A Survey of High-Rate Constrained Codes

Kees A. Schouhamer Immink

Institute for Experimental Mathematics,
Ellernstrasse 29-31, D-45326 Essen, Germany.
Phone +31 402 851 854 Fax +31 402 860852.
E-mail: immink@exp-math.uni-essen.de.

Abstract– The construction of high-rate codes is far from obvious, as table look-up for encoding and decoding is an engineering impracticality. The usual approach is to supplement the p source bits with $m = q - p$ bits. Under certain, usually simple, rules the source word is modified in such a way that the modified word plus supplement bits comply with the constraints. The information that certain modifications have been made is carried by the m supplement bits. The receiver, on reception of the word, will undo the modifications. In order to reduce complexity and error propagation, the number of bits affected by a modification should be as small as possible. We will survey some examples of code constructions.

1 Introduction

Codes for the noiseless recording channel, often called *recording codes*, on which we will concentrate, convert the input stream to a signal suitable for the specific physical recorder requirements. The special attributes that the recorded sequences should have to render it compatible with the physical characteristics of the available transmission channel are called *channel constraints.* For example in optical recording, information is recorded in the form of pits and the absence of pits, called lands. If 'ones' were be written as pits and 'zeros' as lands, then a long sequence of 'zeros' would mean that, as no pits are written, the track would be absent for some time. This may pose problems with the tracking during reading, and track loss could be the result. This serious flaw can be solved by forbidding such vexatious sequences to be generated. In order to do so, a recording code is used which transforms the source sequence into a sequence, where such long strings of 'zeros' cannot occur. This is an example of a *runlength* constraint, which is specified in the time domain. Constraints can also be formulated in the frequency domain. *Frequency domain constraints* are certain specifications of the *power density function* or *spectrum* of the generated sequence. Sequences with a spectral null at a given frequency are called *spectral null* sequences. Sequences with a spectral null located at the zero frequency are often referred to as *dc-free* or *dc-balanced* sequences.

In coding practice, the source sequence is partitioned into blocks of length p, and under the code rules such blocks are mapped onto words of q channel symbols. The rate of such an encoder is $R = p/q \leq C$. A code may be state dependent, in which case the codeword used to represent a given source block is a function of the channel or encoder state, or the code may be state independent. State independence implies that codewords can be freely concatenated without violating the sequence constraints. A set of such codewords is called *self-concatenable.* When the encoder is state dependent, it typically takes the form of a synchronous finite-state machine.

Concatenation of codewords can also be established by using *merging* bits between constrained words [1]. Each source word has a unique q'-bit channel representation. We require one look-up table for translating source words into constrained words of length q' plus some logic circuitry for determining the $q - q'$ merging bits. Decoding is extremely simple: discard the merging bits and translate the q'-bit word into the p-bit source word.

2 High-Rate Constrained Codes

The well-known ACH algorithm is in its niche when p and q are relatively small. If, on the other hand, these numbers are large, the ACH algorithm is less suited to a practical design. The largest dc-balanced codes or $(0,k)$-constrained code designed with the principal state method are usually of rate 8/10 or 8/9 [2] [3] [4]. For certain applications it is desirable that the code rate is much higher than 8/9. The construction of such high-rate codes is far from obvious, as table look-up for encoding and decoding is an engineering impracticality.

Immink [5] gave a constructive proof that (dk) codes with merging bits can be made for which $C - R < 1/(2q)$. As a result, (dk) codes with a rate only 0.1% less than Shannon's capacity can be constructed with codewords of length $q \approx 500$. The number of codewords grows exponentially with the codeword length, and the key obstacle to practically approaching capacity is the massive hardware required for the translation. The massiveness problem can be solved by using a technique called *enumeration* [6], which makes it possible to translate source words into codewords and vice versa by invoking an algorithmic procedure rather than performing the translation with a look-up table. Single channel bit errors could corrupt the entire data in the decoded word, and, of course, the longer the codeword the greater the number of data symbols affected. This difficulty can be solved by a special configuration of the error correcting code and the recording code [5]. The usual approach is to supplement the p source bits with $m = q - p$ bits. Under certain, usually simple, rules the source word is modified in such a way that the modified word plus supplement bits comply with the constraints. The information that certain modifications have been made is carried by the m supplement bits. The receiver, on reception of the word, will undo the modifications. In order to reduce complexity and error propagation, the number of bits affected by a modification should be as small as possible. Below we have given some examples to show how such constructions can be executed.

