, achieve the best performance. These techniques also require the highest complexity. In the PRML family, the next best performance comes from the E2PRML combined with the MTR code. A similar performance is obtained from the $\tau=2$ FDTS/DF, again with MTR coding, but with a much lower implementation complexity. The performance improvements of the E2PRML and the $\tau=2$ FDTS/DF with MTR coding are more than 6 dB over the PR4ML with the (0,4/4) code. Note that the $\tau=2$ FDTS/DF can also be implemented in signal space with a very low hardware complexity and with practically no loss of performance by utilizing the "110" forward equalization, as described earlier.

While the FDTS/DF shows a step improvement as τ increases from 1 to 2 with the MTR code, it exhibits a similar behavior as τ goes from 0 to 1 with the (1,7) code. In fact, the $\tau=1$ FDTS/DF combined with the (1,7) code, which can be implemented efficiently as MDFE, performs nearly as well as the E2PRML with the (0,4/4) code, which is considerably more complex. The (1,7) code is a poor choice for PRML techniques since the high symbol density associated with the code results in large mismatch between the natural channel and the target, which in turn causes a significant equalization loss. For the FDTS/DF schemes, however, the (1,7) code yields better results than the $(0,k)$ code, despite the rate disadvantage, except in the case of $\tau=0$ or DFE. This coding gain is realized as the worst-case error events in the FDTS/DF are eliminated by the $d=1$ code constraint.

The role of advanced detection schemes becomes more critical as the demand for higher user density continues. It is important to see how much improvement in detector error performance is achieved by using such advanced methods as the density increases. Figure 24 shows a plot of the readback SNR required to obtain a fixed BER of 10^{-5} for selected detectors versus user density. The noise is additive Gaussian. The results show that FDTS/DF combined with the $d=1$ or the MTR code outperform the PR4ML and EPRML techniques used in conjunction with the (0,4/4) code. The relative improvements become larger as linear density increases. This trend can be explained by the coding gain of the $d=1$ and MTR codes, which are more pronounced at high linear densities, along with the advantage of the DFE-type equalization, which results in a considerably smaller noise boost than the fixed target systems like the PRML techniques. We finally note that there exist a rate 16/17 $(0,G/I)$ code and a rate 6/7 MTR $j=2$ code, which provide some rate advantages compared to the codes considered here.

Other Considerations

Although nonlinearities and transition noise encountered in high-density magnetic channels have been discussed, the performance study presented here does not include these effects. The amount of nonlinearity present in the channel certainly depends on the minimum transition spacing allowed by the code. To investigate the impact of nonlinearities on a particular choice of code, understanding of nonlinearities on a quantitative level is required. Accurate modeling of transition noise as well as investigation of its impact on code/detector performance are also topics of ongoing research.

Understanding the error-propagation behavior is important for practical application of decision-feedback techniques such as the DFE and FDTS/DF. While the BER simulation results presented here counted the errors caused by propagation of previous errors, more thorough simulation and analysis can be done to understand the statistics on the lengths of the error events by, for example, invoking a Markov process model for error propagation [34, 69]. This kind of approach will be useful in designing more suitable error-correcting codes. There also exist suboptimal detectors based on the combination of the DFE and Viterbi detection [24, 26, 50]. The performance is expected to be similar to that of the FDTS/DF, but the error-propagation property may improve at the expense of increased complexity.

An important channel impediment that has not been considered in this article is offtrack interference, which arises as the signal from neighboring data tracks interfere with the main track data [1]. Although in present commercial systems the effect of offtrack is minimized by the "write wide and read narrow" approach, offtrack interference will likely be one of the major noise sources in future storage systems where high track densities are inevitable. In tape recorders, time-varying separation of the head-tape space causes occasional deep fading of the signal amplitude, resulting in long bursts of errors [80]. These types of channels may require different signal-processing strategies. Among other important practical issues that have not been considered here are timing recovery and track servo signal processing.

Summary

The read/write process of magnetic recording has been viewed as a PAM communication channel subject to severe ISI. Some of the unique noise and nonlinear characteristics encountered in the recording channel have been described. Modulation-coding and symbol-detection strategies suitable for this channel also have been discussed. The FDTS/DF and related signal-space detectors have been described in detail. As density increases, the

FDTS/DF technique, when combined with an appropriate modulation code, exhibits significant performance improvement over the PRML technique. With certain types of codes, the signal-space implementation of the FDTS detector offers highly attractive complexity and performance tradeoffs.

