Modulation and Coding for Information Storage

The success of certain modulation and coding techniques in data communications have inspired promising new applications in digital data storage.

by Paul H. Siegel and Jack K. Wolf

The communications industry is concerned with reliable and efficient means for transmitting information from one place to another. The information storage industry is concerned with reliable and efficient means for transporting information from one time to another. The teachings of information and communications theory apply equally well to both scenarios. In communications as well as storage systems, information must be transported through noisy channels that accept input signals (perhaps from some constrained class of signals) and produce output signals from which the transmitted or stored information must be recovered. Yet the readers of this magazine are, for the most part, unaware of the applications of their skills to the storage industry, which has worldwide annual sales in excess of $50 billion. (This figure is for magnetic storage alone.)

While communicators are concerned with maximizing the rate (in bits per second) whereby digital information can be transmitted and reliably received, storage researchers are largely concerned with maximizing the areal density (in bits per square inch) and volumetric density (in bits per cubic inch) for storing and reliably retrieving information.

The storage industry has made steady progress increasing the density of information storage of digital data. Over the last 25 years, the areal density of storage of digital data on magnetic hard disks has grown geometrically at a compound growth rate of about 29%. This remarkable growth is reflected in Fig. 1, which shows the areal density of high-end IBM disk drives as a function of production year.

Most of this increased storage density has resulted from improvements in the part of the system that we call "the channel," including the storage medium itself, the read and write heads, mechanical features determining the flying heights and positioning of these heads, and so on.

Though less dominant, the contribution of advances in signal processing and coding has not been insubstantial. For example, channel modeling indicates that without the improvements in run length-limited codes (described in more detail later) the compound growth rate in areal density would have been approximately 24 percent. If we restrict attention to linear density gains, the modeling shows that progress in signal processing and coding technology alone has accounted for almost a quadrupling of the linear density achievable with a "typical" set of recording components.

As communications theorists, we define the "channel" as the part of the system over which we have no control. As such, we assume that the channel is fixed. Although read/write engineers for storage systems may not have direct control over the choice of the channel, the systems that make up the channel are constantly being improved and these improvements have been the principle reason for the growth in information density in Fig. 1. Thus, instead of utilizing modern modulation and coding systems that yield performance closer to the channel capacity, the designers of these systems have taken the alternative (and probably harder) approach of increasing the channel capacity itself.

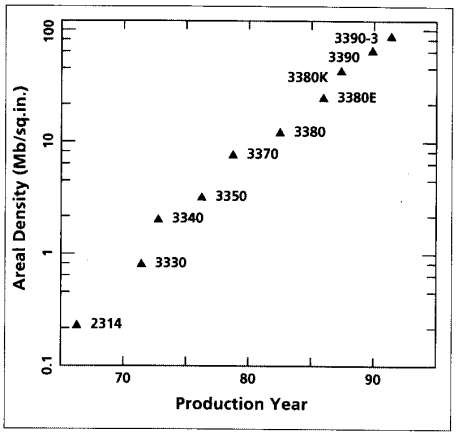
As a matter of fact, the modulation and coding systems used in most of today's products do not differ markedly from the modulation and coding systems used in products a decade ago. As we shall see, the modulation and coding systems in these products were chosen to match a particular type of detector, called a peak detector, which has been an integral part of the system. The peak detector has seen a remarkable life. However, we may be at a turning point, where the peak detector and its associated modulation/coding/signal processing systems could be replaced by a sampling detector and an entirely new type of modulation/coding/ signal processing system. The new system looks very much like the systems that communications engineers are accustomed to seeing in advanced communications systems: partial-response equalization, trellis codes, Viterbi detection, adaptive digital filtering, and so on. Both the old and new systems will be described in this paper.

Although many different types of media can be utilized for the storage of digital data, we concentrate in this paper on systems based upon tra-

Paul H. Siegel is manager, Signal Processing and Coding project, IBM Almaden Research Center, San Jose, California.

Jack Keil Wolf is a chaired professor of Electrical Engineering and Computer Engineering and a member of the Center for Magnetic Recording Research at the University of California, San Diego.

0163-6804/91/$01.00 1991© IEEE



■ **Figure 1.** *Storage density versus time*

ditional magnetic recording. The techniques described, however, have applications in other types of storage systems such as those using magnetic-optic and optical recording.

One important difference between communication systems and storage systems is the requirement on decoded error rate. Often in communication systems, the goal is a user error rate of 10^{-5} or 10^{-6}. Storage systems, however, often require error rates of 10^{-12} or better. Furthermore, implementations of error recovery procedures and their impact on the performance measures of storage devices have often mandated that there be a strict requirement on the "raw" error rate at the output of the channel before the error correction decoder.

Channel Model

The "guts" of a magnetic recording system are: the write head, the magnetic medium, and the read head. (The write head could be the same as the read head and usually has been for disk drives.) The write head is driven by a current source that carries the information to be stored. The write head radiates flux, which changes the state of magnetization of the magnetic medium immediately under the head. Actually, since the head is moving with respect to the magnetic medium, any point on the magnetic medium retains the state of magnetization corresponding to the last flux it experienced from the write head as the head moves away from that point.

On a rigid disk, the disk moves in a circular motion under the head. Information is stored on the disk in concentric tracks, the width of a track roughly being governed by the size of the write head. The density of recording is then the product of the number of tracks per inch (tpi) and the linear density of information along a track measured in bits per inch (bpi). Typical numbers for today's high end (i.e., expensive) rigid disk drives are: 3,000 tpi and 30,000 bpi.

There are at least two types of magnetic tape systems. In the first type, the head (or heads) remains stationary while the tape is pulled over the head. In the second, called a rotary head system, the head (or heads) is fastened to a spinning drum while the tape is moved slowly past the drum. This type of system is used in videocassette recording and other applications where larger bandwidth is required. The head-to-tape speed in the first type is governed by the speed of the moving tape, while in the second type, the head-to-tape speed is mostly a function of the rotational speed of the drum and not the speed of the tape past the drum. In multiple write head tape systems, information usually is written simultaneously on many tracks, while in rigid disk storage systems, a single head almost always writes information on a single track. Rigid disk systems usually have a single head for both writing and reading (or, as in some recently announced products, write head physically coupled with a read head), while tape systems may have a separate read head so that the system can read what is being written.

The current into the write head induces a magnetization pattern on the track immediately below the write head. When a track is to be read, a read head is positioned over the track. Then, the magnetization pattern "frozen" on that track radiates flux that is sensed, or "read," by the read head. The read head produces a voltage that is symptomatic of the magnetization on the track being read. There are primarily two types of read heads: inductive heads which contain coils of very fine wire and which produce a voltage proportional to the time derivative of the flux that passes through its coils, and magneto-resistive (MR) heads which produce a voltage directly proportional to the flux sensed by the head. MR heads produce larger read voltages than inductive heads, but have a limited dynamic range for linear operation. Only inductive heads have been used for writing, to this date.

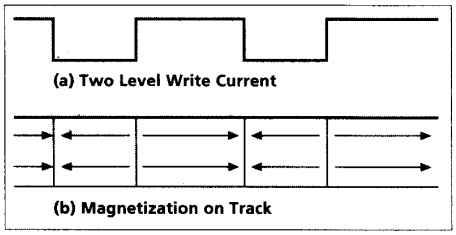
In rigid disk systems there is a separation between the head and the disk called the air bearing. This separation is extremely small and the slightest imperfection on the surface of the disk (or any contaminant in the air bearing) could cause the head to "crash." Tape systems have the head (or heads) in contact with the magnetic surface. Because of the roughness of the tape, there is actually a nonzero average separation between the head (or heads) and the tape.

The magnetic recording channel is inherently nonlinear because of the hysteresis that affects all magnetic media. However, magnetic recording specialists have found a way of linearizing the channel for a restricted range of inputs. They constrain the current into the write head to take on only two possible values, for example, +A and –A, where the amplitude A is chosen sufficiently large so as to completely magnetize the magnetic storage medium in one of two directions. Thus, the hysteresis effect can be ignored. This type of recording is called saturation recording, and all practical digital storage devices use this approach. After information has been written on a track using saturation recording, the magnetic medium on that track would be alternately magnetized along the track, either in the direction of rotation or the opposite direction. A two-level write current waveform and its corresponding magnetization pattern on a track are shown schematically in Fig. 2.

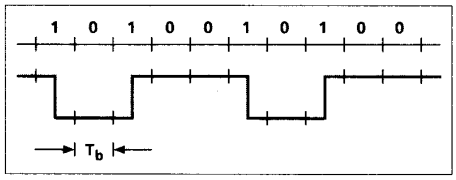
Since the write current can only take on the values +A and –A, the stored information is represented by the times when transitions occur. Usually, one assumes that the time interval assigned to each channel bit is fixed. We call the

The magnetic recording channel is inherently nonlinear because of the hysteresis that affects all magnetic media.

A major problem affecting the read signal is the time-varying gain of the channel.



■ **Figure 2.** *Saturation recording*



■ **Figure 3.** *NRZI recording*

duration of a channel bit T_b. In one convention called NRZI modulation, a channel bit equal to "1" is written as a transition in the write current (from +A to −A or vice versa) in the middle of a bit cell, and a channel bit equal to 0 is written as no transition in the write current (i.e., the write current in the present bit cell remains at the same level as at the end of the previous bit cell). A representative sequence of channel bits and the corresponding write current after NRZI modulation are shown in Fig. 3.

Assume that the write current is initially equal to +A and that transitions in the write current occur at times $j_1T_b, j_2T_b, \ldots, j_iT_b$, where $j_1, j_2, \ldots j_i$ are integers such that $j_1 < j_2 < \ldots < j_i$. Let $u(t)$ be a unit step that occurs at $t = 0$,

$$u(t) = \begin{cases} 0 \text{ if } t < 0, \\ 1 \text{ if } t \geq 0. \end{cases}$$

Then the write Current, $I(t)$, can be written as:

$$I(t) = A + 2A \sum_{i=1}^{\infty} (-1)^i u(t - j_i T_b).$$

Let $g(t)$ denote the read voltage corresponding to an isolated positive-going transition of the write current (from $-A$ to $+A$) occurring at time $t = 0$. Then, if the channel were linear, the output voltage, $V(t)$, corresponding to the magnetization induced by the write current $I(t)$ would be given as [1]:

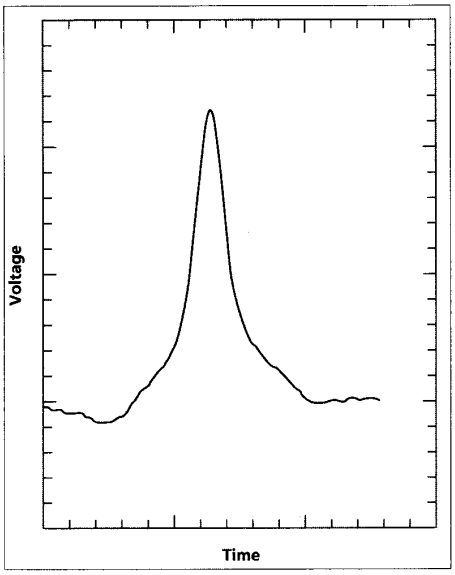
$$V(t) = A \sum_{i=1}^{\infty} (-1)^i g(t - j_i T_b).$$

A plot of a typical $g(t)$ taken from a thin film disk with thin film (inductive) heads is shown in Fig. 4.

A common mathematical model for such an isolated transition response is the Lorentzian pulse shape given by the formula:

$$g(t) = \frac{1}{1 + \left(\frac{2t}{PW_{50}} \right)^2}.$$

The constant denoted by PW_{50} is a parameter that represents the pulse width of the transition



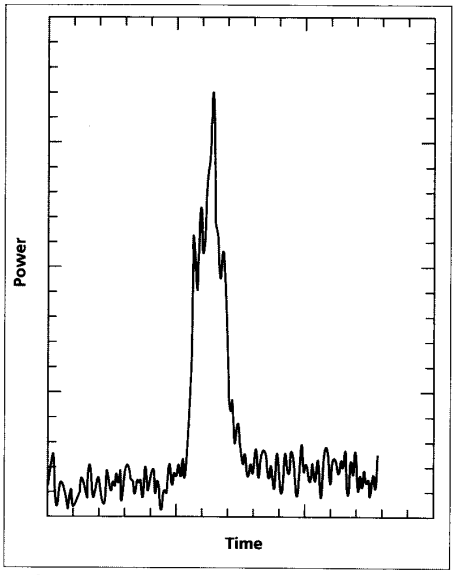
■ **Figure 4.** *Isolated transition response from a thin film disk and thin film lead*

response at 50 percent of the maximum amplitude. The advantage of using this admittedly imperfect model is that it portrays the shape of the isolated transition response reasonably well, and requires only the single parameter PW_{50}.

The preceding discussion assumes the validity of using linear superposition of the isolated transition responses to form the composite output due to many transitions. This is, again, only a rough approximation to what actually occurs because of the interaction of the transition to be written with those that have already been written. The new transition could be very close to the previous one, and the magnetic flux "radiated" from the medium where the previous transition was written influences what currently is being written. This flux, called the demagnetization field, pulls the transition to be written closer to the previous transition. We will neglect nonlinear effects in this paper, but one must bear in mind that these cannot be completely ignored in the practical design of high density systems.