A traditional example a simple dc-free code is called the *polarity bit code* [7]. The p source symbols are supplemented by one bit called the *polarity bit.* The encoder has the option to transmit the $(p+1)$-bit word without modification or to invert all $(p+1)$ symbols. The choice of a specific translation is made in such a way that the running digital sum is as close to zero as possible. It can easily be shown that the running digital sum takes a finite number of values, so that the sequence generated is dc-balanced.

A surprisingly simple method for transforming an arbitrary word into a codeword having equal numbers of 'ones' and 'zeros', a *zero-disparity* word, was published by Knuth [8] and Henry [9]. Let

$$d(\mathbf{w}) = \sum_{i=1}^{p} w_i \tag{1}$$

be the *disparity* of the binary source word $\mathbf{w} = (w_1, \ldots, w_p)$, $w_i \in \{-1, 1\}$. Let $d_k(\mathbf{w})$ be the running digital sum of the first k, $k \leq p$, bits of $\mathbf{w}$, or

$$d_k(\mathbf{w}) = \sum_{i=1}^{k} w_i, \tag{2}$$

and let $\mathbf{w}^{(k)}$ be the word $\mathbf{w}$ with its first k bits inverted. For example, if $\mathbf{w}$ = (-1, 1, 1, 1, -1, 1, -1, 1, 1, -1), we have $d(\mathbf{w}) = 2$ and $\mathbf{w}^{(4)}$ = (1, -1, -1, -1, -1, 1, -1, 1, 1, -1). If $\mathbf{w}$ is of even length p, and if we let $\sigma_k(\mathbf{w})$ stand for $d(\mathbf{w}^{(k)})$, then the quantity $\sigma_k(\mathbf{w})$ is

$$\begin{aligned} \sigma_k(\mathbf{w}) &= -\sum_{i=1}^{k} w_i + \sum_{i=k+1}^{p} w_i \\ &= -2\sum_{i=1}^{k} w_i + d(\mathbf{w}). \end{aligned} \tag{3}$$

It is immediate that $\sigma_0(\mathbf{w}) = d(\mathbf{w})$, (no symbols inverted) and $\sigma_p(\mathbf{w}) = -d(\mathbf{w})$ (all symbols inverted). We may therefore conclude that every word $\mathbf{w}$ can be associated with at least one k, so that $\sigma_k(\mathbf{w}) = 0$, or $\mathbf{w}^{(k)}$ is balanced. The value of k is encoded in a (preferably) zero-disparity word $\mathbf{u}$ of length m, m even. If m and p are both odd, we can use a similar construction. The maximum codeword length of $\mathbf{w}$ is governed by

$$\binom{m}{m/2}. \tag{4}$$

Some other modifications of the basic scheme are discussed in Knuth [8] and Alon [10].

3 $(0, k)$ codes

We will now focus on procedures for translating arbitrary date into words that have at most k 'zeros' between consecutive pairs of 'ones'.

3.0.1 Codes by bit manipulation

We will now focus on simple k-constrained block codes that translates $n-1$ user bits into codewords of n, $n \geq 9$, channel bits. The bit stream formed by concatenating the codewords has the virtue that at most $k = 1+$ entier $(n/3)$, $n \geq 9$, between logical 'ones' will occur. The construction is characterized by the fact that in order to build the codeword at most eight bits of the $(n-1)$-tuple have to be altered, which has a bearing on the worst-case error propagation.

Definitions

Let $k(\boldsymbol{x})$ be the maximum runlength of 'zeroes' in the word $\boldsymbol{x}$. Let $l(\boldsymbol{x})$ be the number of consecutive leading 'zeroes' of the word x, that is, the number of 'zeroes' preceding the first 'one'. And let $r(\boldsymbol{x})$ be the number of consecutive trailing 'zeroes' of the word $\boldsymbol{x}$, that is, the number of 'zeroes' succeeding the last 'one'.

Preliminaries

Let the $(n-1)$-tuple $\boldsymbol{z} = (z_1, \ldots, z_{n-1})$ be the binary input word and let $p = (n-1) \div 2 + 1$.