Jaekyun Moon is an Associate Professor at the University of Minnesota's Department of Electrical and Computer Engineering in Minneapolis, Minnesota, USA.

References

1. W.L. Abbott, J.M. Cioffi and H.K. Thapar, "Offtrack interference and equalization in magnetic recording," *IEEE Trans. Mag.*, vol. 24, no. 6, pp. 2964–2966, Nov. 1988.
2. K. Abend and B.D. Fritchman, "Statistical detection for communication channels with intersymbol interference," *Proc. IEEE*, vol. 58, no. 5, pp. 779–785, May 1970.
3. R.D. Barndt, A.J. Armstrong, N. Bertram, and J.K. Wolf, "A simple statistical model of partial erasure in thin film disk recording systems," *IEEE Trans. Magnet.*, vol. 27, no. 6, Nov. 1991.
4. L. Barbosa, "A model for magnetic recording channels with signal dependent noise," *IEEE Trans. Mag.*, vol. 31, no. 2, pp. 1062–1064, March 1995.
5. C.A. Belfiore and J.H. Park, Jr., "Decision feedback equalization," *Proceedings of the IEEE*, vol. 67, no. 8, pp 1143–1156, Aug. 1979.
6. N.R. Belk, P.K. George, and G.S. Mowry, "Noise in High Performance Thin-film Longitudinal Magnetic Recording Media," *IEEE Transactions on Magnetics*, Sept. 1985.
7. J.W.M. Bergmans, "Density improvements in digital magnetic recording by decision feedback equalization," *IEEE Trans. Magnet.*, vol. MAG-22, no.3, pp. 157–162, May 1986.
8. J.W.M. Bergmans, S.A. Rajput and F.A.M. van der Laar, "On the use of decision feedback for simplifying the Viterbi detector," Philips J. Res., vol. 42, no. 4, pp. 399–428, 1987.
9. J.W.M. Bergmans, J.O. Voorman and H.W. Wong-Lam, "Dual decision feedback equalizer," Submitted to *IEEE Trans. Comm.*, 1996.
10. J.W. Bergmans, "Digital Baseband Transmission and Recording," KAP, 1996.
11. L. Bi, J. Hong, Y. Lee, H. Mutoh, Q. Sun, H. Ueno, J. Wang and R. Wood, "An experimental MDFE detector," presented at the International Conference on Magnetics, April, 1997.
12. W. Bliss and L. Sundell,"Performance of Generalized Maximum Transition run Trellis codes," *IEEE Trans. Magn.*, vol. 34, no. 6, Jan. 1998.
13. N.K. Bose and A.K. Garga, "Neural network design using Voronoi diagrams," *IEEE Trans. Neural Networks*, vol. 4, no. 5, pp. 778–787, Sept. 1993.
14. K.C. Bracken and L.R. Carley, "BiCMOS analog arbitration circuits," Proceedings of the Bipolar/BiCMOS Circuits and Technology Meeting, pp. 246–249, Oct 1992.
15. B. Brickner and J. Moon, "A high dimensional signal space implementation of FDTS/DF," *IEEE Trans. Magn.*, vol. 32, no. 5, pp. 3941–3943, Sept. 1996.
16. B. Brickner and J. Moon, "Architectures for the implementation of a fixed delay tree search detector," *IEEE Trans. Magn.*, vol. 33, no. 2, March 1997.
17. B. Brickner and J. Moon, "Design of a rate 6/7 maximum transition run code," *IEEE Trans. Magn.*, vol. 33, no. 5, Sept. 1997.
18. L.R. Carley, K.C. Bracken, R. Mittal and J. Park, "A low-power analog sampled-data VLSI architecture for equalization and FDTS/DF detection," *IEEE Trans. Mag.*, vol. 31, no. 2, pp. 1202–1207, March 1995.
19. J.M. Cioffi, W.L. Abbott, H.K. Thapar, C.M. Melas, and K.D. Fisher, "Adaptive equalization in magnetic-disk storage channels," *IEEE Commun. Mag.*, pp. 14–29, Feb. 1990.
20. J.M. Cioffi, G.P. Dudevoir, M.V. Eyuboglu, and G.D. Forney, Jr., "MMSE decision-feedback equalizers and coding," *IEEE Trans. Commun.*, vol. 43, pp. 2582–2604, Oct. 1995.