Another major problem affecting the read signal is the time-varying gain of the channel. This can be due to many phenomena, including variation in the spacing between the read head and the medium over time, or the presence of physical defects in the medium itself. In disk systems, a map of each disk surface is usually made, identifying portions of the disk containing defects so that these portions will not be used.

So far we have not discussed the effects of noise. It is common to assume that the noise in the system is additive and Gaussian. Usually, it is also assumed that the noise consists of two components: a Gaussian white noise component due to the electronics on the read (i.e., receiver) side, and a Gaussian colored component due to the medium. The spectral characteristics of this colored noise are essentially the same as would be obtained from passing white noise through the linear transfer function characterizing the system. More complicated models for the noise exist, particu-



■ **Figure 5.** *Noise variance as a function of the position in a pulse*

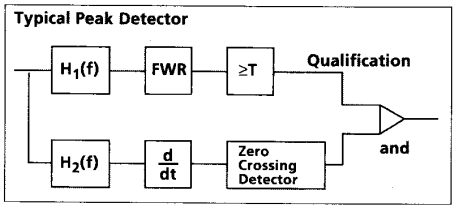
larly for thin film media, where it has been shown, for example, that there is more noise at the center of a transition than where there is no recorded transition. A plot of the variance of this noise as a function of position within an isolated transition response for a representative thin film medium is shown in Fig. 5. This signal-dependent noise leads to opportunities for improved detectors, but this advanced subject is still exploratory and lies beyond the scope of this paper.

Peak Detection Systems

The design of the modulation, coding and signal processing in past magnetic recording products has been driven by the detector chosen to detect the transitions in the channel input waveform. This detector, called a peak detector [2], has the advantage of being both robust and extremely simple to implement. However, by its very nature, it works best at low linear densities. A block diagram of a typical peak detector is shown in Figure 6.

There are two paths through the detector. The top path is used to "qualify" a peak, i.e., to ensure that the peak has sufficient amplitude. This path consists of a linear filter, $H_1(f)$, a full wave rectifier (FWR), and a threshold testing circuit. The bottom path is used to locate the peak by differentiating the signal after the linear filter $H_2(f)$ and then passing the differentiated signal through a zero-crossing detector. The detector only accepts a peak if the peak amplitude was large enough to pass the qualification test.

Once a peak is detected by the peak detector, it is thought to be due to a transition in the input waveform. A device called a phase-lock loop (PLL) is used to derive timing from the position of the detected peaks. The PLL produces a clock of period T_b seconds by which to identify channel bit intervals (sometimes called "bit cells"). Then, if an output pulse is located in a bit interval, that bit interval is said to contain a transition. Using



■ **Figure 6.** *Block diagram of a peak detector*

the NRZI precoding convention of Fig. 3, a bit interval with a transition corresponds to a recorded "1", and an interval without corresponds to a "0."

Note that the use of the NRZI precoder ensures that the reconstruction of the recorded sequence is insensitive to polarity inversion of the channel output waveform: a peak, regardless of its polarity, corresponds to a "1", and the absence of a peak corresponds to a "0." From another perspective, if the detector had to recover the actual recorded write current from the correspondence between peaks and write current transitions, a single error from a "missed" peak would propagate until the next "missed" peak.

The output of the peak detector is used as an input to the PLL, and the output clock produced by the PLL is constantly being adjusted so that the average peak position is centered with respect to the edges of the bit interval.

If one examines the waveform produced by the linear superposition of two Lorentzian pulses (of opposite sign) separated by αPW_{50} seconds, one finds that this waveform will contain two peaks separated by βPW_{50} seconds, where $\beta > \alpha$. The parameters α and β are related by the formula

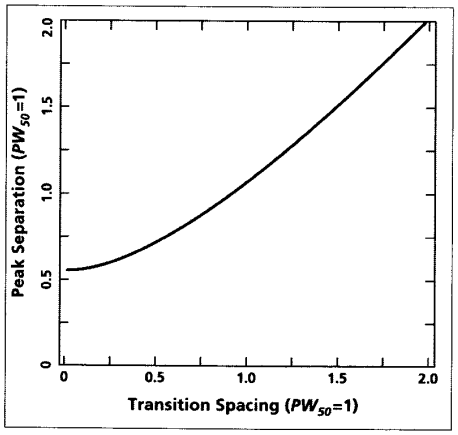
$$\beta = \sqrt{\frac{\alpha^2 - 1 + 2\sqrt{1+\alpha^2+\alpha^4}}{3}}$$

which for small α becomes

$$\beta \approx \frac{1+\alpha^2}{\sqrt{3}} .$$

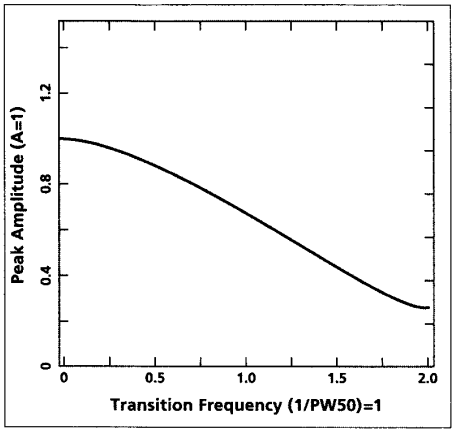
For α much greater than 1, β is approximately equal to α, but as α approaches zero, β approaches a fixed, limiting distance given by the value $\frac{1}{\sqrt{3}}$. Thus, the peaks will be centered in their bit interval only at low densities. Figure 7 shows a plot of

■ **Figure 7.** *Peak separation as a function of transition spacing*



Once a peak is detected by the peak detector, it is thought to be due to a transition in the input waveform.

A second performance enhancement technique for peak detection channels is the use of a modulation code.



■ **Figure 8.** *Roll-off curve for Lorentzian transition response*

the normalized peak separation β as a function of normalized transition spacing α.

Of course, another effect of the interference of pulses is the decrease of the peak amplitude. The amplitude of peaks in the superposition of an infinite sequence of Lorentzian pulses (of alternating polarity) as a function of the transition separation $T = \alpha PW_{50}$ seconds (sometimes called the "roll-off curve") also has a simple closed-form solution:

$$\gamma(\alpha) = \frac{(\pi/2\alpha)}{\sinh(\pi/2\alpha)},$$

where $\sinh(x) = (e^{x}-e^{-x})/2$ is the hyperbolic sine function. (For fans of complex analysis, we remark that one approach to deriving this formula provides a pleasant exercise in contour integration.) Figure 8 shows the rolloff curve as a function of normalized transition frequency. The amplitude $\gamma(\alpha)$ also plays a role in determining the design density as limited by the amplitude qualification of peaks in the detector.

To improve the performance of the system at moderate information densities, two techniques are used, often in combination. These techniques, "equalization" and "modulation coding," are now briefly described.

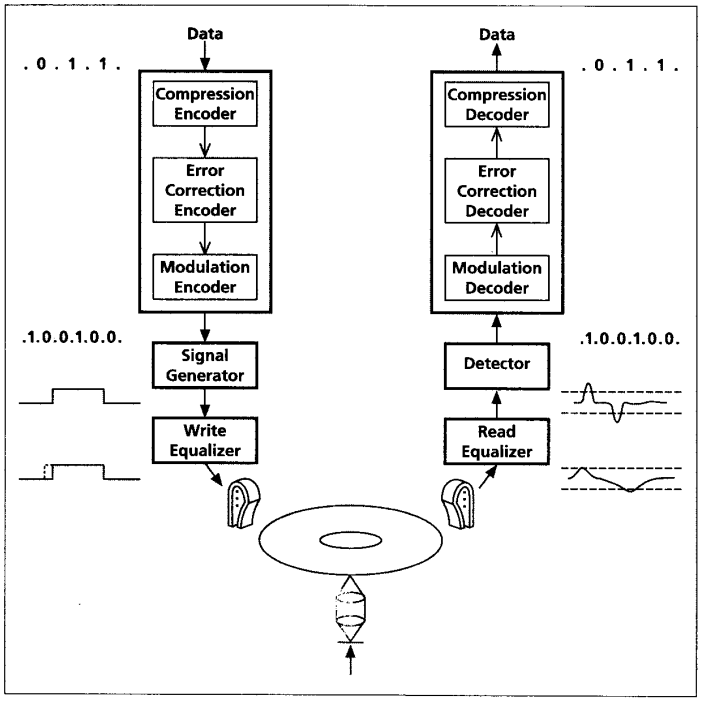
Equalization refers to the use of a linear filter at the output which changes the overall impulse response of the system before the peak detector. This equalizer is often called a pulse slimmer [3], its purpose being to create a channel whose isolated transition response is a thinner pulse than that produced by the unequalized channel. Many different equalization techniques are utilized, but they all have the effect of increasing the noise at the equalizer output since pulse slimming is achieved by boosting the high frequencies where there is little signal and much noise. We will not devote much space in this paper to the discussion of equalization. However, we will mention a new class of target transfer functions derived from partial-response signaling, a method familiar to many communications engineers.

A second performance enhancement technique for peak detection channels is the use of a modulation code, or, more specifically, a special class of these codes called run length-limited (RLL) or (d, k) codes. Here d and k are nonnegative integers with k strictly larger than d. A (d, k)-encoded sequence must satisfy the constraint that symbols "1" must be separated by at least d and at most k symbols "0." (The choice of the letters d and k to describe these codes is unfortunate because of their different meaning in the discussion of error-correction codes, but several attempts to change the notation have met with little success. Fortunately, in this paper we will make only brief remarks about error-correction codes, so no such confusion should arise).

We will describe (d, k) codes in terms of the restrictions that these codes produce on the write current. Whereas, for the uncoded case, the minimum time interval between transitions in the write current can be as small as T_b and the maximum time interval between adjacent transitions could be infinite, when a (d, k) code is utilized, the minimum time interval between transitions is $(d+1)T_b$ and the maximum time interval between transitions is $(k+1)T_b$. Since the magnetic medium is moving past the head at some velocity, v, the minimum spacing between transitions on the medium is $v(d+1)T_b$. The value $v(d+1)T_b$ is referred to as the linear transition density, and if this distance is measured in inches, the linear density is measured in flux changes per inch (fci). Typical high-end magnetic disk systems today operate at between 20,000 and 30,000 fci, and magnetic tape systems operate at linear densities as high as 62,000 fci.

The necessity for the parameter k to be finite follows from the fact that the PLL requires feedback from the peak detector in order to position the bit interval boundaries. If there is a large time interval between transitions, there will be a corresponding large time interval during which the PLL sees no corrective signal. Such a situation would result in an undesirable drift in the clock signal produced by the PLL. One might think that a data

■ **Figure 9.** *Block diagram of a recording system with peak detection*



scrambler could be used to make the likelihood of such an event very small. However, disk designers are interested in worst case behavior: once a file written on a disk can no longer be read reliably, there is no second chance to retransmit the file.

A block diagram of a typical recording system using a peak detector, linear equalization, a (d, k) modulation code, and an error-correcting code is shown in Fig. 9.

The user data (possibly compressed) is first passed through an encoder for an error-correction code (ECC) such as for a Reed-Solomon code. ([4].) The output of the ECC encoder is then encoded again using a (d, k) modulation encoder. Finally, the output of the (d, k) encoder is NRZI modulated by the signal generator, forming a write current of two levels, where a transition in the write current corresponds to a "1" in the (d, k)-encoded stream. On the read side, the output voltage waveform from the read head is equalized and passed through a peak detector (the ever-present phase-lock loop is not shown), where bit cells containing detected peaks are converted to 1s and bit cells without peaks are converted to 0s. The corresponding binary stream from the peak detector is next passed through the (d, k) decoder, then through the ECC decoder (and decompressed, if appropriate). Notice that the NRZI precoding convention provides a very elegant way to translate the constraints imposed upon the readback signal by the peak detection system into constraints on the sequences to be generated by the modulation encoder.

Most communication engineers have some familiarity with data compression and ECC schemes, but not as much with modulation coding techniques, particularly those used in data storage. The remainder of this paper will be devoted to a discussion of such modulation codes, beginning with (d, k) coding.

(d, k) Codes

In this section we describe some of the (d, k) codes that have been used in commercial storage products. Further details may be found in a variety of references [5–8]. We begin with a very simple example of a (d, k) code where $d = 0$ and $k = 2$. This code, sometimes called Group Code Recording (GCR) [9], has found application in magnetic tape drives. The code description is shown in Table 1.

User bits are encoded, four bits at a time, into five-bit code words. It is easily verified that any concatenation of these five-bit code words satisfies the (0,2) constraint.

The rate of the GCR code is 4/5 and one might wonder what is the maximum code rate possible for any (0,2) code. The answer to four decimal places is .8792. For (d,k) codes in general, the maximum code rate (sometimes called the capacity of the code) is given by the base-2 logarithm of the largest real root of one of the following equations, depending on whether k is finite or not:

$$x^{k+2} - x^{k+1} - x^{k+1-d} - 1 = 0, \quad k < \infty,$$

or

$$x^{d+1} - x^d - 1 = 0, \quad k = \infty$$

A listing of the maximum code rates for some (d, k) codes is given in Table 2.