Define the intermediate n-tuple $\boldsymbol{x} = (x_1, \ldots, x_n)$ by $x_i = z_i$, $1 \leq i \leq p-1$, $x_{i+1} = z_i$, $p \leq i \leq n$, and set the pivot bit $x_p := 1$. The left and right parts of the vector $\boldsymbol{x}$, $\boldsymbol{xl}$ and $\boldsymbol{xr}$, are defined by $xl_i = z_i$, $xr_i = z_{i+p}$, $1 \leq i \leq n \div 2$. A more judicious allocation of the source bits is possible in a byte-oriented system (see later). Set the pivot bit $x_p := 1$.

MAIN ALGORITHM, k=1+entier(n/3)

Let r=entier$(k/2)$;$l = k - r$, and define the intermediate n-tuple $\boldsymbol{y} = (y_1, \ldots, y_n)$, where $y_i = x_i$, $1 \leq i \leq n$

If $l(\boldsymbol{y}) \leq l$ and $r(\boldsymbol{y}) \leq r$ and $k(\boldsymbol{y}) \leq k$ then transmit $\boldsymbol{y}$ as is.

If $(l(\boldsymbol{y}) \leq l$ and $k(\boldsymbol{xl}) \leq k)$ and $(r(\boldsymbol{y}) > r$ or $k(\boldsymbol{xr}) > k)$ then begin
$y_{p-1} := 1; y_p := 0; y_{p+1} := 0; y_{n-r} := 1; y_{n-r+1} := x_{p-1}; y_{n-r+2} := x_{p+1};$
transmit $\boldsymbol{y}$;end;

If $(r(\boldsymbol{y}) \leq l$ and $k(\boldsymbol{xr}) \leq k)$ and $(l(\boldsymbol{y}) > l$ or $k(\boldsymbol{xl}) > k)$ then begin
$y_{p-1} := 0; y_p := 0; y_{p+1} := 1; y_{l+1} := 1; y_{l-1} := x_{p+1}; y_l := x_{p-1};$
transmit $\boldsymbol{y}$;end;

If $(l(\boldsymbol{y}) > l$ or $k(\boldsymbol{xl}) > k)$ and $(r(\boldsymbol{y}) > r$ or $k(\boldsymbol{xr}) > k)$ then begin
$y_{p-1} := 1; y_p := 0; y_{p+1} := 1; y_{l+1} := 1; y_{n-r} := 1; y_l := x_{p-1}; y_{n-r+1} := x_{p+1}$; transmit $\boldsymbol{y}$;end;

During decoding, the pivot symbol y_p is observed. If $y_p = 1$ then decoding is straightforward. If, on the other hand, $y_p = 0$ the bits y_{p-1} and y_{p+1} are used to uniquely re-constitute the original $(n-1)$-tuple. Note that in total at most eight bits of the original $(n-1)$-tuple $\boldsymbol{z}$ are involved in the scheme, namely y_1, y_2, y_3, y_{p-1}, $y_{p+1}, y_{n-2}, y_{n-1}$, and y_n. In order to avoid error propagation in a byte-oriented system, these bits should be taken from one input byte. Then any error propagation resulting from an error in the pivot bit y_p occurred during transmission is confined to one decoded byte. In the next subsection, the algorithm is worked out for a rate 16/17 code.

3.1 Description of the rate 16/17, (0,6) code

The code translates two bytes of user data into 17 channel bits. The 17-bit codewords are characterized by the fact that they have at most six consecutive 'zeroes' and have at most three leading and at most three trailing 'zeroes'. Error propagation is limited as any single channel bit error made during retrieval will result in at most one decoded byte error. Let $\boldsymbol{z} = (z_1, \ldots, z_{16})$ be the 16-bit input word. The 17-bit word $\boldsymbol{y} = (y_1, \ldots, y_{17})$ is obtained by shuffling:

z_1	z_9	z_{10}	z_{11}	z_2	z_3	z_4	z_{12}	1	z_{13}	z_5	z_6	z_7	z_{14}	z_{15}	z_{16}	z_8

3.2 Encoding algorithm

Define the Boolean variables (the '+' denotes the logical 'or'-function) $L1 = y_1 + y_2 + y_3 + y_4$, $L2 = y_2 + \cdots + y_8$ and let $R1 = y_{14} + y_{15} + y_{16} + y_{17}$, $R2 = y_{10} + \cdots + y_{16}$.

Transmission of a word is based on the following 4-step algorithm:

If $L1L2R1R2$ then transmit $\boldsymbol{y}$ as is.