21. R.D. Cideciyan, F. Dolivo, R. Hermann, W. Hirt, and W. Schott, "A PRML system for digital magnetic recording," *IEEE JSAC*, vol.10, no.1, pp. 38-56, Jan. 1992.
22. A.P. Clark, L.H. Lee and R.S. Marshall, "Developments of the conventional nonlinear equalizer," *Proc. IEE*, vol. 129, part F, no. 2, pp. 85–94, Apr. 1982.
23. E. Dahlman and B. Gudmundson, "Performance improvement in decision feedback equalizers by using "soft decision," *Electronics Lett.*, vol. 24, no. 17, pp. 1084–1085, Aug. 1988.
24. A. Duel and C. Heegard, "Delayed Decision-feedback sequence estimation," *IEEE Trans. Commun.*, vol. 37, pp. 428–436, May 1989.
25. E. Eleftheriou and W. Hirt, "Improving performance of PRML/EPRML through noise prediction," *IEEE Trans. Mag.*, vol. 32, no. 5, pp. 3968–3970, Sept. 1996.
26. M.V. Eyuboglu and S.U.H. Qureshi, "Reduced-state sequence estimation for coded modulation on intersymbol interference channels," *IEEE JSAC*, vol. 7, no. 6, pp. 989–995, Aug. 1989.
27. M.V. Eyuboglu and G. D. Forney, Jr., "Combined equalization and coding using precoding," *IEEE Commun. Mag.*, vol. 29, no. 12, pp. 25–34, Dec. 1991.
28. M.J. Ferguson, "Optimal reception for binary partial response channels," *Bell Syst. Tech. J.*, vol. 51, no. 2, pp. 493–505, Feb. 1972.
29. K. Fisher, J. Cioffi, W.A. Abbott, P.S. Bednarz and C.M. Melas, "An adaptive RAM-DFE for storage channels," *IEEE Trans. Commun.*, vol. 39, no. 11, pp. 1559–1568, Nov. 1991.
30. K.D. Fisher, W.L. Abbott, J.L. Sonntag and R. Nelson, "PRML detection boosts hard-disk drive capacity," *IEEE Spectrum*, pp 70–76, Nov. 1996.
31. K.K. Fitzpatrick, "Time-varying MTR Codes for High Density Magnetic Recording," Global Telecommunications Conference, Phoenix, Arizona, November 3–8, 1997
32. G.D. Forney, Jr., "The Viterbi algorithm", *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.
33. H. Harashima and H. Miyakawa, "Matched-Transmission technique for channels with intersymbol interference," *IEEE Trans. Comm.* vol. COMM-20, August 1972.
34. K. Han and R. Spencer, "Comparisons of different detection techniques for digital magnetic recording channels," *IEEE Trans. Mag.*, vol. 31, no. 2, pp. 1128–1133, March 1995.
35. R. Hermann, "Volterra modeling of digital magnetic saturation recording channels," *IEEE Trans. Magnet.*, vol. 26, no. 5, pp. 2125–2127, Sept. 1990.
36. T.D. Howell, D.P. McCown, T.D. Diola, Y.-S. Tang, K.R. Hense, and R.L. Gee "Error rate performance of experimental gigabit per square inch recording components," *IEEE Trans. Magnet.*, Sept. 1990.
37. K.A.S. Immink, "Coding methods for high-density optical recording," Philips J. Res., vol. 41, pp. 410–430, 1986.
38. K.A.S. Immink, "Coding techniques for the noisy magnetic recording channel," *IEEE Trans. Comm.*, vol. 37, no .5, pp. 413–419, May 1989.
39. R. Karabed and P.H. Siegel, "Matched spectral-null codes for partial-response channels," *IEEE Trans. Info. Theory*, vol. 37, no. 2, pp. 818-855, May 1991.
40. R. Karabed and P.H. Siegel, "Coding for high-order partial response channels," SPIE Conference, Philadelphia, PA, Oct. 1995.
41. J.G. Kenney and L.R. Carley, "Geometric representation of the tree-search detector," Int. Conf. Commun., Chicago, Illinois, June 1992.

