A. M. Patel

Zero-Modulation Encoding in Magnetic Recording

Abstract: This paper deals with waveform encoding methods in which binary data are mapped into constrained binary sequences for shaping the frequency spectrum of corresponding waveforms. Short and long pulse widths in the waveform are limited by constraints on the minimum and maximum run-lengths of zeros in the coded sequences. These constraints reduce the intersymbol interference in magnetic recording and provide an adequate rate of transition for accurate clocking. Signal power at low frequencies is limited by means of a constraint on a parameter that corresponds to the maximum imbalance in the number of positive and negative pulses of the waveform. This constraint on the maximum accumulated dc charge also eliminates the zero-frequency component.

Zero modulation is one such code that is especially suitable for magnetic recording channels. The encoding and decoding algorithm is presented. A one-to-one correspondence between binary data and constrained sequences is established by creating data states that are isomorphic to the charge states having the same growth rate. Sequences with other values of run-length and charge constraint are examined as candidates for other codes with zero dc component.

Introduction

In magnetic recording, digital information is recorded as magnetic-flux transitions in predetermined, fixed-length partitions of the magnetic media. Unlike the binary convention of "up" and "down" states in data transmission, the intervals in which the magnetic recording transitions occur are assigned the value "one" and those with no transition are assigned the value "zero." The readback process consists in detecting these flux transitions with a transducer positioned over these intervals in a fixed time sequence, thus producing a read waveform.

If clocking data are derived from the read waveform, the transitions must occur frequently enough to provide synchronization pulses for the free-running clock. On the other hand, consecutive transitions must be far enough apart to limit the interference to an acceptable level for reliable detection. For this purpose we encode binary data into coded binary sequences that correspond to waveforms in which the maximum and minimum distances between consecutive transitions are constrained by prescribed coding rules. Phase encoding (PE), modified frequency modulation (MFM), synchronized NRZI (NRZI-S), and run-length-coded NRZI (RLC-NRZI) are some of the encoding methods used in digital magnetic recording. Examples of their use are PE in IBM tape machines, MFM in the IBM 3330 disk file, NRZI-S in the IBM 1311 and 1405 disk files, and RLC-NRZI in the latest models of IBM 3420 tape machines.

Another reason for encoding the binary data into binary sequences is the shaping of the frequency spectrum of the signal waveform. The maximum and minimum distances between consecutive transitions correspond to the minimum and maximum pulse rates in the waveform, respectively. This range of pulse rate in the highly nonlinear read-write process of magnetic recording causes irregular read signal amplitudes and results in phase-shift errors. For any given data density it is desired to constrain this range of pulse rate to be as narrow and as low as possible. This, however, is not the only consideration if the channel uses ac coupling networks to process read-write signals. For example, in the IBM 3850 Mass Storage System, the read-write function is performed by a transformer-coupled rotary head, in which case signal waveforms should not have a dc component. A dc component in the waveform results in a nonzero average value of the amplitude and causes charge accumulation at any ac coupling element in the channel. A constraint on the maximum accumulated charge is an effective means of reducing the signal distortion caused by the ac coupling networks. The result is a reduction of errors in signal detection. In a waveform corresponding to a binary coded sequence, the accumulated charge increases by one unit for a positive pulse and decreases by one unit for a negative pulse from digit to digit in the waveform. Thus, the accumulated charge at any digit in a binary coded sequence is the difference between the numbers of positive and negative pulses in the corresponding waveform up to that digit.

A data encoding method, then, is in general a one-to-one mapping of binary data into constrained binary sequences. Such coded sequences may be denoted by the design parameters $(d, k; c)$ corresponding to the following three constraints:

1. The shortest run-length (sequence) of zeros between any two consecutive ones in the coded sequence is d digits. This determines the minimum distance between the recorded transitions and hence the highest transition density.
2. The longest run-length (sequence) of zeros between any two ones in the coded sequence is k digits. This determines the maximum distance between the recorded transitions and hence the lowest transition density.
3. The accumulated charge at any digit position in the sequence is bounded by $\pm c$ units.