If $L1L2\overline{R1R2}$ then reshuffle and transmit:

z_1	z_9	z_{10}	z_{11}	z_2	z_3	z_4	1	0	0	z_5	z_6	z_7	1	z_{12}	z_{13}	z_8

If $\overline{L1L2}R1R2$ then reshuffle and transmit:

z_1	z_{12}	z_{13}	1	z_2	z_3	z_4	0	0	1	z_5	z_6	z_7	z_{14}	z_{15}	z_{16}	z_8

If $\overline{L1L2R1R2}$ then reshuffle and transmit:

z_1	z_{12}	0	1	z_2	z_3	z_4	1	0	1	z_5	z_6	z_7	1	0	z_{13}	z_8

It can easily be seen that error propagation due to any single channel bit error in the received codeword is restricted to at most one decoded byte error.

3.2.1 Systematic approach

The construction and properties of the codes discussed in the previous section lead to an interesting question. Essentially, at most eight bits are altered by the encoder; the remaining $n-8$ are transmitted unchanged. This property has both an effect on the error propagation and the complexity of the encoder/decoder.

3.2.2 Algorithmic approaches

The *sequence replacement technique* [11] converts source words of length p into $(0,k)$-constrained words of length $q = p+1$. The control bit is set to 'one' and appended at the beginning of the p-bit source word. If this $(p+1)$-bit sequence satisfies the prescribed constraint it is transmitted. If, on the other hand, the constraint is violated, i.e. a runlength of at least $k+1$ 'zeros' occur, we remove the trespassing $k+1$ 'zeros'. The position where the start of the violation was found is encoded in $k+1$ bits, which are appended at the beginning of the $p+1$-bit word. Such a modification is signaled to the receiver by setting the control bit to 'zero'. The codeword remains of length $p+1$. The above procedure is repeated until all forbidden subsequences have been removed. The receiver can reconstruct the source word as the position information is stored at a predefined position in the codeword. In certain situations the entire source word has to be modified which makes the procedure prone to error propagation. The class of rate $(q-1)/q$ $(0,k)$-constrained codes, $k = 1 + \lfloor q/3) \rfloor$, $q \geq 9$, was constructed to minimize error propagation [12]. Error propagation is confined to one decoded byte (8 bits) irrespective of the codeword length q.

4 Multi mode codes

Recently, the publications by Fair *et al.* [13] and Immink & Patrovics [14] on *guided scrambling* brought new insights into high-rate code design. Guided scrambling is a member of a larger class of related coding schemes called *multi-mode* codes. In multi-mode codes, the p-bit source word is mapped into $(m+p)$-bit codewords. Each source word $\boldsymbol{x}$ can be represented by a member of a *selection set* consisting of $L = 2^m$ codewords. Examples of such mappings are the guided scrambling algorithm presented by Fair *et al.* [13], the dc-free coset codes of Deng & Herro [15], and the scrambling using a Reed-Solomon code by Kunisa *et al.* [16]. A mapping is considered to be "good" if the selection set contains sufficiently distinct and random codewords.

The encoder opts for transmitting that codeword that minimizes, according to a prescribed criterion, for example, the low-frequency spectral contents of the encoded sequence. There are two key elements which need to be chosen judiciously: (a) the mapping between the source words and their corresponding selection sets, and (b) the criterion used to select the "best" word. It should be appreciated that the usage of multi-mode codes is not confined to the generation of dc-free sequences. Provided that 2^m is large enough and the selection set contains sufficiently different codewords, multi-mode codes can also be used to satisfy almost any channel constraint with a suitably chosen selection method. For given rate and proper selection criteria, the spectral content of multi-mode codes is very close to that of maxentropic (z)-constrained sequences. A clear disadvantage is that the encoder needs to generate all 2^m possible codewords, compute the criterion, and make the decision.

4.1 Weakly Constrained Codes

In the context of high-rate multi-mode codes, there is a growing interest in *weakly constrained codes* [17]. Weakly constrained codes do not work strictly to the rules, as they produce sequences that violate the constraints with probability P. It is argued that if the channel is not free of errors, it is pointless to feed the channel with perfectly constrained sequences. To illustrate the effectiveness of this idea we worked two examples.