User Bits	Channel Bits
0000	11001
0001	11011
0010	10010
0011	10011
0100	11101
0101	10101
0110	10110
0111	10111
1000	11010
1001	01001
1010	01010
1011	01011
1100	11110
1101	01101
1110	01110
1111	01111

■ **Table 1.** *Encoding/decoding table for (0,2) GCR code*

The purpose for choosing the parameter d to be strictly greater than zero is to increase the information density along a track while keeping the time interval between adjacent transitions greater than some fixed constant. Assume that T_{min} is the smallest time interval that can be allowed between neighboring transitions in the two-level write current. If d is equal to zero, then we must choose T_b such that $T_b \geq T_{min}$. For this choice of T_b, since one coded binary digit cannot contain more than one information bit, the maximum information rate that can be supported by the two-level input waveform would be $1/T_b \leq 1/T_{min}$ bits/second. Now assume that d is chosen as an integer strictly greater than zero. Since the minimum time interval between transitions is now $(d + 1)T_b$, T_b can be chosen to equal $T_{min}/(d + 1)$ and the coded binary digits then occur at a rate of $1/T_b = (d + 1)/T_{min}$ binary digits/second. For a fixed value of T_{min}, this corresponds to a coded symbol rate that is a factor of $(d + 1)$ times the transition rate. Unfortunately, this increase is not in the information rate but in the coded binary symbol rate, and the number of coded binary digits required to represent one information bit generally increases as d does. Let R be the ratio of the number of information bits to coded binary digits for a given (d,k) code. Then the information rate for this system using the (d, k) code is $R/T_b = R(d+1)/T_{min}$ bits/second. The product $R(d + 1)$, called the density ratio of the code [10], represents the increase (if $R(d + 1) > 1$) or decrease (if $R(d + 1) < 1$) in information rate using a (d,k) code as compared to an uncoded system. We are particularly interested in systems where the density ratio is strictly greater than 1, for then we are storing information at a higher

■ **Table 2.** *Maximum code rates for selected (d,k) constraints*

	d=0	d=1	d=2
k=1	.6942		
k=2	.8792	.4057	
k=3	.9468	.5515	.2878
k=4	.9752	.6175	.4057
k=5	.9881	.6509	.4650
k=6	.9942	.6690	.4979
k=7	.9971	.6793	.5174

The (1,3) code has many names: Miller code, Delay Modulation, and Modified Frequency Modulation (MFM).

d	k	m	n	R(d+1)
0	2	4	5	.8
1	3	1	2	1.0
1	7	2	3	1.33
2	7	1	2	1.5

■ **Table 3.** *Parameters of commonly used (d,k) codes*

rate than the highest possible transition rate allowed in the two-level input waveform. The most common (d,k) codes used in past and present products are listed in Table 3.

We have already discussed the (0,2) (GCR) code. The (1,3) code has many names, including Miller code, Delay Modulation, and Modified Frequency Modulation (MFM) code [9]. It is a rate 1/2 code where one information binary digit is encoded into two coded binary digits. It is a systematic code in that the information sequence $i_1, i_2, \ldots, i_k \ldots$ gets transformed into the coded sequence $i_1, z_1, i_2, z_2, \ldots i_k, z_k, \ldots$. There is a simple rule for inserting the extra digits $\{z_k\}$: Choose z_k to be 0 unless it is to be inserted between two information 0s, in which case choose z_k to be a 1.

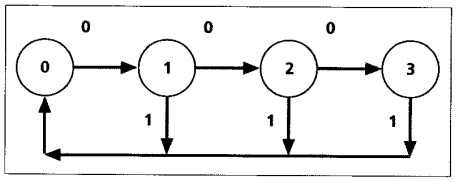
We will now demonstrate how to derive the coding rule for the (1,3) code. The derivation illustrates some of the basic ideas involved in the design of constrained recording codes. However, more powerful techniques are required in general, as will be discussed in more detail. We begin with a finite state transition diagram (FSTD) which produces all binary sequences satisfying the (1,3) constraint. This FSTD is shown in Fig. 10. Constrained code sequences are generated by taking walks on the graph, following the arrows and reading off the code symbols that label the edges traversed.

The capacity for a (1,3) code is $C \approx .5515$, and we desire a code rate close to C, such as a rate $R = 1/2$. Thus, we seek a graph that can represent a rate 1/2 encoder finite-state-machine, namely, one with two edges emanating from each state (one for encoding a 0, the other for encoding a 1) and with binary code words of length 2 on each edge. In order to obtain a graph with edges labeled by code sequences of length 2, we form the "second power of the FSTD of Fig. 10," that is, the FSTD obtained by taking steps of size 2 in Fig. 10. This graph is shown in Fig. 11.

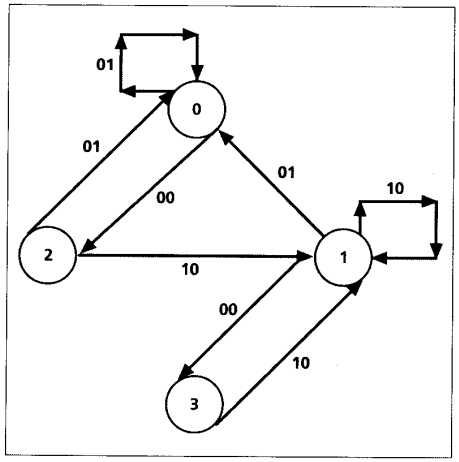
Note that state 3 in this graph is deficient in that it has only one edge emanating from it. However, we are very fortunate in that the only way to enter state 3 is from state 1, and state 1 has 3 edges emanating from it. Eliminating the edge that goes from state 1 to state 3 eliminates state 3 from consideration altogether, resulting in the FSTD shown in Fig. 12.

We now note that states 1 and 2 are "equivalent," meaning that both states produce the same set of code sequences. Thus, these states can be combined into one state as shown in Fig. 13 [8].

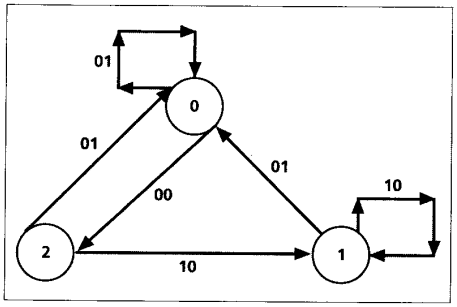
Finally, we obtain the state diagram for a rate $R = 1/2$ encoder for a (1,3) code by labeling the edges in the form a/bc, where a is the information bit that is the input to the encoder and bc is the pair of (1,3) constrained binary digits produced by the encoder. This is shown in Fig. 14. This encoder is precisely the same encoder as used in the Miller (1,3) code.



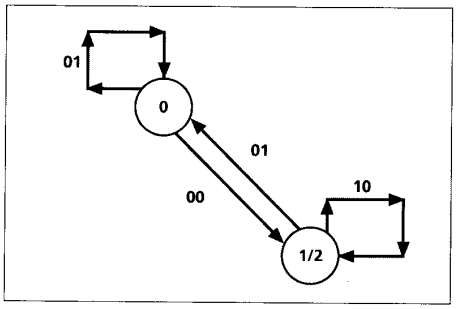
■ **Figure 10.** *Finite-state transition diagram for (1,3) sequences*



■ **Figure 11.** *Two-step transition diagram for (1,3) constrained binary sequences*



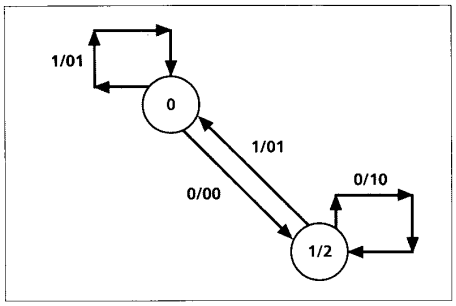
■ **Figure 12.** *Modified two-step transition diagram for (1,3) constrained binary sequences*



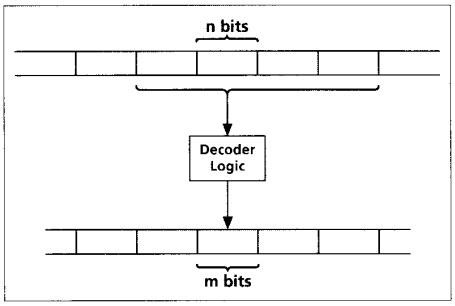
■ **Figure 13.** *Two-state transition diagram for rate 1/2 (1,3) constrained sequences*

The rate 2/3 (1,7) code is arguably the most popular (d,k) code in use today. Several variations of this code exist [11–13]. A simple and elegant description, due to Jacoby, begins with the encoding table in Table 4.

If the encoder is based solely upon a table lookup, however, we note that the user data sequences 00.00,



■ **Figure 14.** *Two-state encoder for rate 1/2 (1,3) code*



■ **Figure 15.** *Sliding block decoder*

00.01, 10.00, and 10.01 produce channel sequences (namely, 101.101, 101.100, 001.101, and 001.100, respectively) which violate the (1,7) constraint. The "substitution table" in Table 5 is used to correct for these violations.

When the encoder is ready to encode a pair of user data bits, it "looks ahead" to the next pair of user data bits to see if using Table 4 for both pairs would result in a violation of the (1,7) constraint. If no violation would occur, the encoder uses Table 4 to encode the first pair. If a violation would occur, the encoder encodes these two pairs using Table 5.

Decoding can be accomplished in a state-dependent manner using a "sliding-block decoder," a generalization of the code word "table look-up" decoder used for the (0,2) and the (1,3). A schematic of a sliding-block decoder is shown in Fig. 15.

As illustrated for a rate m/n code, the decoding of a code word of length n depends on the contents of a decoder window that contains the code word in question, as well as a fixed number of past and future code words ("lookback" and "look-ahead"). In the case of the (1,7) code, the decoder decodes the current three-bit code word by looking ahead at the next two upcoming code words. In this way, a single incorrectly detected code symbol can propagate into a burst of at most six user bits (in fact, the burst length does not exceed five user bits).

Another modulation code used in several disk drive products is a rate 1/2 (2,7) code. One encoding and decoding table for such a code, invented by P. Franaszek, is given in Table 6.

It is easily seen that the code rate is 1/2 (every code word contains exactly twice as many binary digits as the information sequence it represents) and that any concatenation of the variable length code words satisfies the (2,7) constraint (each 1 in every code word is followed by at least two 0s, and no code word begins with more than four 0s or ends in more than three 0s). It can also be verified that every information sequence can be decomposed uniquely into a sequence of variable length strings in the left-hand column of the table. In addition, the variable length code words constitute a prefix-free code (no code word is a prefix of another code word), so that unique decoding of code sequences can be accomplished. One can also describe a sliding-block decoder for this code in such a way that decoding errors due to a single code symbol in error cannot affect more than four user bits.

Prior to a few years ago, no general theory existed for the design of modulation codes (such as (d,k) codes) with minimum code word length, finite-state encoders, and sliding-block decoders, at rates arbitrarily close to capacity (or equal to capacity, when the capacity is a rational number). Now, however, there is systematic technique for code construction [8, 14]. The method, called the sliding-block code algorithm, allows for the design of a practical, efficient (d,k) code for any choice of the parameters d and k at any rational rate up to

User Data	Channel Bits
00	101
01	100
10	001
11	010

■ **Table 4.** *Basic Encoding Table for (1,7) code*

User Data	Channel Bits
00.00	101.000
00.01	100.000
10.00	001.000
10.01	010.000

■ **Table 5.** *Substitution Table for violations in (1,7) basic encoder*

Information Digits	Code Words
10	0100
11	1000
000	000100
010	100100
011	001000
0010	00100100
0011	00001000

■ **Table 6.** *Encoding/decoding table for rate 1/2 (2,7) code*

the capacity C, specifically for any integers m and n satisfying $m/n \leq C$, the algorithm yields a finite-state encoder that accepts m binary inputs and generates n binary outputs, and a state-independent decoder requiring only finite look-ahead and look-back, thereby limiting error propagation.

The sliding-block algorithm was first used to find a five-state encoder for a(1,7) code. It converts two binary information digits into three coded binary digits [12], and has been shown to generate the same set of code sequences as the (1,7) code described earlier. More recently, another rate 2/3 encoder with only four states [13], the minimum possible for a rate 2/3, (1,7) code [15],

The sliding block algorithm was first used to find a five-state encoder for a(1,7) code.

The nature of sliding-block codes is to propagate errors at the decoder input into a finite burst of decoded data errors.

was invented using the sliding-block code algorithm.

The nature of sliding-block codes is to propagate errors at the decoder input into a finite burst of decoded data errors. Therefore, random-burst correcting ECC have been applied as a sort of add-on to handle these rare, short burst errors. Single burst error correction codes and some specially crafted codes similar to Fire codes were used in some products, for example. Now, this design philosophy is slowly changing as engineers come to realize that error correction codes must be an integral part of any storage system and can lead to both higher reliability and storage efficiency. Today, the most general class of codes used are Reed-Solomon codes, although most products using Reed-Solomon codes are still not very aggressive in their error correction (e.g., the use of two-byte error correcting codes). Recently, however, several general purpose chips have become available [16, 17] that can correct (on the fly) many byte errors (e.g., 10) in a code word at speeds in excess of what is required for today's products. The optimization of the recording system, incorporating advanced modulation, detection, recording codes, and ECC is a challenging problem, beyond the scope of this paper, but it seems quite reasonable to conjecture that the newly available ECC power will find its way into future storage devices.