If, on the average, x data bits require y binary digits, where $x \leq y$, in a coded sequence, the ratio of m/n is called the rate of the code. The highest recorded density of magnetic transitions for given data density is determined by x/y and the minimum run-length constraint d. This density ratio DR is a measure of recording efficiency and is given by

$$DR = \frac{\text{data density}}{\text{highest recorded density}} = \left(\frac{x}{y}\right)(d+1).$$

For given data rate and density, the lower rate code requires faster clock and more complex detection circuits. On the other hand, a higher density ratio DR is desired to limit the intersymbol interference to a reasonable degree. Codes with rates ranging from 0.5 to one are used in magnetic recording. Codes with much lower rates than 0.5 are usually impractical.

In this paper we present a theory and the implementation of such an encoding method. In particular, we deal with 0.5-rate run-length-limited codes that produce waveforms with no zero-frequency (dc) component. A specific code is presented called zero modulation (ZM), which is designed with parameters $d = 1$, $k = 3$, and $c = 3$ and is the first binary code in which the waveform possesses zero dc component and yet has a density ratio close to one. Heretofore, phase encoding was the only known binary code with the zero-dc property. However, it is characterized by a very low recording efficiency ($DR = 0.5$). Delay modulation [1] or modified frequency modulation (MFM) provided high recording efficiency ($DR = 1$) but the accumulated charge in the waveform often increased indefinitely. Zero modulation combines the desirable properties of phase encoding and delay modulation. Franaszek [2] has reported many (d, k) run-length-constrained binary codes. Other related results are reported by Tang [3], Gabor [4], and Frieman and Wyner [5]. Some dc-free ternary codes were reported by Franaszek [6], who used a parameter called the digital sum variation. This parameter is related to our accumulated charge in binary codes. Croisier [7] reported compilation of many results on pseudoternary codes with zero dc component. These nonbinary codes are widely used in data transmission but are not as suitable for magnetic recording because of the highly nonlinear characteristics of the magnetic recording process. Zero modulation was designed for, and is used in, magnetic recording. It is however, equally suitable for transmission of binary data over other types of channels.

The second section presents the ZM algorithm and includes discussion of the waveform parameters and practical implementation with limited memory. The third section provides the proof for the algorithm using a method called "isomorphism of state diagrams." The waveform sequences are analyzed by means of state diagrams. The growth rate of constrained sequences is compared with that of binary data sequences. Isomorphic state diagrams are created for data and constrained sequences which then provide the one-to-one mapping between them. In the fourth section the constraints are exploited to provide detection of errors in zero modulation sequences at the receiver. A subsequent section provides a useful result regarding synchronization sequences for clocking of zero modulation patterns.

The appendix gives a mathematical derivation for the growth rate of the constrained sequences. It is shown that the growth rate of ZM sequences is two, the same as that of binary sequences. The appendix also includes some results regarding other 0.5-rate charge-constrained codes with higher density ratios and different run-length-limit parameters.

Zero-modulation algorithm

In this section we present the zero-modulation algorithm for mapping binary data sequences into constrained sequences with parameters $d = 1$, $k = 3$, and $c = \pm 3$. This algorithm was the outcome of a multifaceted approach involving most desirable parameters for magnetic recording and the structure of constrained binary sequences. The initial approach was to look for "good" block codes of fixed-length constrained sequences by means of a computer search and then generate a one-to-one mapping using minimum logic. The results showed a rather surprising structure of constrained binary sequences which finally led to the generalized mapping with a convolutional algorithm which is presented here. The background results on block codes are omitted for the sake of brevity. The coded sequences generated by the ZM algorithm with limited memory, discussed in the latter part of this section, turned out to be the same as those obtained in the earlier results using block codes. The algorithm appears in this section without a formal proof of uniqueness of the mapping. The proof is deferred to the third section, entitled "State diagrams," where the data sequences and constrained sequences are represented by state diagrams and the proof of unique-