Each source word of length p, $p >> 1$, is supplemented by m bits, so that the codeword length is $q = p + m$. The rate of the code is $R = p/q$. The m supplement bits make it possible to generate a selection set of $L = 2^m$ q-sequences. The particular method for generating the selection set is not discussed; we merely assume that the selection set comprises sufficiently distinct and random words. Assume we have a channel constraint which limits the channel capacity to C. Then the number of constrained codewords can be approximated by [18]

$$N(q) \approx A2^{qC}, \tag{5}$$

where A is a constant independent of q. The probability that in L drawings from randomly generated sequences we will not find a single sequence obeying the given constraint is simply

$$p = (1 - p_0)^L, \tag{6}$$

where

$$p_0 = \frac{N(q)}{2^q} \approx A2^{(C-1)q}. \tag{7}$$

As $L = 2^m$ and $m = (1 - R)q$, we have

$$p = \left(1 - A2^{(C-1)q}\right)^{2^{(1-R)q}}. \tag{8}$$

If for simplicity it is assumed that $A2^{(C-1)q} << 1$, we have

$$\ln(p) = -A2^{(C-R)q}. \tag{9}$$

4.2 $(0, k)$ codes

The number of self-concatable $(0, k)$ words, $N_c(n)$, of length n, equals the coefficient a_n of the following generating function

$$\sum a_i x^i = \frac{q(x)}{p(x)} = \frac{x(1 - x^{l+1})(1 - x^{r+1})}{(1 - x)(1 - 2x + x^{k+2})} \tag{10}$$

The parameters $l = \lceil k/2 \rceil$ and $r = k - l$ denote the maximum number of 'zeros' with which the words start or end, respectively. For large codeword length n the number of codewords can be approximated by

$$N(n) \approx A\lambda^n, \tag{11}$$

where λ is the largest real root of the characteristic equation

$$z^{k+2} - 2z^{k+1} + 1 = 0 \tag{12}$$

and the constant A, which is independent of n, equals

$$A = -\lambda \frac{q(1/\lambda)}{p'(1/\lambda)}. \tag{13}$$

Note that the channel capacity $C(0,k)$ satisfies

$$C(0,k) = \log_2 \lambda.$$

The capacity $C(0,k)$ equals $\log_2(\lambda)$, where λ is the largest root of [19]

$$x^{k+2} - 2x^{k+1} + 1 = 0. \tag{14}$$

For sufficiently large k, we derive

$$\lambda \approx 2(1 - \frac{1}{2^{k+2}}),$$

so that

$$C(0,k) \approx 1 - \frac{1}{\ln 2} 2^{-k-2}, \ , \ k \gg 1. \tag{15}$$

Combining the above with (9) we find

$$\ln(p) = -2^{(C-R)q}. \tag{16}$$

Figure 1 shows the probability, P, that no sequence taken from a selection set of size L of random sequences obeys the $(0,k)$ constraint. Let the code rate $R = 99/100$, the code word length $q = 400$, and the size of the selection set $L = 16$. Then we observe that with probability $P = 10^{-12}$ a codeword violates the $k = 9$ constraint. The alternative implementation [12] requires a rate of $R = 24/25$ –four times the redundancy of the weakly constrained code– to strictly guarantee the same (0,9) constraint.

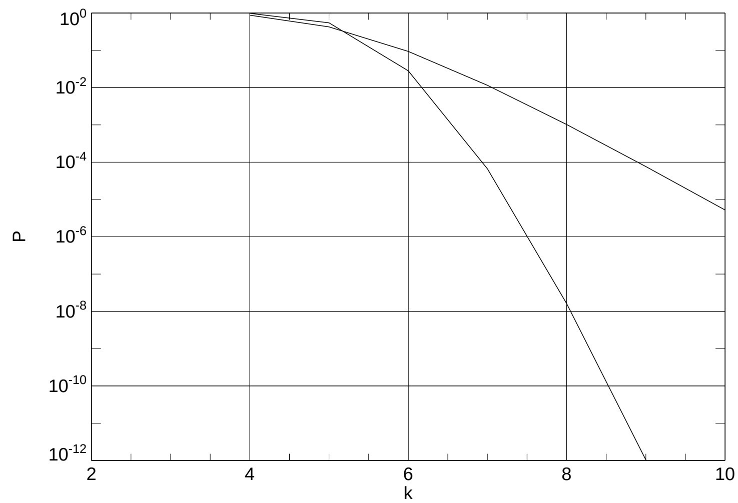

Figure 1: Probability that no sequence of L drawings from a selection set of random sequences satisfies the $(0, k)$ constraint. Code rate $R = 0.99$. Upper curve: codeword length $q = 200$, selection set size $L = 4$; lower curve: codeword length $q = 400$, selection set size $L = 16$.