Partial Response Systems

Recently, researchers from IBM laboratories reported the results of an experiment demonstrating that an areal density of 1 gigabit per square inch could be achieved for the storage and reliable retrieval of digital data on a hard disk system [18]. This many-fold increase in density was achieved using a number of advanced techniques. One of these techniques was a different approach to combatting intersymbol interference, sometimes referred to as PRML, using partial-response (PR) signaling with maximum-likelihood (ML) sequence detection.

Instead of keeping the transitions far apart using (d,k) codes, PRML allows the transitions to be close together, and the read signal, with its resulting intersymbol interference, is equalized to a frequency response known as a class-4 partial-response channel [19]. The equalized signal is then detected by a maximum likelihood sequence estimator, i.e., a Viterbi detector [20]. In this section we will give a brief summary of the PR and ML components of this system as they apply to magnetic recording.

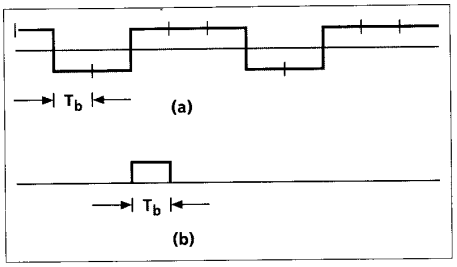
Partial Response Equalization

We now turn to a more detailed description of partial-response signaling. Consider the two-level write current, $I(t)$, shown in Fig. 16(a), where it should be noticed that the bit cell boundaries are now located such that the transitions occur at the edge of a bit cell. Shown in Fig. 16(b) is the elementary pulse $p(t)$.

Note that the write current $I(t)$ can be written in terms of the elementary pulse $p(t)$ as:

$$I(t) = \sum_{i=o}^{\infty} a_i p(t - iT_b)$$

where a_i takes on the values $+A$ or $-A$. Suppose that the channel is equalized so that the response



■ **Figure 16.** *(a) Write current, (b) Elementary pulse p(t)*

to the pulse $p(t)$ is a signal $h(t)$, referred to as the partial response signal. Then the readback voltage, $V(t)$, is of the form:

$$V(t) = \sum_{i=o}^{\infty} a_i h(t - iT_b).$$

Furthermore, assume that when $h(t)$ is sampled at bit interval boundaries every T_b seconds, the only nonzero samples are: $h_0 = h(0)$, $h_1 = h(T_b)$, ... , $h_L = h(LT_b)$. The nonzero samples can be conveniently represented by the "partial-response polynomial" $h(D) = h_0 + h_1 D + \ldots + h_L D^L$, where the factor D^i signifies a delay of i time units T_b. (In other words, $h(D)$ is the D-transform of the sampled pulse response.) The j-th sample of the readback voltage $V(jT_b)$ can be written as:

$$V(jT_b) = \sum_{i=j-L}^{j} a_i h(jT_b - iT_b).$$

or

$$V(jT_b) = \sum_{i=j-L}^{j} a_i h_{j-i}.$$

For magnetic recording systems with PW_{50}/T_b approximately equal to 2, comparatively little equalization is required to force the equalized channel to match a class-4 partial-response (PR4) channel where $h(D) = 1 - D^2$ [21]. At higher recording densities, one can choose a partial-response polynomial of the form $h(D) = (1 - D)(1 + D)^n$ where n is chosen as a positive integer greater than 1, to produce a response $h(t)$ that is a better match to the channel pulse response [22]. For PW_{50}/T_b approximately equal to 2.25, the proper choice of n is 2, leading to the so-called EPR4 (i.e., extended class-4 partial-response) channel with $h(D) = 1 + D - D^2 - D^3$ [22]. Eye diagrams for PR4 and EPR4 waveforms are shown in Figs. 17 and 18, respectively. These diagrams [19] represent the overlaying of the channel output signal seen in each time interval T_b, assuming a random binary input sequence. One can clearly see the nominal three-level (respectively, five-level) set of values at the sample times for the PR4 (respectively, EPR4) response. The eye diagrams provide some useful, qualitative indication of the robustness of the sample values at bit cell boundaries in the presence of additive noise and timing jitter.

Although PR equalization in communications systems and, as will be discussed, in magnetic recording systems is primarily used in conjunction with detection schemes that process samples of the channel output waveform, it is interesting to note that

it has also found application in recording systems employing peak detection. For example, raised-cosine and cosine-squared filters (corresponding to PR polynomials $h(D) = 1+D$ and $h(D) = (1+D)^2$) have long been used to achieve high frequency noise reduction and some degree of pulse slimming, although the realizations have typically not been minimum-bandwidth. In addition, more recently it has been demonstrated that equalizing the channel to an extended PR with polynomial of the form $h(D) = (1 - D)(1 + D)^n$ provides performance improvement in peak detection systems [23, 24].

In a later section, we will describe a newly proposed coding method designed for a peak detection system employing EPR4 equalization.

PRML

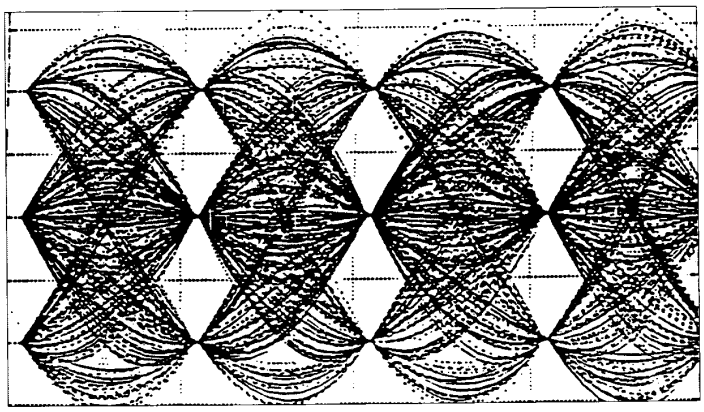
The applicability of partial-response signaling as a means of coping with intersymbol-interference in magnetic recording channels was suggested over 20 years ago [21], and the use of a Viterbi detector for maximum-likelihood sequence estimation in the storage context was proposed almost simultaneously with similar proposals for data communications [25, 26].

During the past 10 years, additional analysis, simulation and, finally, laboratory experimentation confirmed the potential value of the PRML system, [27–30]. In both simulation and experiments, the benefits in linear density that can be obtained over systems using RLL-coded peak detection have been found to be approximately 30 percent [31]. In addition, further research results indicated that the digital nature of the signal processing in PRML leads to advantages in electronic implementation (particularly VLSI) and extendibility, e.g., via coded-modulation [32, 33], digital adaptive equalization [34], and digital timing and gain control [30].

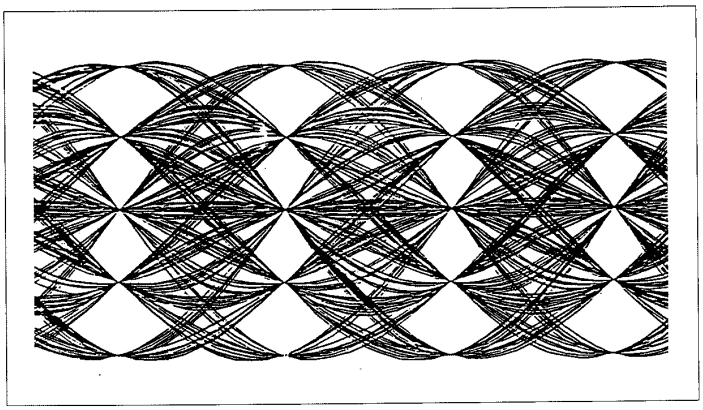
This activity has finally culminated in the incorporation of PRML into magnetic tape products, [35] and, very recently, magnetic disk products, [29].

We now briefly review the principles of Viterbi detection for combatting intersymbol interference, particularly as these concepts relate to coding for recording channels based upon partial-response. Recall that, in the NRZI $(1/(1 \oplus D))$ precoded channel, a code symbol 1 produces a positive or negative transition in the write current (respectively, a positive or negative pulse at the channel output) and a code symbol 0 produces no transition (respectively, pulse). This correspondence translated the restrictions on minimum and maximum transition spacing (respectively, run lengths of time intervals with no pulses at the input to the peak detector) into the easily represented (d,k) constraints.

Similarly, in the PRML setting, an Interleaved NRZI or INRZI $(1/(1 \oplus D^2))$ precoder converts the constraints on the samples at the input to the detector into simply described constraints on the code sequences applied to the input of the precoder, as we will describe. In the INRZI-precoded PR4 channel, a code symbol 0 at the input to the precoder will produce a sample 0 at the channel output, and a code symbol 1 will produce a sample value of either +1 or −1. The maximum-likelihood sequence detector based upon the Viterbi algorithm takes



■ **Figure 17.** *Eye diagram for PR4 channel*



■ **Figure 18.** *Eye diagram for EPR4 channel*

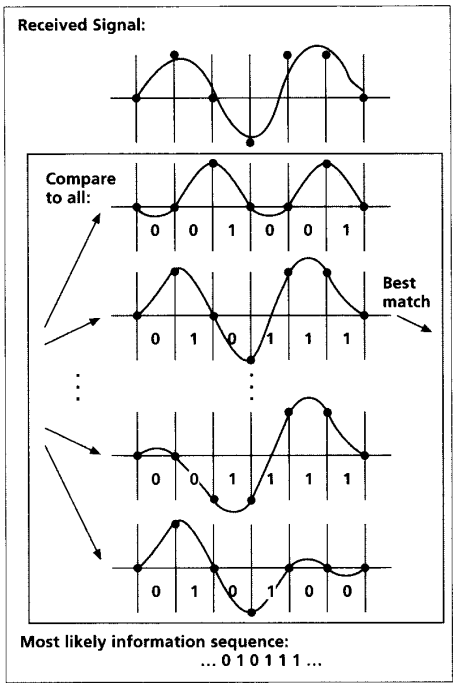
a received (noisy) sample sequence of length n,

$$\mathbf{y} = y_1, y_2, \cdots, y_n,$$

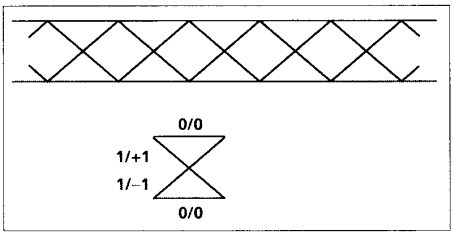
and determines in a recursive manner the channel output sequence (and the corresponding data sequence) that provides the best fit in a least-squares sense to the observed sample sequence, as shown schematically in Fig. 19 for the INRZI-precoded PR4 channel. (The authors wish to acknowledge that this visualization of ML detection is due to Gottfried Ungerboeck). A rough analogy to linear regression can be made, where one fits a line to a set of observations in such a way as to minimize the sum of squared errors.

The PR4 channel, with polynomial $h(D) = 1 - D^2$, can be considered as two time-interleaved "dicode" partial-response channels, each with polynomial $h(D) = 1 - D$. A common method of applying the Viterbi algorithm to maximum-likelihood sequence detection for the PR4 channel is to de-interleave the samples of the readback waveform to form two streams of samples, a stream made up of the samples at odd time indices and a stream made up of the samples at even times indices. Then two Viterbi detectors matched to the $1 - D$ channel can be used, one for each stream. In practice, one might even use just one detector in a pipelined fashion. This interleaving approach will be important when we come to discussing codes for this channel.

For an NRZI-precoded $1 - D$ channel, the

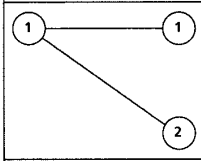
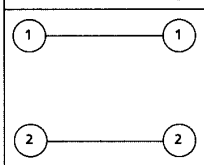
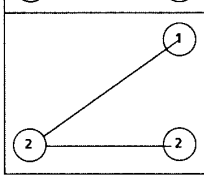


■ **Figure 19.** *Maximum-likelihood detection schematic*



■ **Figure 20.** *Trellis representation of 1 – D channel outputs*

■ **Figure 21.** *Difference-metric algorithm for 1 – D Viterbi detector*

Extension	Condition	Update
	$DM_k \leq 2y_{k+1} - 1$	$DM_{k+1} = 2y_{k+1} - 1$
	$2y_{k+1} - 1 < DM_k < 2y_{k+1} + 1$	$DM_{k+1} = DM_k$
	$2y_{k+1} + 1 \leq DM_k$	$DM_{k+1} = 2y_{k+1} + 1$

channel input/output sequences are conveniently represented by the trellis structure in Fig. 20.

Within each stage of the trellis, shown in the lower portion of Fig. 20, each edge is labeled with an input bit (before the slash) and a channel output symbol (after the slash). Input/output sequence pairs are generated by reading off the edge labels as one follows a path through the trellis from left to right. The two states at each time correspond to the parity of the number of input 1s so far, which determines the polarity of the channel output resulting from the next input 1. Alternatively, the states can be thought of as the polarity of the write current at the end of the last bit cell.