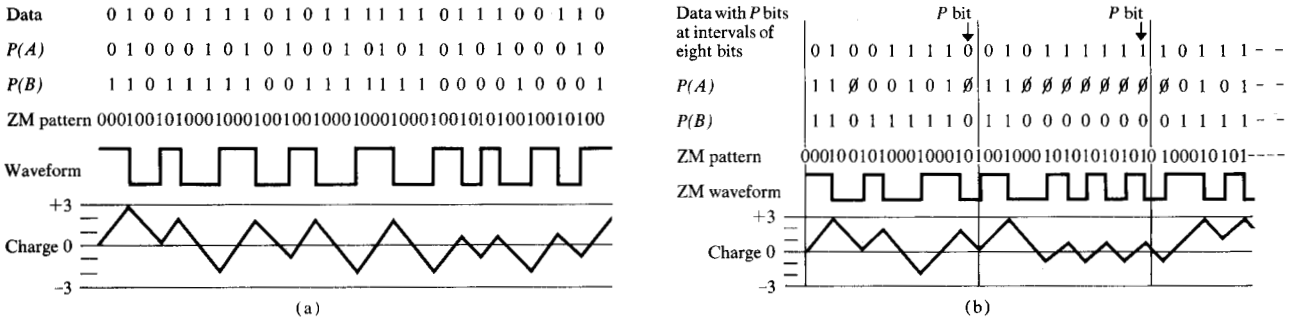


Figure 1 Relations among parameters of the data sequence and the corresponding ZM waveform. (a) Example of ZM waveform for the case of unlimited memory. (b) ZM waveform for the case of limited memory.

ness is obtained from the isomorphism of these state diagrams.

The algorithm maps every data bit into two binary digits in a sequential manner. The resulting binary sequence is then converted into a waveform using NRZI rules, i.e., a transition for one and no transition for zero in the binary sequence. The mapping of the data bit into a two-digit ZM pattern is a nonlinear function of the preceding and following data sequences and requires an encoder with memory. In a practical implementation, the amount of memory can be limited to f bits, where f is any positive integer, by adding a small amount of redundancy. This procedure is described later. First, the ZM algorithm is given in its functional form which, in general, requires unlimited memory.

• *ZM algorithm with unlimited memory*

Let d_0 denote the data bit to be encoded in the sequential encoding process, d_{-1} the preceding data bit, and d_{+1} the following data bit. Similarly, let a_0b_0 denote the ZM digits corresponding to the data bit d_0, with $a_{-1}b_{-1}$ denoting the ZM digits corresponding to the data bit d_{-1}, and $a_{+1}b_{+1}$ denoting the ZM digits corresponding to the data bit d_{+1}. The algorithm is given in terms of these digits and two parity functions requiring look-ahead and look-back. These functions are denoted by $P(A)$ and $P(B)$, signifying look-ahead (A) and look-back (B), respectively. The two functions are defined as follows.

$P(A)$: *Look-ahead, one-sequence-parity* is the modulo-2 count of ones in the binary sequence from d_0 to the next zero in the following data. This is the parity of the sequence of ones looking ahead from d_0 (including d_0). For example, in the data sequence 01011110, $P(A)$ is 1 at the second, fifth, and seventh digits from the left; $P(A)$ is 0 if d_0 is 0.

$P(B)$: *Look-back zero-parity* is the accumulated modulo-2 count of zeros from the start of the data up to and including d_0. For example, in the data sequence 01011110, $P(B)$ is one at the first, second, and eighth digits from the left.

The encoding algorithm may be described as follows:

$d_0 \rightarrow a_0b_0$	*Condition*
$0 \rightarrow 10$	$d_{-1} = 0$
$0 \rightarrow 10$	$d_{-1} = 1$ and $a_{-1}b_{-1} = 00$
$0 \rightarrow 00$	$d_{-1} = 1$ and $a_{-1}b_{-1} \neq 00$
$1 \rightarrow 10$	$d_{-1} = 0$ and $P(A) = 0$ and $P(B) = 1$
$1 \rightarrow 10$	$d_{-1} = 1$ and $a_{-1}b_{-1} = 00$
$1 \rightarrow 00$	$d_{-1} = 1$ and $a_{-1}b_{-1} = 10$
$1 \rightarrow 01$	otherwise.

The decoding algorithm may be described as follows:

$a_0b_0 \rightarrow d_0$	*Condition*
$10 \rightarrow 1$	$a_1b_1 = 00$
$10 \rightarrow 0$	$a_1b_1 \neq 00$
$00 \rightarrow 1$	$a_{-1}b_{-1} = 10$
$00 \rightarrow 0$	$a_{-1}b_{-1} \neq 10$
$01 \rightarrow 1$	none.