42. J.G. Kenney, L.R. Carley and R.W. Wood, "Multi-level decision feedback equalization for saturation recording," *IEEE Trans. Magnet.*, vol. 29, no. 3, pp. 2160–2171, July 1993.

43. G. Kerwin, R. Galbraith, and J. Coker, "Performance evaluation of the disk drive industry's second-generation PRML data channel," *IEEE Trans. Magnet.*, Sept. 1993.

44. Y. Kim and J. Moon, "Multi-dimensional signal space partitioning using a minimal set of hyperplanes," submitted to the *IEEE Transactions on Communications*, Feb. 1998.

45. H. Kobayashi, "Application of probabilistic decoding to digital magnetic recording systems," *IBM J. Res. Develop.*, vol.15, pp.65–74, Jan. 1971.

46. Y. Lin and R. Wood, "An estimation technique for accurately modeling the magnetic recording channel including nonlinearities," *IEEE Trans. Mag.*, vol. 25., no. 5, pp. 4084–4086, Sept. 1989.

47. B.H. Marcus, P.H. Siegel, J.K. Wolf, "Finite-state modulation codes for data storage," *IEEE JSAC*, vol.10, no.1, pp. 5–37, Jan. 1992.

48. H. Matui, "Adaptive reduced-state sequence estimation for linearly and nonlinearly distorted signals in magnetic recording channels," *IEEE Trans. Magnet.*, Nov. 1993.

49. C.M. Melas, P.C. Arnett, I.A. Beardsley, and D. Palmer, "Nonlinear superposition in saturation recording of disk media," *IEEE Trans. Magnet.*, vol. MAG-23, no. 5, pp. 2079–2081, Sept. 1987.

50. C.S. Modlin, K.D. Fisher, and J.M. Cioffi, " An analysis of the RAM-RSE read channel," presented at the 1995 Int. Conf. Mag., San Antonio, Texas, April 1995.

51. J. Moon, L.R. Carley, and R.R. Katti, "Density Dependence of Noise in Thin Metallic Longitudinal Media," *J. of Applied Physics*, April 1988.

52. J. Moon and L.R. Carley, "Performance comparison of detection methods in magnetic recording," *IEEE Trans. Magnet.*, vol. 26, no. 6, pp. 3155–3172, Nov. 1990.

53. J. Moon, "Discrete-time modeling of transition-noise-dominant channels and study of detection performance," *IEEE Trans. Magnet.*, vol. 27, no. 6, pp. 4573–4578, Nov. 1991.

54. J. Moon, "Signal-to-noise ratio degradation with channel mismatch," *IEEE Trans. Magnet.*, vol. 27, no. 6, Nov. 1991.

55. J. Moon and J.-G. Zhu, "Nonlinearities in thin-film media and their impact on data recovery," *IEEE Transactions on Magnetics*, vol. 29, No. 1, Jan. 1993.

56. J. Moon and L.R. Carley, "Efficient sequence detection for intersymbol channels with runlength constraints," *IEEE Trans. Comm. , Sept. 1994.*

57. J. Moon and S. She, "Constrained-complexity equalizer design for fixed delay tree search with decision feedback," *IEEE Transactions on Magnetics,* vol. 30, no. 5, Sept. 1994.

58. J. Moon and B. Brickner, "Maximum transition run codes for data storage systems," presented at the International Magnetics Conference, Seattle, Washington, April 1996.

59. J. Moon and T. Jeon, "Sequence detection for ISI channels using signal space partitioning," to appear in *IEEE Transactions on Communications*, 1998.

60. S. Nair and J. Moon, "Data storage channel equalization using neural networks," *IEEE Transactions on Neural Networks*, vol. 8, no. 5, Sept. 1997.

61. D. Palmer, P. Ziperovich, R. Wood, and T.D. Howell, "Identification of nonlinear write effects using pseudorandom sequences," *IEEE Trans. Magnet.*, vol. MAG-23, no. 5 pp. 2377–2379, Sept. 1987.

62. A.M. Patel, "A new digital signal processing channel for data storage products," *IEEE Trans. Magnet.*, vol.27, no. 6, Nov. 1991.

63. A. Patel, R. Rutledge, B. So, "Six-sample look-ahead (1,7) ML detection channel and performance data," *IEEE Trans. Magnet.*, Sept. 1993.