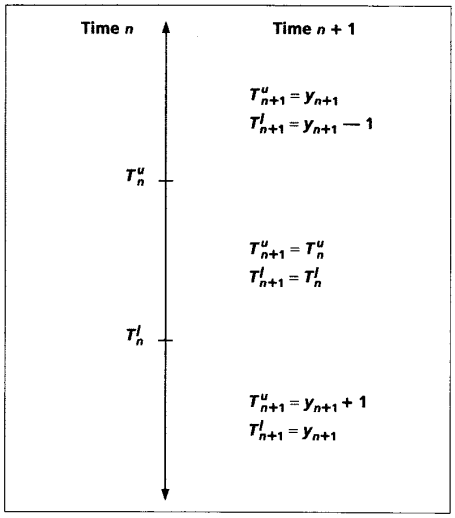
The Viterbi algorithm fits the time-indexed observations at the channel output with *two* allowable channel output sequences: one minimizing the sum of squared errors over all possible noiseless output sequences ending in the first state of the trellis at time n; the other minimizing the sum of squared errors over the noiseless output sequences ending in the second state of the trellis at time n.

The recursive algorithm for finding these two "survivor" sequences is based upon a concept from dynamic programming called the principle of optimality. What it says, and what can be readily checked, is that each of the two survivor sequences at any time $n + 1$ must correspond to an extension of one of the survivor sequences from time n. The application of this principle to the $1 - D$ channel leads to several elegant interpretations and implementations of the decoding algorithm. One description is based on the "difference metric," denoted DM_n, which is simply the difference of the accumulated squared errors for the two survivor sequences at a specified time n. The recursive "difference metric algorithm," first published in [36] (see also [30]), is described in Fig. 21. Note that only three of the four possible survivor extensions can occur.

This algorithm also can be interpreted as a "dynamic thresholding" scheme [37], as follows. Two threshold values, an upper threshold T_n^u and a lower threshold T_n^l, are initialized at time $n = 0$ to +1/2 and −1/2. If the sample value at time $n + 1$, y_{n+1} falls in the upper interval, $y_{n+1} > T_n^u$, the first extension is selected, and the new thresholds are set to $T_{n+1}^u = y_{n+1}$ and $T_{n+1}^l = y_{n+1} - 1$. If y_{n+1} falls in the middle interval, $T_n^l \leq y_{n+1} \leq T_n^u$, the second (parallel) extension is selected, and the thresholds remain unchanged, $T_{n+1}^u = T_n^u$ and $T_{n+1}^l = T_n^l$. Finally, if y_{n+1} falls in the lower interval, $y_{n+1} < T_n^l$, the third extension is selected, and the thresholds are set to $T_{n+1}^u = y_{n+1} + 1$ and $T_{n+1}^l = y_{n+1}$. See Fig. 22.

The first and third extension possibilities cause the survivor sequences to "merge" and, therefore, in both survivor sequences, all decisions prior to the merge agree with those in the maximum-likelihood estimate that is ultimately generated. Note that, in the absence of noise, the merges take place precisely when the input bit is a 1.

The parallel extension option, on the other hand, defers the decision about the bit following the last merge until a future merge settles the matter. Consequently, the detector must keep a record of the survivors (called the path memory or trellis history) at least back to the last merge. In practical applications, it is therefore very desirable to force frequent merges by limiting the separation of 1s at the precoder input, thereby reducing the likelihood of



■ **Figure 22.** *Dynamic threshold interpretation of Viterbi detection*

errors caused by truncation of the path memory to a manageable length. As mentioned earlier, in recording systems the preferred practice is to guarantee such a property of the recorded sequences by means of constrained coding, rather than to achieve it only probabilistically via scrambling. This constraint on the input will play a role in the next section when we discuss the recording code constraints for PRML channels.

In general, if the partial-response polynomial is of degree L, and if the input to the channel is a binary sequence, then the Viterbi detector will require $2L$ states. For example, the Viterbi detector for the EPR4 partial-response channel, where $h(D) = 1 + D - D^2 + D^3$, uses an eight-state trellis, shown in Fig. 23. Here the trellis states represent the write-current levels, denoted 0 and 1, at the ends of the last three-bit cells.

An important performance indicator for a trellis-based detector is the free squared-Euclidean distance which, roughly speaking, measures the separation between the sequences most likely to be confused when corrupted by channel noise. The free distance can be derived from the trellis, as follows. We consider the sum of the squared differences between the noiseless outputs for every pair of paths in the trellis that start in a common state and end in a common (but perhaps different) state. The minimum of this sum is the free (squared-Euclidean) distance. For the $1 - D$ trellis shown in Fig. 20, the free squared-Euclidean distance is equal to 2.

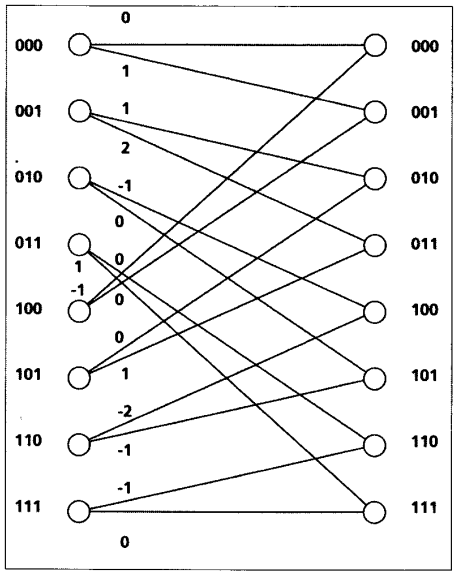
In a channel with additive white Gaussian noise, with zero mean and standard deviation σ, the probability of an error event, for moderate to high SNR, is then well-approximated by:

$$Pr(error\ event) \approx NQ\left(\frac{d_{free}}{2\sigma}\right),$$

where N is a constant determined by the trellis, and $Q(x)$ is the familiar complementary error function

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{x^2}{2}}\,dx.$$

We remark that sometimes in defining the free squared-Euclidean distance we do not insist that the ending state be the same for the two paths. Of course, the expression for the probability of error event must then be modified accordingly. With this definition, for example, the free squared-Euclidean distance for the $1 - D$ trellis would be equal to 1.



■ **Figure 23.** *Eight-state trellis for EPR4 channel*

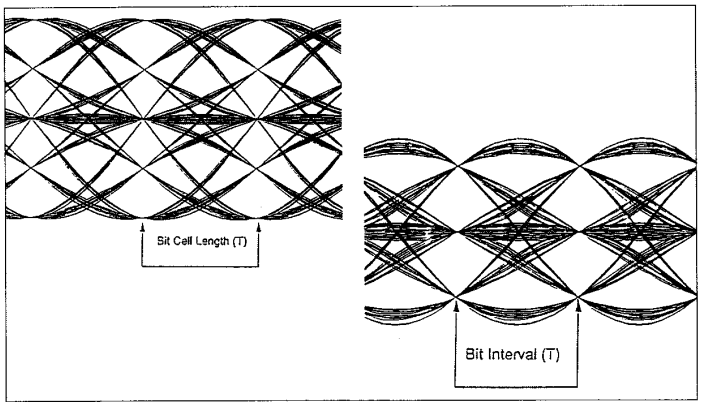
Codes for Partial Response Channels

In this section, we will describe several coding techniques developed for use in recording systems using partial response equalization. We will first describe a code designed for a new peak detection system incorporating EPR4 equalization. We then turn to a class of constrained codes, denoted $(0, G/I)$ codes, that have been used in the implementation of PRML channels in disk drives. These constraints incorporate the analogue of the k constraint in (d,k) codes, as well as a new constraint that affects the required length of the Viterbi detector path memory. Finally, we turn to the consideration of codes intended to improve the noise immunity of partial-response channels, and we indicate several exploratory directions proposed for trellis-coded modulation in storage channels.

Codes for Peak Detection for the EPR4 Channel

This section describes an exploratory technique for extending the use of peak detection to the EPR4 channel [38]. Recall that the EPR4 channel corresponds to the partial-response polynomial $h(D) = 1 + D - D^2 - D^3$. The eye pattern for this equalization (using the minimum bandwidth signal having these sampled values) was given in Fig. 18. It should be noted from this figure that the channel output waveforms exhibit a number of different types of peaks with varying amplitudes and that, at the normal sampling point for the EPR4 channel, the waveforms can assume five different values.

If one restricts the input waveform so that the cor-



■ **Figure 24.** *(a) Eye pattern for (1,7)-coded EPR4 Channel*
(b) Eye pattern for INRZI-precoded (1,7) EPR4 Channel

responding binary sequence (using NRZI modulation) satisfies the (1,7) constraint, the resulting eye pattern is as shown in Figure 24(a).

Now all but the large peaks have been eliminated by the code, and a peak detector could be used to identify transitions in the input write signal. At the normal sampling point for the EPR4 channel the waveform still has five levels. If the (1,7) coded sequence is first passed through a precoder with transfer function $1/(1 \oplus D^2)$, the resulting eye pattern is as shown in Figure 24(b). Now only the smallest peaks are retained in this waveform, and once again a peak detector can be used to recover the input write current. At the normal sampling point for the EPR4 waveform, this coded waveform has only three levels.

■ **Figure 25.** *Response of EPR4 channel to different input signals*

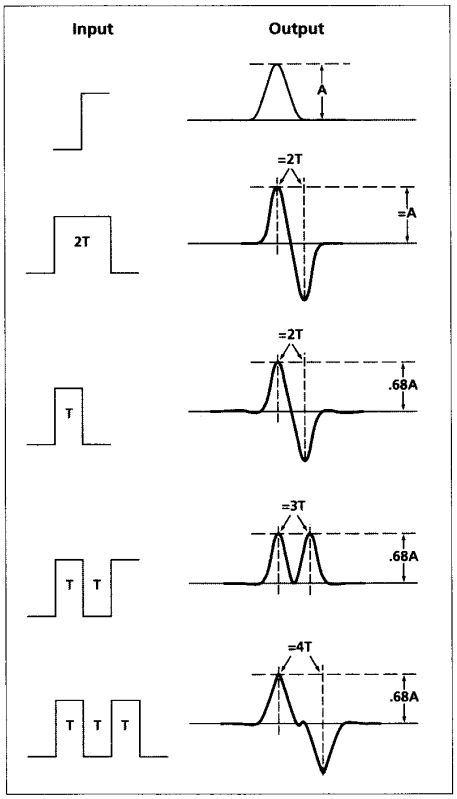


Figure 25 shows the response of the EPR4 channel to a number of different input waveforms. It is important to note that a pulse of duration $2T$ (here T is the sampling interval) results in a pair of peaks of heights $+A$ and $-A$ separated by approximately $2T$, while a pulse of duration T produces a pair of peaks of height approximately $+.68A$ and $-.68A$ and separated again by almost $2T$. If the number N of input transitions spaced T apart exceeds 2, the ouput waveform will contain two peaks of amplitude .68A spaced approximately NT apart, as illustrated for $N = 3, 4$ at the bottom of Fig. 25. In general, the peaks of amplitude A are out of phase by 1/2 a bit cell from the peaks of amplitude $.68A$. When the basic waveforms shown in Fig. 25 are concatenated with each other, the shapes of the output waveforms remain approximately as shown in Fig. 25, provided that the last transition of the preceding basic waveform is no closer than $3T$ from the first transition of the basic waveform that follows it. The basic waveforms producing large peaks need only be separated by two T or more.

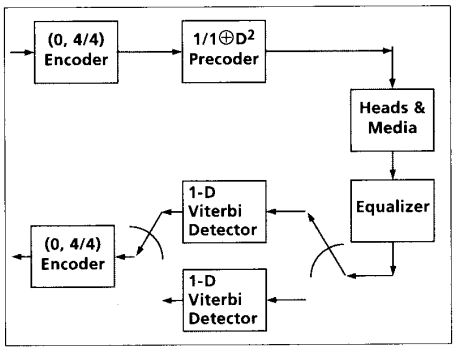
Using these observations as motivation, a code has been devised for use with this INRZI-precoded EPR4 channel (where the precoder has transfer function $1/(1 \oplus D^2)$) such that a peak detector that can reliably detect peaks and discriminate between large and small peaks can identify the binary sequence that corresponds to the write current. The code also produces either a large or a small peak in the readback waveform every eight bit cells for timing recovery. The maximum possible rate for this code is $C \approx .8485$, which is approximately 25 percent higher than for a (1,7) code.

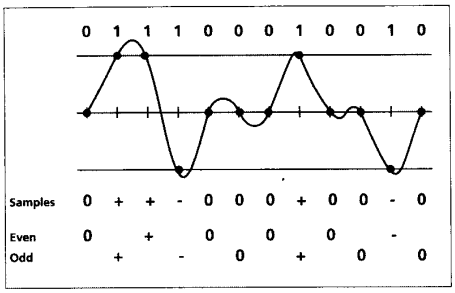
(0,G/I) Codes for the PRML Channel

The gigabit per square inch experiment and the first United States disk drive product employing PR4 equalization utilized code sequences satisfying a new type of constraint called a $(0, G/I)$ constraint. This constraint imposes run length limitations that aid timing and gain recovery and simplify the design of the Viterbi detector for the channel. A block diagram of the PRML system using this code is shown in Fig. 26. The channel response is equalized to that of the PR4 system. As described previously, the PR4 Viterbi detector is decomposed into a pair of interleaved detectors matched to the $1 - D$ channel.