Alternatively, the *encoding rule* for the mapping of data into a ZM sequence can be given by the binary logic functions

$$a_0 = \bar{d}_0\bar{d}_{-1} + d_0\bar{d}_{-1}\,\overline{P(A)}\,P(B) + d_{-1}\bar{a}_{-1}\bar{b}_{-1}, \quad (1)$$

$$b_0 = d_0[P(A)\,\bar{d}_{-1} + \overline{P(B)} + b_{-1}]. \quad (2)$$

The *decoding rule* for mapping a ZM sequence into data can be given by the binary logic function

$$d_0 = b_0 + a_0\bar{a}_1\bar{b}_1 + \bar{a}_0a_{-1}\bar{b}_{-1}. \quad (3)$$

It is convenient to use the following boundary conditions in the encoding and decoding process: $d_{-1} = 1$ and $P(B)_{-1} = 0$ are assumed at the first data bit, and $d_i = 0$ at the last data bit.

The example, Fig. 1(a), illustrates the relationships among the various parameters of the data sequence and the corresponding ZM waveforms. Note that the maximum pulse width in terms of data bit width is two units and the minimum pulse width is one unit. Some pulses are 1.5 units wide.

The ZM waveform looks similar to the well known MFM, or delay modulation, waveform [1]. The important difference, however, is that the MFM waveform contains the dc component, and the accumulated charge often increases indefinitely. The maximum accumulated charge in the ZM waveform is ±3 units in terms of the coded digit intervals.

The functions $P(A)$ and $P(B)$ represent information from the following and previous data sequences, respectively; $P(B)$ is the accumulated parity of the number of zeros and hence can be derived simply by updating the content of a one-bit storage as the data bits are encoded; $P(A)$, however, depends on the run length of ones in the following data sequence. Thus, the memory requirement for computation of $P(A)$ is, in general, unlimited. The section following describes the ZM algorithm in which the memory requirement is limited to f bits, where f is any positive integer.

• *ZM algorithm with limited memory*

Consider the functions in Eqs. (1) and (2) when $P(B)$, the accumulated look-back zero-parity value, is zero. These functions can be written as

$$\left.\begin{aligned} a_0 &= \bar{d}_0\bar{d}_{-1} + d_{-1}\bar{a}_{-1}\bar{b}_{-1}, \\ b_0 &= d_0 \end{aligned}\right\} \text{ if } P(B) = 0.$$

The mapping is thus independent of the look-ahead data sequence when the look-back zero-parity is even.

Now consider a continuous data stream in which a digit P is inserted at fixed intervals of f bits. This P-bit in the $f+1$ position in the modified data stream sets $P(B)$ equal to zero at the end of each interval section of $f+1$ bits. This implies that the look-ahead parity $P(A)$ of the sequence of ones at the end and beginning of any section has no effect on the ZM mapping in the modified data sequence. [See Fig. 1(b): $P(A)$ is denoted by ∅, which is a DON'T CARE value when $P(B) = 0$.] The only sequences of ones affecting the mapping, then, are those which exist between two zeros in the same section. The longest such one-sequence has length $f-1$ digits, with a zero at the beginning and a zero at the end of an $f+1$ digit section. Thus, the memory requirement for computation of $P(A)$ is $f-1$ bits. It will be observed later that $P(B) = 0$ corresponds to a zero value of accumulated charge in the coded waveform. Consequently, at the end of every section the accumulated charge is zero. The ZM algorithm with limited memory then has the following two important modifications:

1. The data sequence is modified by inserting an extra P-bit at the end of every section of f data bits, where P is given by the value of $P(B)$ at position f.
2. The computation of the function $P(A)$ at any data bit is truncated beyond the following $f-1$ data bits.

Implementation of the ZM algorithm with eight bits of memory is shown in the following example. To limit the memory, one bit of redundancy is added for every f bits of data. The percentage redundancy decreases with f and, hence, the memory size increases.

The look-ahead one-sequence parity function $P(A)$ is given by a binary logic function of the data stored in f bits of memory,

$$P(A) = d_0\bar{d}_1 + d_0d_2\bar{d}_3 + d_0d_2d_4\bar{d}_5 + \cdots$$
$$\cdots + d_0d_2d_4\cdots d_{t-4}\bar{d}_{t-3} + d_0d_2d_4\cdots d_{t-4}d_{t-2},$$

where $t = f$ if f is even and $t = f-1$ if f is odd. The look-back zero-parity $P(B)$ is given by the function $P(B) = P(B)_{-1} \forall \bar{d}_0$, where $P(B)_{-1}$ is the value of $P(B)$ at the previous data bit and ∀ represents the binary EXCLUSIVE OR function. Note that the encoding process is delayed by f bit-periods in a continuous stream of data. The decoding process is delayed by only a one bit-period. Thus, the decoding errors in ZM do not propagate.