5 Conclusions

An overview has been given of high-rate constrained codes for the noiseless recording channel. A new type of codes, called weakly constrained code, has been introduced. This code violates with (small) probability P the prescribed channel constraints.

References

[1] G.F.M. Beenker and K.A.S. Immink, 'A Generalized Method for Encoding and Decoding Runlength-Limited Binary Sequences', *IEEE Trans. Inform. Theory,* vol. IT-29, no. 5, pp. 751-754, Sept. 1983.

[2] S. Fukuda, Y. Kojima, Y. Shimpuku, and K. Odaka, '8/10 Modulation Codes for Digital Magnetic Recording', *IEEE Trans. Magn.,* vol. MAG-22, no. 5, pp. 1194-1196, Sept. 1986.

[3] A.X. Widmer and P.A. Franaszek, 'A Dc-balanced, Partitioned-Block, 8b/10b Transmission Code', *IBM J. Res. Develop.,* vol. 27, no. 5, pp. 440-451, Sept. 1983.

[4] H. Yoshida, T. Shimada, and Y. Hashimoto, '8-9 Block Code: A dc-free Channel Code for Digital Magnetic Recording', *SMPTE Journal,* vol. 92, pp. 918-922, Sept. 1983.

[5] K.A.S. Immink, 'A Practical Method for Approaching the Channel Capacity of Constrained Channels', *IEEE Trans. Inform. Theory,* vol. IT-43, no. 5, pp. 1389-1399, Sept. 1997.

[6] T.M. Cover, 2204'Enumerative Source Coding', *IEEE Trans. Inform. Theory,* vol. IT-19, no. 1, pp. 73-77, Jan. 1973.

[7] F.K. Bowers, US Patent No. 2,957,947, 1960.

[8] D.E. Knuth, 'Efficient Balanced Codes', *IEEE Trans. Inform. Theory,* vol. IT-32, no. 1, pp. 51-53, Jan. 1986.

[9] P.S. Henry, 'Zero Disparity Coding System', US Patent 4,309,694, Jan. 1982.

[10] N. Alon, E.E. Bergmann, D. Coppersmith, and A.M. Odlyzko, 'Balancing Sets of Vectors', *IEEE Trans. Inform. Theory,* vol. IT-34, no. 1, pp. 128-130, Jan. 1988.

[11] A.J. de Lind van Wijngaarden and K.A.S. Immink, 'Construction of Constrained Codes using Sequence Replacement Techniques', Submitted *IEEE Trans. Inform. Theory,* 1997.

[12] K.A.S. Immink and A.J. de Lind van Wijngaarden, 'Simple high-rate constrained codes', *Electronics Letters,* vol. 32, no. 20, pp. 1877, Sept 1996

[13] I.J. Fair, W.D. Gover, W.A. Krzymien, and R.I. MacDonald, 'Guided Scrambling: A New Line Coding Technique for High Bit Rate Fiber Optic Transmission Systems', *IEEE Trans. Commun.,* vol. COM-39, no. 2, pp. 289-297, Feb. 1991.

[14] K.A.S. Immink and L. Patrovics, 'Performance Assessment of DC-Free Multimode Codes', *IEEE Trans. Commun.,* vol. COM-45, no. 3, March 1997.

[15] R.H. Deng and M.A. Herro, 'DC-Free Coset Codes', *IEEE Trans. Inform. Theory,* vol. IT-34, no. 4, pp. 786-792, July 1988.

[16] A. Kunisa, S. Takahashi, and N. Itoh, 'Digital Modulation Method for Recordable Digital Video Disc', *IEEE Trans. Consumer Electr.,* vol. 42, pp. 820-825, Aug. 1996.

[17] K.A.S. Immink, 'Weakly Constrained Codes', *Electronics Letters,* vol. 33, no. 23, pp. 1943-1944, Nov. 1997.

[18] C.E. Shannon, 'A Mathematical Theory of Communication', *Bell Syst. Tech. J.,* vol. 27, pp. 379-423, July 1948.

[19] K.A.S. Immink, *Coding Techniques for Digital Recorders,* Prentice-Hall International (UK) Ltd., Englewood Cliffs, New Jersey, 1991.