The constraints on the PR4 channel outputs

■ **Figure 26.** *Block diagram of system using a (0,4/4) code*





■ **Figure 27.** *Waveform and samples for (0,4/4) sequence*

are twofold. First, for accurate timing and gain recovery, it is desirable to limit the number of consecutive zero samples in the noiseless PR4 channel output sequences generated by the code sequences. Second, in order to minimize performance degradation resulting from the truncation of the path memory in the interleaved $1 - D$ Viterbi detectors, the maximum run length of zero samples in each of the two interleaves comprising a channel output sequence should be limited.

As mentioned previously, the INRZI-precoded PR4 channel generates a sample 0 at the channel output for a code symbol 0 at the input, and a sample value of either +1 or −1 at the output for a code symbol 1 at the input. Therefore, the first constraint translates into a global constraint, denoted by the symbol G, on the maximum run length of 0 symbols in any code string. This constraint is essentially the same as the k constraint in the (d, k) codes. The second constraint corresponds to an interleaved constraint, denoted by I, on the maximum run lengths of 0 symbols in each of the interleaves of a code sequence. The "0" in the notation $(0,G/I)$ can be treated as the analogue of the global d constraint. Here, it serves to emphasize that there is no restriction on the minimum separation of nonzero samples at the channel output; that is, the code sequences are not forbidden to have adjacent 1s. Figure 27 shows an example of a code sequence satisfying a $(0, G/I) = (0, 4/4)$ constraint, similar to the constraints used in the commercial PRML channel, along with the corresponding noiseless channel output waveform, the global sample sequence, and the interleaved sample sequences. Note that adjacent nonzero samples can have the same polarity, that is, they do not correspond directly to peaks in the output waveform.

As with the (d, k) constraints, there are simple finite-state transition diagrams that describe the $(0, G/I)$ constraints from which one can compute capacities and construct sliding-block codes. The capacities of several $(0, G/I)$ constraints of practical interest, as well as the parameters of rate 8/9 codes satisfying these constraints [39] are given in Table 7.

J. Eggenberger first discovered the optimal block list of length 9 for the (0, 4/4) and (0, 3/6) constraints, i.e., the largest collection of nine-bit code words that satisfy the prescribed constraints when freely concatenated. The (0, 4/4) optimal block list contains 279 words, listed in decimal representation in Table 8.

(0,G/I)	Capacity	Code Rate	Encoder States	Decoder Window
(0,4/4)	.961	8/9	1	1
(0,4/3)	.939	8/9	3	1
(0,3/6)	.944	8/9	1	1
(0,3/5)	.942	8/9	2	1
(0,3/4)	.934	8/9	3	2
(0,3/3)	.915	8/9	4	2

■ **Table 7.** *Parameters of some rate 8/9 (0, G/I) codes*

■ **Table 8.** *Maximal list of length-9 words for the (0,4/4) constraint*

73	116	183	225	268	310	361	402	438	479
75	117	185	227	269	311	363	403	439	481
76	118	186	228	270	313	364	406	441	483
77	119	187	229	271	314	365	407	442	484
78	121	188	230	281	315	366	409	443	485
79	122	189	231	282	316	367	410	444	486
89	123	190	233	283	317	369	411	445	487
90	124	191	235	284	318	370	412	446	489
91	125	195	236	285	319	371	413	447	491
92	126	198	237	286	329	372	414	451	492
93	127	199	238	287	331	373	415	454	493
94	146	201	239	289	332	374	417	455	494
95	147	203	241	291	333	375	419	457	495
97	150	204	242	292	334	377	420	459	497
99	151	205	243	293	335	378	421	460	498
100	153	206	244	294	345	379	422	461	499
101	154	207	245	295	346	380	423	462	500
102	155	210	246	297	347	381	425	463	501
103	156	211	247	299	348	382	427	466	502
105	157	214	249	300	349	383	428	467	503
107	158	215	250	301	350	390	429	470	505
108	159	217	251	302	351	391	430	471	506
109	177	218	252	303	353	393	431	473	507
110	178	219	253	305	355	395	433	474	508
111	179	220	254	306	356	396	434	475	509
113	180	221	255	307	357	397	435	476	510
114	181	222	265	308	358	398	436	477	511
115	182	223	267	309	335	399	437	478	

For the (0, 3/6) constraint, Eggenberger found two optimal block lists containing 272 code words; the code words in one code being the time-reverse of the code words in the other. For these constraints, specific rate 8/9 codes were obtained by selecting 256 code words from these optimal lists. (For more details about these codes and their logic implementations, see [8, 39, 40].)

Codes for Improving Noise Immunity for the PR4 Channel

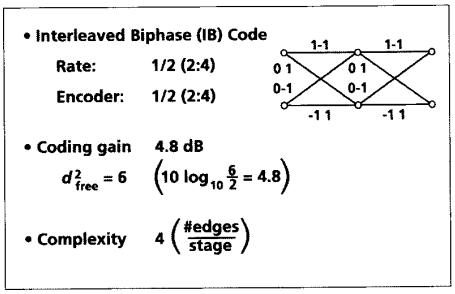
The main reason for choosing the particular class of partial-response channels discussed previously is that the required equalizers do not boost the noise power excessively. The equalizer does change the spectral characteristics of the noise somewhat, but often this factor is ignored in the design and analysis of the Viterbi detectors (when the slight loss in accuracy and performance is deemed acceptable).

As mentioned previously, the performance of the Viterbi detector at high signal-to-noise ratios (SNR) is known to be governed by the free squared-Euclidean distance of the code, and the free squared-Euclidean distance for the $1 - D$ channel is equal to 2. Codes for improving the noise immunity for the PR4 channel involve eliminating some of the sequences generated by paths through the trellis shown in Fig. 20, with the objective of increasing the free squared-Euclidean distance by a coding gain factor g, where $g > 1$. In any coded system of this kind, one is trading information rate (represented by the code rate R) and complexity (of the encoder/decoder logic as well as the modified detector) for coding gain. The parameter Rg is called the asymptotic coding gain (ACG) for the coded system, and to a first approximation represents how much more (or less) noise can be tolerated by the coded system as compared with an uncoded system that yields the same error performance (at low error rates).

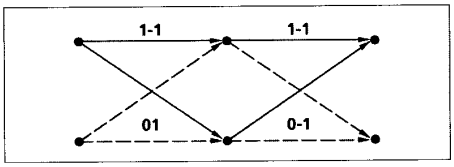
The interleaved-FM code for the PR4 channel provides a simple, but representative, example of this tradeoff and the potential value of similar coded-modulation schemes. The channel configuration is the same as in the previous section, with the PR4 channel decomposed into two interleaved $1 - D$ channels, each preceded by a $1/(1 \oplus D)$ precoder. The interleaved-FM code, as the name suggests, is defined by applying separately to the even and odd interleaves of the data string a simple rate 1/2 code, often referred to as the FM (frequency modulation) code. The FM code is a block code with the encoding rule:

$$0 \rightarrow 01$$
$$1 \rightarrow 11$$

The interleaved-FM code therefore satisfies a $(0, G/I) = (0, 2/1)$ constraint. We remark that the sequences produced at the precoder output comprise the widely known biphase code, a simple rate 1/2 block code with code words 01 and 10. The precoded PR4 channel with the interleaved-FM code is therefore equivalent to the (unprecoded) PR4 channel with interleaved-biphase coding, and we will use the names interchangeably. The trellis describing the output sequences as a function of the input data in the FM-coded $1 - D$ channel is shown in Fig. 28. If we use a pair of detectors in an interleaved fashion for detection of the interleaved-FM coded PR4 channel, as was done for



■ **Figure 28.** *Trellis structure for FM-coded 1 – D channel*



■ **Figure 29.** *Minimum distance event for FM-coded 1 – D channel*

the uncoded PR4 channel, it suffices to examine this trellis. The minimum squared-Euclidean distance generated by an error-event is 6, so the distance gain factor is $g = 3$. An example of such a minimum distance event is depicted in Fig. 29. The ACG is therefore given by:

$$ACG = 10\log_{10} Rg = 10\log_{10} \frac{1}{2} \cdot 3 = 1.8\, dB.$$

To assess the added hardware complexity associated with the trellis-coded scheme, one can use rough meaningful measures for the encoder/decoder logic and Viterbi detector. Specifically, for the encoder/decoder, one might look at the number of states and the data and code word length in the finite-state encoder, along with the length of the sliding-block decoder window. With regard to this measure, the FM code adds only minor complexity relative to the uncoded channel: the encoder has only one state and can be implemented by simply inserting a 1 following each data bit; the decoder window is two code bits long, and decoding is implemented by simply dropping the second code bit of each detected pair.

From the trellis structure, we can get a qualitative indication of the hardware complexity by looking at the number of states (each corresponding to an Add-Compare-Select processor), the number of edges per trellis stage (the number of possible survivor extensions that need to be examined), and the number of samples detected per trellis stage. In the FM case, there are only two states, four edges per stage, and two samples per stage, again representing a minimal increment in complexity relative to the uncoded PRML detector. There is even a simple difference metric form for the Viterbi algorithm [41].

As an approach to increasing linear density, the advantage of such a coded-PR4 system, although possibly substantial, may not be apparent without careful analysis of the channel signal-to-noise ratio as a function of transition density and the required channel equalization [42]. In disk recording systems, however, there is a second coordinate, corresponding to the radial direction, along

which density can be increased. Overall areal density can be optimized by choosing the appropriate balance of linear and radial (i.e., track) density. In this setting, the potential benefit of coded systems can be appreciated by considering the following interesting, although simplistic, argument.

Suppose one begins with a benchmark PRML channel using a rate 8/9 (0,4/4) code on a nominal track width of W, with linear density L. Let's assume that head and servo technology permit the physical reduction of track width by a factor of three. One could then divide the original track into three subtracks, each of width $W/3$. Theory and experiment [43] indicate that the amplitude of the readback signal and the noise power due to the medium would both be reduced by approximately this factor of three, corresponding to a 4.8 dB signal-to-noise ratio penalty. If, on each subtrack, the original channel scheme were applied (leading to a tripling of the areal density) the performance would be unacceptably poor due to the SNR loss. However, if we apply the rate 1/2 interleaved-FM code on each subtrack, the gain factor g exactly offsets the SNR loss, implying that the probability of error on each subtrack will be virtually unchanged from the nominal value. The catch, of course, is that the code rate on each subtrack is now 1/2, reducing the linear density per subtrack to

$$\frac{(1/2)}{(8/9)}L = \frac{9}{16}L.$$

The track density has been tripled, however, leading to an overall areal density that is a factor of 27/16 (or approximately 1.7) times the original, implying a 70 percent increase (Fig. 30).

Although this back-of-the-envelope calculation ignores several important technology issues, such as intertrack interference, narrow-track width head design, and position-servo accuracy, it at least suggests that the development of practical coded systems might provide a route to significant increases in areal density in disk drives.

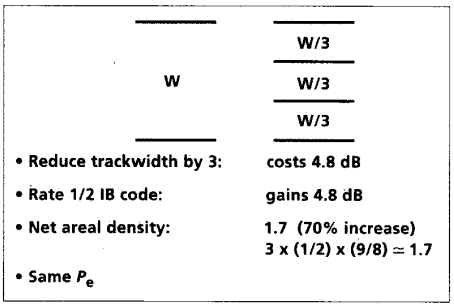
The example clearly illustrates that the objective is to design codes with high rate and large coding gain, in order to minimize the track width reduction and to provide the greatest noise immunity and areal density increase in the scenario just sketched. Soon we will give examples of codes with rate 4/5 and gain factor $g = 2$. A calculation similar that above shows that cutting the track width by two and applying a code with these parameters on each subtrack provides an estimated areal density increase of almost 80 percent.

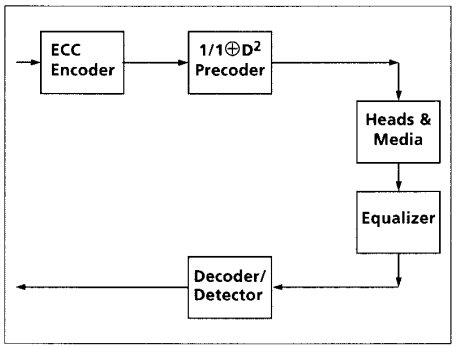
Several methods have been found for the design of coded systems with ACG greater than 1, for a range of code rates. The simplest such scheme is to use an ordinary binary ECC code and a precoder [32] as shown in Fig. 31. (See also [44].) Assume that the ECC code has rate R and minimum Hamming distance equal to d^H_{min}, so that when the code is used as an error correction code over a random binary channel it is capable of correcting $(d^H_{min} - 1)/2$ or fewer errors. Recall that an input 0 to the precoded channel produces an output 0, and an input 1 produces an output +1 or −1. It follows that the minimal squared-Euclidean distance will satisfy the inequality

$$d^2_{min} \geq d^H_{min}.$$

In fact, it can be shown that for the coded PR4 system incorporating this code d^2_{min} must be even. Therefore, the coded channel has an ACG bounded below by $Rd^H_{min}/2$ if d^H_{min} is even or $R(d^H_{min} + 1)/2$ if d^H_{min} is odd.