Table 1: Definition of accumulated charge

Pattern and waveform	*Value of s*
0 1	0
0 1 0 0	2
0 1 0 0 0 1	−2

State diagrams

• *State diagram of constrained sequences*

The accumulated charge in a binary sequence is defined in the following manner. Every digit in the coded sequence corresponds to one of the two signal levels in the waveform, starting at the center of the digit position and ending at the center of the next digit position. Let n_1 and n_2 denote the numbers of digits corresponding to the two levels, respectively, in a waveform for a sequence of $n_1 + n_2$ digit lengths. Then, the accumulated charge s at the end of this waveform is given by $s = \pm(n_1 - n_2)$.

Table 2 Possible state transitions.

Transition symbol (*catenation*)	*New state values* s_0	r_0	*Conditions for transition*
0	$s_0 = s_{-1} + 1$	$r_0 = r_{-1} + 1$	$s_{-1} + 1 \leq c, r_{-1} + 1 \leq k$
1	$s_0 = -s_{-1} + 1$	$r_0 = 0$	$-s_{-1} + 1 \leq c, d \leq r_{-1}$

Table 3 State transitions, for 0.5-rate codes.

Transition symbol a_0b_0	*Present state values* s_0	r_0	*Conditions for transition*
00	$s_0 = s_{-1} + 2$	$r_0 = r_{-1} + 2$	$s_{-1} + 2 \leq c, r_{-1} + 2 \leq k$
01	$s_0 = -s_{-1}$	$r_0 = 0$	$d \leq r_{-1} + 1 \leq k$
10	$s_0 = -s_{-1} + 2$	$r_0 = 1$	$2 - s_{-1} \leq c, d \leq r_{-1}$
11	$s_0 = s_{-1}$	$r_0 = 0$	$d = 0$

Table 4 State identification for all possible values of s and r in a ZM state diagram.

s/r	0	1	2	3
+2	X	A	A′	D
0	Y	B	C	N(3)
−2	Z	N(1)	N(2)	N(4)

Only the absolute value of s is important, because the sign of s changes with the choice of starting the waveform with a positive or a negative level. For convenience in iterative computation of s from digit to digit, the convention followed is that s is the value of accumulated charge associated with the waveform in which the last level is a positive level. The waveform patterns in Table 1 illustrate this convention.

It is easily seen that the value of s in any pattern can be computed iteratively from digit to digit. With the above convention, the "present" value s_0 of s depends only on the "previous" value of s_{-1}, and the "present" digit being catenated to the pattern. For example, catenation of a one or a zero to a pattern causes the following changes in the charge values:

Catenation of $0 \rightarrow s_0 = s_{-1} + 1$,

Catenation of $1 \rightarrow s_0 = -(s_{-1} - 1)$.

These catenations can be considered as state transitions in descriptions of the constrained sequences by means of state diagrams. The states can be characterized by two parameters, namely, the accumulated charge s and the run-length r of zeros at the end of sequence, using the identifying pair $(s; r)$. The constraints on s and r are $|s| \leq c$ and $d \leq r \leq k$, respectively, where $\pm c$ represents the value of maximum accumulated charge, and d and k denote the minimum and maximum run-lengths of zeros. Table 2 gives the possible state transitions. In any binary sequence, the state at every symbol can be identified as shown in the example, Fig. 2. Alternatively, the run-length sequences can be represented as a series of state transitions on a state diagram.

In general, the encoding of binary data into constrained sequences requires n coded digits for every x data bits, where $x < y$. It is convenient to use symbols of at least y digits in dealing with such x/y-rate codes. In particular, for a 0.5-rate code, two-digit symbols denoted by a_0b_0 are used as state transitions. Zero modulation is a 0.5-rate code.

Table 3 represents the state transitions for 0.5-rate codes. The new state values (s_0, r_0) are given in terms of the previous state values (s_{-1}, r_{-1}), using concatenations of two digits a_0b_0 at a time. This table is constructed using the transition information in Table 2.