64. R. Price, "Nonlinearly-equalized PAM vs. capacity for noisy filter channels," *Proc. 1972 IEEE Int. Conf. Commun.*, pp. 22.12–22.17, June 1972.

65. J.G. Proakis and A. Khazen-Terezia, "A decision-feedback tree-search algorithm for digital communication through channels with intersymbol interference," *Proc. of Int. Conf. Commun.*, June 1986.

66. P.H. Siegel, "Applications of a peak detection channel model," *IEEE Transactions on Magnetics*, vol. MAG-18, no. 6, pp. 1250–1252, Nov. 1982.

67. P.H. Siegel, "Recording codes for digital magnetic storage," *IEEE Transactions on Magnetics*, vol. MAG-21, no. 5, pp. 1344–1349, Sept. 1985.

68. P.H. Siegel and J.K. Wolf, "Modulation and coding for information storage," *IEEE Commun. Mag.*, vol. 29, no. 12, pp. 68–86, Dec. 1991.

69. G. Tamburelli, "On the exact error probability evaluation of a decision feedback and feedforward receiver for finite impulse response (FIR) channels," *Proc. Nat. Telecommun. Conf.*, Dec. 1977.

70. H. Thapar and A.M. Patel, "A class of partial response systems for increasing storage density in magnetic recording," *IEEE Trans. Magnet.*, vol. 23, no. 5, Sept. 1987.

71. H. Thapar, P. Siegel, B. Shung, J. Rae, and R. Karabed, "Trellis-coded partial response (TCPR): an improved recording method over PRML," *IEEE Trans. Magnet.*, Sept. 1993.

72. M. Tomlinson, "New automatic equalizer employing modulo arithmetic," *Electronics Letters*, vol. 7, March 1971.

73. K.P. Tsang, "Method and apparatus for implementing run length limited codes in partial response channels," US patent No. 5,537,112, July 16, 1996.

74. K.P. Tsang and B. Rub, "Maximum transition run length codes with location dependent constraints," U.S. Patent Application.

75. D.J. Tyner and J.G. Proakis, "Partial response equalizer performance in digital magnetic recording channels," *IEEE Trans. Mag.*, vol. 29, no. 6, pp. 4194–4208, Nov. 1993.

76. J. Uddenfeldt and L.H. Zetterberg, "Algorithms for delayed encoding in delta modulation with speech-like signals," *IEEE Transactions on Communications*, vol. COM-24, pp. 652–658, June 1976.

77. B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and back propagation," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415–1442, Sept. 1990.

78. D. Williamson, R.A. Kennedy, and G.W. Pulford, "Block decision feedback equalization," *IEEE Transactions on Communications*, vol. 40, no. 2 pp. 255–264, Feb. 1992.

79. J.K. Wolf, "A survey of codes for partial response channels," *IEEE Trans. Mag.*, vol. 27, no. 6, pp. 4585–4589, Nov. 91.

80. R.W. Wood and R.W. Donaldson, "The helical-scan magnetic tape recorder as a digital communication channel," *IEEE Trans. Mag.*, vol. MAG-15, no. 2, pp. 935–943, March 1979.

81. R.W. Wood and D.A. Petersen, "Viterbi detection of class IV partial response on a magnetic recording channel," *IEEE Trans. Commun.*, vol. COM-34, no. 5, pp. 454–461, May 1986.

82. R.W. Wood, "Turbo-PRML: A compromise EPRML detector," *IEEE Trans. Mag.*, vol. 29, no. 6, pp. 4018–4020, Nov. 1993.

83. T. Yamauchi and J.M. Cioffi, "A nonlinear model for thin film disk recording systems," *IEEE Trans. Mag.*, vol. 29, no. 6, pp. 3993–3995, Nov. 1993.

84. W. Zeng and J. Moon, "Modified Viterbi Algorithm for a Jitter-Dominant 1-D^2 Channel," *IEEE Transactions on Magnetics*, vol. 28, No. 5, Sept. 1992.

85. W. Zeng and J. Moon, "A practical nonlinear model for magnetic recording channels" *IEEE Transactions on Magnetics,* vol. 31, no. 2, Nov. 1994.

86. W. Zeng and J. Moon, "Decision feedback equalizer with pattern dependent threshold," *IEEE Transactions on Magnetics,* vol. 32, no. 4, pp. 3266-3273, July 1996.