■ **Figure 30.** *Track cutting for increased areal density*





■ **Figure 31.** *Block diagram of coded system for noise immunity*

To get the full benefit of any coding gain, the decoder for this system should be matched both to the code and also to the precoded $1 - D^2$ partial-response channel. Just as in the uncoded case, one could bit-wise interleave code words and treat the channel as two $1 - D$ channels. If the ECC is a convolutional code, the combination of the code and the $1 - D$ channel can be decoded by a single trellis. If one interleaves a convolutional code with a 2^s state encoder, the resulting combined Viterbi detector/decoder operates on a trellis with at most 2^{s+1} states.

The coding gain bound suggests that for a given rate, one would like to use an ECC with the largest minimum Hamming distance. Optimal convolutional codes have been found for a wide range of rates and trellis complexity by computer search, and tables of these are now available in many textbooks [45]. Using these codes, the lower bound on minimum distance of the trellis-coded PR4 channel has been found to be tight in virtually all cases.

One still has to limit the global runs of consecutive 0s entering the precoder in order to guarantee a usable signal for timing recovery, as well as the runs of 0s on the interleaves to reduce path memory requirements. One approach to accomplishing this is to use a coset of the ECC code, i.e., an additive translation of the code sequences, obtained by the componentwise addition (modulo 2) of a fixed binary sequence to each code sequence [32]. For a rate k/n code, one could do this by complementing the code symbols in a specific set of positions

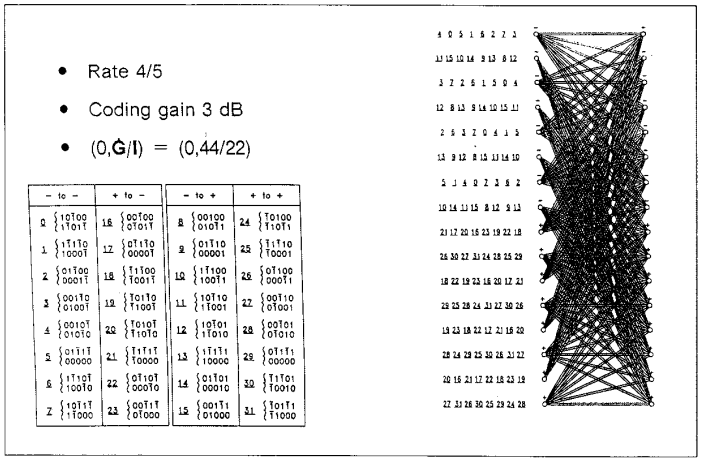


Figure 32. *Trellis structure for rate 4/5 code with ACG 2.0 dB*

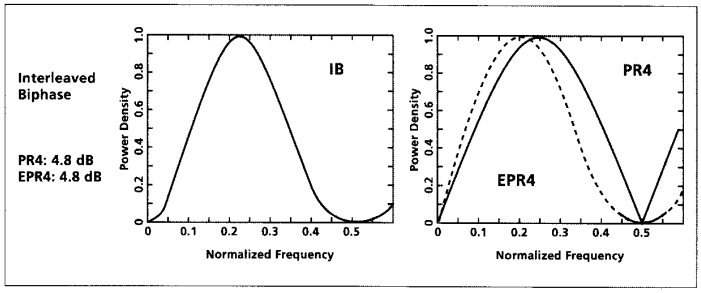


Figure 33. *Power spectrum of Interleaved-FM and PR4/EPR4 frequency response*

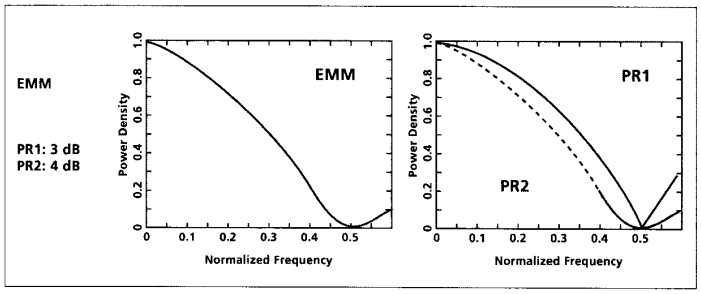


Figure 34. *Power spectrum of EMM and PR1/PR2 frequency response*

in each code word of length n. (Another approach might be to first encode the data using a $(0, k)$ code, and then to apply a systematic error correcting code with the desired Hamming distance.)

An example of the precoded ECC scheme, given in [32], makes use of a coset of a rate 4/5 convolutional code, with minimum Hamming distance 3, applied to each of the interleaved $1-D$ channels comprising the PR4 channel. The resulting gain factor is $g = 2$, providing ACG equal to 2 dB. The decoding trellis for each interleave, derived from the eight-state trellis of the underlying convolutional code, has 16 states, as shown in Fig. 32 [32, 33]. This code satisfies the run length constraints $(0, G/I) = (0, 44/22)$.

The reader may have observed that the interleaved-FM code, which has a minimum Hamming distance 1 and is not particularly attractive as an ECC code, achieves a coding gain that far exceeds the lower bound just derived. This code is an example of another class of codes, called matched-spectral-null (MSN) codes, which recently have been shown to provide an efficient method of achieving moderate coding gain at high rates. An MSN code is designed in such a way that the average power spectrum of the write-current waveforms generated by the code has the value zero (i.e., a spectral null) at frequencies where the partial-response channel frequency response is zero.

Aside from the intuitive reasonableness of this design criterion (why waste signal power by transmitting at frequencies where the channel response is zero?) it has been shown mathematically that matching of the code and channel spectral null frequencies provides significant coding gain for a large class of partial response channels, including those relevant to magnetic and optical recording channels [33].

For example, Fig. 33 shows the power spectrum of the interleaved-biphase code or, equivalently, the precoded interleaved-FM code, alongside the frequency response of the PR4 channel, as well as the frequency response of the EPR4 channel, where it also provides ACG of 1.8 dB.

Another example of an MSN code, intended for optical recording channels, is the rate 2/3, even-mark-modulation (EMM) code [46]. It has a spectral null at the Nyquist frequency (one-half the recorded symbol frequency) and provides coding gain factors $g = 2$ and $g = 2.5$ for the $1 + D$ (class 1 or PR1) and $(1 + D)^2$ (class 2 or PR2) partial-response channels. The corresponding power spectral density and frequency response curves are shown in Fig. 34.

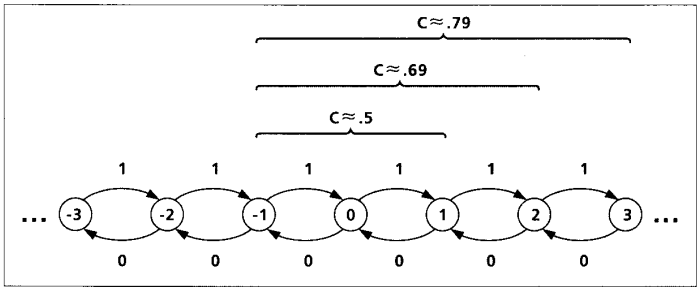
Returning to the $1 - D$ channel, it has been shown [33] that the minimum squared-Euclidean distance at the output of the coded channel is bounded below by $2K$, where K is the order of the code's spectral null at zero frequency (meaning that the first $2K - 1$ derivatives of the code power spectrum are zero at zero frequency).

Practical codes with spectral nulls of a given order at specified frequencies can be designed using the sliding-block code construction techniques alluded to earlier. We remark that the spectral null constraints automatically provide the necessary run length constraints characteristic of the $(0, G/I)$ codes described in the previous section, as illustrated by the interleaved-biphase code. The initial FSTD used in the code design procedure is chosen to describe a family of spectral null sequences with capacity large enough to satisfy the target code rate. For example, Fig. 35 shows such a so-called canonical diagram from which one can extract initial FSTDs describing sequences with a spectral null at zero frequency [33]. (The labels should be interpreted as write-current levels.)

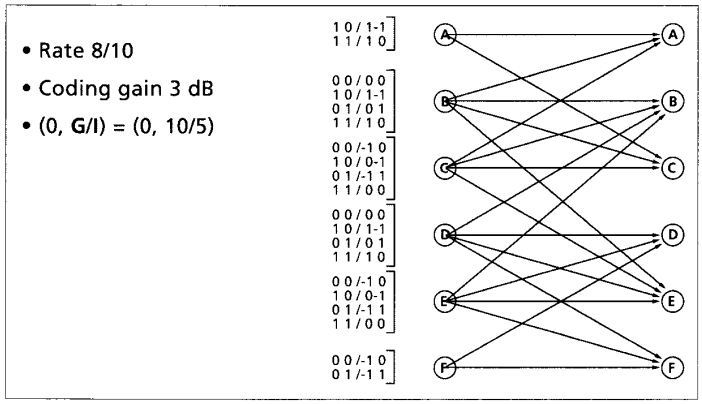
A rate 8/10 spectral null code with a four-state encoder, satisfying $(0, G/I) = (0, 10/5)$ constraints when interleaved, was designed for the $1 - D$ channel. Unfortunately, as might be expected, the complexity of the encoder finite state machine generally increases as the code rate does, so the trellis structure reflecting the combination of the modulation code and channel can be quite complicated for high code rates. Indeed, in the case of the rate 8/10 MSN code, the trellis would have eight states, and 256 branches emanating from each state.

As shown in [33], however, there is a natural, reduced-complexity Viterbi detector for MSN codes that asymptotically achieves maximum-likelihood performance as a function of the signal-to-noise ratio. The detector structure is based on the much simpler trellis derived from the initial FSTD used in the sliding-block code construction. The MSN code sequences belong to the supercode of spectral null sequences that are represented by the reduced-complexity trellis, and the MSN code can be designed to ensure that, for any code sequence, no sequence generated by the trellis is closer to the code sequence (in Euclidean distance) than the minimum Euclidean distance of the code. Thus, the decoder can apply the Viterbi algorithm to the reduced-complexity trellis to find the spectral null sequence in the supercode that best fits the noisy channel output sequence. In the unlikely event that the sequence produced by the detector is in error or is not in the range of the MSN code, the sliding-block decoder limits the propagation of errors in the decoded data sequence. For the rate 8/10 code, the corresponding reduced-complexity trellis is shown in Fig. 36. Issues related to the VLSI implementation of an exploratory rate 8/10 MSN code for PR4 are discussed in more detail in [47].

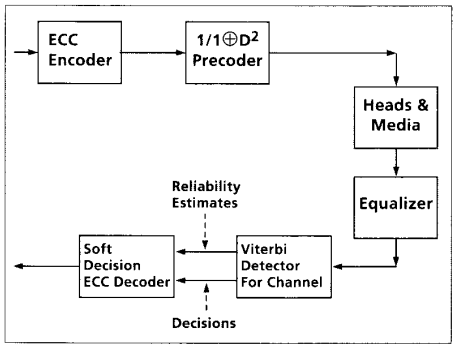
A recently proposed alternative method of achieving coding gain is shown in Fig. 37. Again, a code with good Hamming distance is used for the ECC code. Now, however, we require that the code be chosen from the class of codes that allow for efficient decoding via a soft decision decoding algorithm. (All convolutional codes fit into this class.) A two-step detector is now used. The first step uses an enhanced Viterbi detector matched to the channel itself, but modified so that the detector outputs reliability estimates on the binary data in the detected data stream. The second step invokes a soft decision decoder for the ECC code that uses the output of the first Viterbi detector as its input. This concatenated detector approach, using an extended BCH code with code word length 64 and minimum Hamming distance six, has been shown to achieve an asymptotic coding gain of close to 3 dB [48].



■ **Figure 35.** *Canonical diagram for spectral null at zero frequency*



■ **Figure 36.** *Trellis structure for rate 8/10 MSN code*



■ **Figure 37.** *A pair of decoders that use soft decisions*

Other Techniques

Several other promising techniques have been proposed to increase the density of recording. One of these, familiar to communication engineers, is decision feedback equalization [49, 50]. Another is a limited tree search algorithm [51]. There also is an interesting hybrid technique, using portions of the peak detection system and the PRML approach to detect (1,7)-coded information [52]. These topics have not been treated here, but the reader is encouraged to consult the references for details.

Summary

In this paper, we have discussed many of the types of modulation codes designed for use in storage devices using magnetic recording. The codes are intended to minimize the negative effects of intersymbol interference. Most commercial disk drives today use a simple type of detector called a peak detector, and a corresponding class of codes called run length-limited (d,k) codes have found wide application. Recently, another recording channel technology, based on sampling detection-partial-response (or PRML), has been introduced in commercial disk drives. This technology hinges on the use of controlled intersymbol interference, and it requires a new class of codes, called (0, G/I) codes.

The paper concluded with several examples illustrating that the introduction of partial response equalization, sampling detection, and digital signal processing has set the stage for the invention and application of advanced modulation and coding techniques in future storage products.