The state diagram can now be constructed for ZM patterns. The $(d, k; c)$ constraints in zero modulation are $(1, 3; \pm 3)$. It is clear from Table 3 that the charge values

Figure 2 Example showing how, in a binary sequence, the state at every symbol can be identified by two parameters: the accumulated charge s and the run-length r.

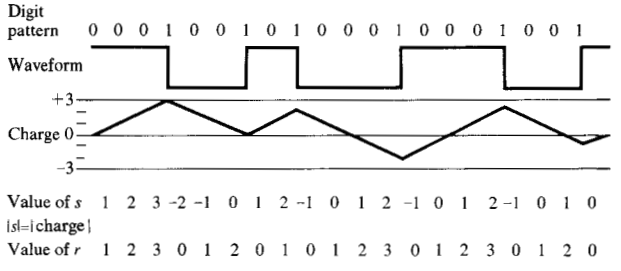


Table 5 All possible state transitions for ZM sequences, according to Table 3 rules. (Note: *S* indicates violation of the charge constraint; *R* indicates violation of the run-length constraint.)

Previous state:		*New state:* (s_0, r_0) *for three conditions*		
(s_{-1}, r_{-1})	*Name*	$a_0b_0 = 01$	$a_0b_0 = 10$	$a_0b_0 = 00$
(2, 0)	X	(−2, 0) Z	*R*	*S*
(0, 0)	Y	(0, 0) Y	*R*	(2, 2) A′
(−2, 0)	Z	(2, 0) X	*S R*	(0, 2) C
(2, 1)	A	(−2, 0) Z	(0, 1) *B*	*S*
(0, 1)	B	(0, 0) Y	(2, 1) *A*	(2, 3) D
(−2, 1)	N(1)	(2, 0) X	*S*	(0, 3) N(3)
(2, 2)	A′	(−2, 0) Z	(0, 1) B	*S R*
(0, 2)	C	(0, 0) Y	(2, 1) A	*R*
(−2, 2)	N(2)	(2, 0) X	*S*	*R*
(2, 3)	D	*R*	(0, 1) B	*S R*
(0, 3)	N(3)	*R*	(2, 1) A	*R*
(−2, 3)	N(4)	*R*	*S*	*R*

must be an even number at the end of each pair a_0b_0 in the coded pattern, with the practical assumption that s is zero at the starting point. Thus, the only possible values of s in ZM state diagram are 0, +2, and −2. Table 4 lists all possible states with the above constraints. It is seen later that the states N(1), N(2), N(3), and N(4) of Table 4 do not exist and that the states A and A′ can be merged together. Table 5 lists all possible state transitions according to the rules of Table 3, using transition symbol a_0b_0.

None of the states maps into the states N(1), N(2), and N(4). Also, only N(1) maps into N(3). Since N(1) cannot be the initial state, it is obvious that states N(1), N(2), N(3), and N(4) are nonexistent. The states A and A′ can be combined. This is true because both map into the same states, namely Z and B, when $a_0b_0 = 01$ and 10, respectively, and 00 is an inadmissible transition from both. The resulting state diagram is shown in Fig. 3. Inadmissible states and transitions are not shown. The state A′ is merged into A.

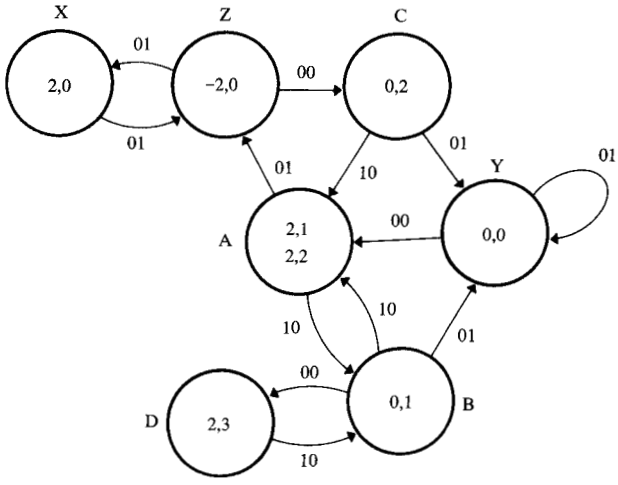
In the Appendix we define the growth rate of constrained sequences and show how to determine the growth rate of the $(1, 3; \pm 3)$ sequences from the eigenvalues of the state transition matrix. A necessary condition for a one-to-one mapping is that the growth rate be at least two, i.e., the same as that of the binary data sequences. If the growth rate is higher than two, then some states or state transitions must be eliminated or modified to reject some of the sequences. If the growth rate is less than two, then, of course, a one-to-one mapping is not possible unless some of the data sequences are rejected. It is shown that the $(1, 3; \pm 3)$ constrained sequences possess a growth rate of exactly two (Appendix) with two-digit symbol catenations. This satisfies the necessary condition. The study of eigenvalues and eigenvectors of the state transition matrix shows much of the information leading to the structure of the mappings.

Table 6 Identification of states in the data sequences in terms of four parameters.

State	d_0	$P(B)$	$P(B1)$	$P(A)$
α	0	1	∅	∅
β	0	0	∅	∅
γ	1	0	∅	∅
ψ_1	1	1	0	0
ψ_2	1	1	0	1
μ_1	1	1	1	1
μ_2	1	1	1	0

The fact that the growth rate turns out to be exactly two is an intriguing mathematical coincidence. Or is it? Could it not be the manifestation of still unknown but more general structural properties of constrained sequences and related band-limited signaling waveforms? This question is open for future investigation.

Figure 3 The ZM state diagram (inadmissible states and transitions are not shown).



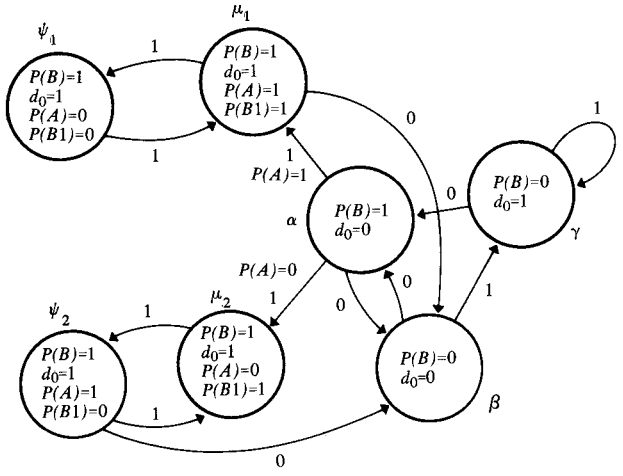


Figure 4 State diagram constructed for data sequences similar to that for ZM sequences (the state transitions are listed in Table 7).

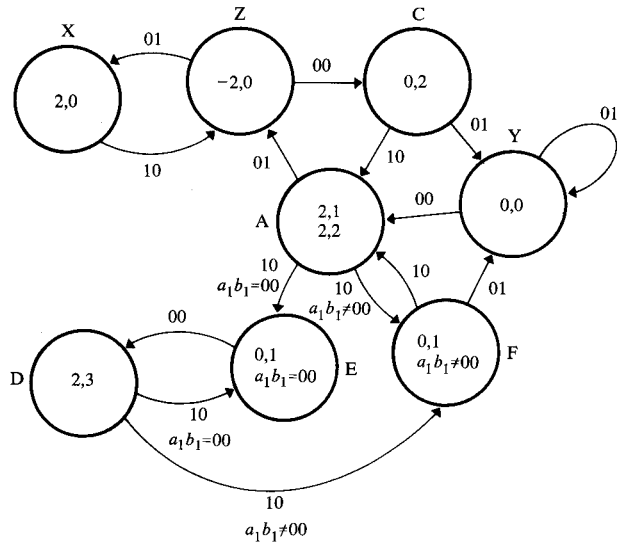


Figure 5 Modified ZM state diagram, in which states E and F are created by splitting state B.

• *State diagram for data sequences*

Next to be presented is a sufficient condition for a one-to-one correspondence between ZM sequences and data sequences. The ZM state diagram of Fig. 1 shows the difficulty of encoding binary data into sequences with the run-length and charge constraints. There are two exits from each of the states A, B, C, Y, and Z, and thus the two distinct transitions can be used to encode a binary one and a binary zero.

By contrast, states D and X each have only a single exit, and only one binary number can be represented when the encoding operation is in these states. For this purpose, state diagrams are constructed for data sequences that are similar to those for the ZM sequences. Later, the ZM state diagram is modified by splitting and combining some of the states to make it fully isomorphic to the data state diagram, thus establishing a one-to-one correspondence between the transitions.