Acknowledgment

The authors are grateful to our many colleagues, particularly those at the Center for Magnetic Recording Research (UCSD) and IBM, who work in the field of signal processing and coding, and whose technical

results form the basis for this article. We also gratefully acknowledge support from National Science Foundation Grant 9105639, the Center for Magnetic Recording Research (UCSD), and the IBM Corporation.

References

[1] J. C. Mallinson, "A Unified View of High Density Digital Recording Theory", *IEEE Transact. Magn.*, vol. MAG-II, no. 5, pp. 1166-69, Sept. 1975.

[2] P. Siegel, "Applications of a Peak Detection Channel Model," *IEEE Trans. Magn.*, vol. MAG-18, no. 6, pp. 1250-52, Nov. 1984.

[3] L. Barbosa, "Minimum Noise Pulse Slimmer," *IEEE Trans. Magn.*, vol. MAG-17, no. 6, pp. 3340-42, Nov. 1981.

[4] E. Berlekamp, "The Technology of Error-Correcting Codes," *Proc. IEEE*, vol. 68, no. 5, pp. 564-93, May 1980.

[5] P. Siegel. "Recording Codes for Digital Magnetic Storage," *IEEE Trans. Magn.*, vol. MAG-21, no. 5, pp. 1344-49, Sept. 1985.

[6] K. A. Schouhamer Immink, "Runlength-Limited Sequences," *Proc. IEEE*, vol. 78, no. 11, pp. 1745-59, Nov. 1990.

[7] K. A. Schouhamer Immink, *Coding Techniques for Digital Recorders*, (Prentice-Hall, 1991).

[8] B. Marcus, P. Siegel, and J. Wolf, "A Tutorial on Finite-State Modulation Codes for Data Storage," *IEEE J. Select. Areas. Comm.* - Signal Processing and Coding for Recording Channels, to appear.

[9] F. Jorgensen, *The Complete Handbook of Magnetic Recording*, (TAB Books, 1980).

[10] A. Patel, "Zero Modulation Encoding in Magnetic Recording," *IBM J. Res. Dev.*, vol. 19, no. 4, pp. 366-78, July 1975.

[11] G. Jacoby, "A New Look-Ahead Code for Increased Data Density," *IEEE Trans. Magn.*, vol. MAG-13, no. 5, pp. 1202-04, Sept. 1977.

[12] R. Adler, M. Hassner, and J. Moussouris, "Method and Apparatus for Generating a Noiseless sliding-block Code for a (1,7) Channel with Rate 2/3," U.S. Patent 4,413,251,1982.

[13] A. Weathers and J. Wolf, "A New Rate 2/3 Sliding-Block Code for the (1,7) Runlength Constraint with the Minimal Number of Encoder States," *IEEE Trans. Inf. Theory*, vol. 37, no. 3, pt. II, pp. 908-13, May 1991.

[14] R. Adler, D. Coppersmith, and M. Hassner, "Algorithms for Sliding-block Codes," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 1, pp. 5-22, Jan. 1983.

[15] B. Marcus and R. Roth, "Bounds on the Number of States in Encoder Graphs for Input-Constrained Channels," *IEEE Trans. Inf. Theory*, vol. 37, no. 3, pt. II, pp. 742-58, May 1991.

[16] AHA Product Specification 4010, "High Speed Reed Solomon Encoder/Decoder T=I-10, Moscow, ID, 1988.

[17] P. Tong, "A 40-MHz Encoder-Decoder Chip Generated by a Reed-Solomon Compiler," Proc IEEE 1990 Custom Integrated Circuits Conf. Boston, MA, pp. 13.5.1-5.4, May 1990.

[18] T. Howell, *et al.*, "Error Rate Performance of Experimental Gigabit per Square Inch Recording Components," *IEEE Trans. Magn.*, vol. 26, no. 5, pp. 2298-2302, Sept. 1990.

[19] P. Kabal and S. Pasupathy, "Partial-Response Signaling," *IEEE Trans. Comm.*, vol. 23, no. 9, pp. 921-34, Sept. 1975.

[20] G. D. Forney, Jr., "The Viterbi Algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268-78, March 1973.

[21] H. Kobayashi and D. T. Tang, "Application of Partial-Response Channel Coding to Magnetic Recording Systems," *IBM J. Res. Dev.*, vol. 14, pp. 368-75, July 1970.

[22] H. Thapar and A. Patel, "A Class of Partial-Response Systems for Increasing Storage Density in Magnetic Recording," *IEEE Trans. Magn.*, vol. MAG-23, no. 5, pp. 3666-68, Sept. 1987.

[23] N. P. Sands, H. K. Thapar, and J. M. Cioffi, "A Comparison of Run-Length-Limited Codes for Equalized Peak Detection," Proceedings of the 1990 Asilomar Conf., pp. 682-86.

[24] H. K. Thapar, *et al.*, "Spectral Shaping for Peak Detection Equalization," *IEEE Trans. Magn.*, vol. 26, no. 5, pp. 2309-11, Sept. 1990.

[25] H. Kobayashi, "Application of Probabilistic Decoding to Digital Magnetic Recording Systems," *IBM J. Res. Develop.*, vol. 15, pp. 64-74, Jan. 1971.

[26] G. D. Forney, Jr., "Maximum Likelihood Sequence Detection in the Presence of Intersymbol Interference, *IEEE Trans. Inf. Theory*, vol. IT-18, no. 3, pp. 363-78, May 1972.

[27] R. W. Wood, *et al.*, "An Experimental Eight-Inch Disc Drive with One Hundred Megabytes per Surface," *IEEE Trans. Magn.*, vol. MAG-20, no. 5, pp. 698-702, Sept. 1984.

[28] R. W. Wood and D. A. Petersen, "Viterbi Detection of Class IV Partial-Response on a Magnetic Recording Channel," *IEEE Trans. Comm.*, vol. COM-34, pp. 454-61, May 1986.

[29] J. D. Coker, *et al.*, "Implementation of PRMI, in a Rigid Disk Drive," Digests of the Magnetic Recording Conf. 1991, paper D3, June 1991.

[30] R. Cidecyian, *et al.*, "A PRML System for Digital Magnetic Recording," *IEEE J. Select. Areas Comm.* - Signal Processing and Coding for Recording Channels, to appear.

[31] H. Thapar and T. Howell, "On the Performance of Partial-Response Maximum Likelihood and Peak Detection Methods in Digital Magnetic Recording," Digests of the 1991 Magnetic Recording Conf., Paper DI, Pittsburgh, PA, June 12-15, 1991.

[32] J. K. Wolf and G. Ungerboeck, "Trellis Coding for Partial-Response Channels," *IEEE Trans. Comm.*, vol. COM-34, no. 8, pp. 765-73, Aug. 1986.

[33] R. Karabed and P. Siegel, "Matched Spectral-Null Codes for Partial-Response Channels," *IEEE Trans. Inf. Theory*, vol. 37, no. 3, pt. II, pp. 818-55, May 1991.

[34] J. Cioffi, *et al.*, "Adaptive Equalization in Magnetic-Disk Storage Channels," *IEEE Comm. Mag.*, pp. 14-29, Feb. 1990.

[35] C. Coleman, *et al.*, "High Data Rate Recording in a Single Channel," Proc. of the Fifth Internationcl Conf. On Video and Data Recording, Hampshire, England, pp. 151-57, April 1984.

[36] M. M. Ferguson, "Optimal Reception for Binary Partial-Response Channels," *Bell Syst. Tech. J.*, vol. 51, no. 2, pp. 493-505, Feb. 1972.

[37] R. Wood, "Denser Magnetic Memory," *IEEE Spectrum*, vol. 27, no. 5, pp. 32-39, May 1990.

[38] A. Armstrong and J. Wolf, "Performance Evaulation of a New Coding Scheme for the Peak Detecting Magnetic Recording Channel," *IEEE Trans. Magn.*, vol. 27, no. 11, November 1991.

[39] B. Marcus and P. Siegel, "Constrained Codes for PRML," IBM Research Report RJ 4371, July 1984.

[40] J. Eggenberger and A. M. Patel, "Method and Apparatus for Implementing Optimum PRML Codes," U.S. Patent 4,707,681, issued November 17, 1987.

[41] T. Howell, R. Karabed, and P. Siegel, "Difference Metric Decoder for Interleaved Biphase Trellis Code," *IBM Technical Disclosure Bulletin*, vol. 31, no. 7, pp. 476-81, Dec. 1988.

[42] K. A. Schouhamer Immink, "Coding Techniques for the Noisy Magnetic Recording Channel," *IEEE Trans. Commun.*, vol. 37, pp. 413-19, May 1989.

[43] S. Lambert, et al., "Reduction of Edge Noise in Thin Film Metal Media: Using Discrete Tracks," *IEEE Trans. Magn.*, vol. 25, no. 5, pp. 3381-83, Sept. 1989.

[44] A. R. Calderbank, C. Heegard, and T. A Lee, "Binary Convolutional Codes with Application to Magnetic Recording," *IEEE Trans. Inf. Theory* , vol. IT-32, no. 6, pp. 797-815, Nov. 1986.

[45] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. (Prentice-Hall, 1983).

[46] R. Karabed and P. Siegel, "Even-Mark-Modulation for Optical Recording," Proc. of the 1989 Int'l Conference Communications (ICC '89), vol. 3, pp. 1628-32, Boston, MA, June 1989.

[47] C. B. Shung, *et al.*, "A 30MHz Trellis Codec Chip for Partial-Response Channels," Digests of the 1991 IEEE International Solid-State Circuits Conf. (ISSCC '91), San Francisco, CA, Feb. 13-15, 1991, pp. 132-33. To appear in *IEEE J. Solid-State Circuits, Special Issue on Analog and Signal Processing Circuits*, vol. 26, no. 12, pp. 1981-1987, December 1991.

[48] K. Knudson, J. K. Wolf, and L. Milstein, "Producing Soft- Decision Information at the Output of a Class IV Partial Response Viterbi Detector," Proc. of the IEEE Int'l Conf. on Comm. '91, pp. 26.5.1-5.5, Denver, CO, June 1991.

[49] J. Bergmans, "Density Improvements in Digital Magnetic Recording by Decision Feedback Equalization," *IEEE Trans. Magn.*, vol. MAG-22, no. 3, pp. 157-62, May 1986.

[50] K. Fisher, J. Cioffi, C. Melas, "An Adaptive DFE for Storage Channels Suffering from Nonlinear ISI," Proc. 1989 IEEE Int'l. Conf. Comm., Boston, MA, June 1989.

[51] J. J. Moon and L. R. Carley, "Performance Comparison of Detection Methods in Magnetic Recording," *IEEE Trans. Magn.*, vol. MAG-26, no. 6, pp. 3155-72, Nov. 1991.

[52] A. M. Patel, "A New Digital Signal Processing Channel for Data Storage Products," Digests of the Magnetic Recording Conf. 1991, Paper E6, June 1991.

Biography

PAUL H. SIEGEL was born in Berkeley, California in 1953. He received the B.S. degree in mathematics in 1975 and the Ph.D. degree in mathematics in 1979, both from the Massachusetts Institute of Technology. He held a Chaim Weizman fellowship during a year of postdoctoral study at the Courant Institute, New York University. He joined the research staff at IBM in 1980. He is currently manager of the Signal Processing and Coding project at the IBM Almaden Research Center in San Jose, California. His primary research interest is the mathematical foundations of signal processing and coding, especially as applicable to digital data storage channels. He holds several patents in the area of coding and detection for digital recording systems. He has taught courses in information and coding at the University of California, Santa Cruz and at Santa Clara University, and was a Visiting Associate Professor at the University of California, San Diego while at the Center for Magnetic Recording Research during the 1989-90 academic year. Dr. Siegel was elected to Phi Beta Kappa in 1974. He is a Senior Member of the IEEE, and is currently a member of the Board of Governors of the IEEE Information Theory Society. He was a co-Guest Editor of the May 1991 Special Issue on Coding for Storage Devices of the IEEE Transactions on Information Theory.

JACK KEIL WOLF received the B.S.E.E. degree from the University of Pennsylvania, Philadelphia, in 1956, and the M.S.E., M.A., and Ph.D. degrees from Princeton University, Princeton, NJ, in 1957, 1958, and 1960, respectively. He was a member of the Electrical Engineering Department at New York University from 1963 to 1965, and the Polytechnic Institute of Brooklyn from 1965 to 1973. He was Chairman of the Department of Electrical and Computer Engineering at the University of Massachusetts from 1973 to 1975, and was Professor there from 1973 to 1984. Since 1985, he has been a Chaired Professor of Electrical Engineering and Computer Engineering and a member of the Center for Magnetic Recording Research at the University of California, San Diego. He also holds a part-time appointment at Qualcomm, Inc., San Diego, CA. His current interest is in signal processing for storage systems. He served on the Board of Governors of the IEEE Information Theory Group in 1974. He was International Chairman of Committee C of URSI from 1980 to 1983. Dr. Wolf is a recipient of the 1990 E. H. Armstrong Achievement Award of the IEEE Communications Society and was corecipient of the 1975 IEEE Information Theory Group Paper Award for the paper "Noiseless Coding for Correlated Information Sources" (coauthored with D. Slepian). From 1971 to 1972, he was an NSF Senior Postdoctoral Fellow, and from 1979 to 1980 he held a Guggenheim Fellowship.