Let d_0 denote the present data bit and let d_{-1} denote the previous data bit in a data sequence; $P(A)$ is the look-ahead one-sequence-parity function and $P(B)$ is the look-back zero-parity function, as defined in the previous section. The function $P(B1)$, the look-back one-sequence parity, is the parity of a sequence of ones looking back from, and including d_0; $P(B1)$ is 0 if d_0 is 0 and can be interatively computed as $P(B1) = d_0\ \overline{P(B1)}_{-1}$, where $P(B1)_{-1}$ is the value of $P(B1)$ at the previous data bit. As an example, in the data sequence 01011110, $P(B1)$ is one at the second, fourth, and sixth digits from the left. The states in the data sequence are characterized by the values of the functions $P(A)$, $P(B)$, $P(B1)$, and the ending bit d_0. Table 6 identifies all the states in terms of these parameters.

In any data sequence the state at every bit position can be identified as shown in the following example.

Example States in data sequence

Data sequence	0	1	0	0	1	1	1	1	0	1	0
$P(B)$	1	1	0	1	1	1	1	1	0	0	1
$P(B1)$	0	1	0	0	1	0	1	0	0	1	0
$P(A)$	0	1	0	0	0	1	0	1	0	1	0
State	α	μ_1	β	α	μ_2	ψ_2	μ_2	ψ_2	β	γ	α.

Any data sequence can thus be traced on a state diagram as transitions on a series of states. Note that when $d_0 = 1$ the state transitions are dependent on the value of $P(A)$. All state transitions are listed in Table 7 and are shown on the state diagram of Fig. 4.

• *Isomorphism of state diagrams*

It is obvious that, to establish a one-to-one correspondence, the states B and C in the ZM state diagram must be modified in accordance with the states μ_2 and β in the state diagram for data sequences. First, B is split into two states E and F such that B mapping into D is called E and B mapping into A or Y is called F. This split can be identified by the next pattern, $a_1b_1 = 00$ or $a_1b_1 \neq 00$. In particular, the transition to the state E assumed that a_1b_1 was known and that $a_1b_1 = 00$. The modified state diagram is given in Fig. 5. States C and F can be combined, since the outgoing transitions are identical. The resultant state diagram is given in Fig. 6.

The following assertions can now be made:

1. There is a one-to-one correspondence of the states (including all possible transitions) between the state diagrams of Figs. 4 and 6. The states α, β, γ, μ_1, ψ_1,

μ_2, and ψ_2 in Fig. 4 are isomorphic to the states A, G, Y, Z, X, E, and D, respectively, in Fig. 6.

2. Every data sequence of length n traces a distinct path, given by a series of n states, on the state diagram of Fig. 4 (starting from state γ and ending in any state other than ψ_1 or μ_2).
3. Corresponding to each path traced by n-digit data sequence on state diagram of Fig. 4, there is an isomorphic path on the state diagram of Fig. 6 traced by a $2n$-digit binary pattern with ZM constraints.
4. Any two distinct paths of given length on Fig. 4 represent two distinct data sequences (with a restriction that a path starts from state γ and does not end in state ψ_1 or μ_2).
5. Any two paths of given length in Fig. 6 represent two distinct ZM sequences (with the restriction that a path starts from state Y and does not end in state X or E).

At this time the theorem can be stated for the uniqueness of the mapping given by the ZM algorithm.

Theorem Every data sequence of length n can be uniquely represented by a ZM pattern of length $2n$ digits, as given by the ZM encoding algorithm.

It can be easily verified that the ZM encoding and decoding algorithms preserve the isomorphism of the state diagrams of Figs. 4 and 6. The proof of uniqueness of the mappings then follows from the above five assertions.

Note that $P(B)$ represents only one bit of information from the past and that $P(A)$ represents only one bit of information from the future in the data sequence. Although computation of $P(A)$ requires an infinite look-ahead in the data sequence, this look-ahead can be reduced to any finite number by simply introducing one-bit redundancy. This was shown in the second section in the discussion of the ZM algorithm with limited memory.

Error detection in ZM patterns

The patterns of digits generated by means of the ZM algorithm satisfy various constraints, including the parity bit on the data in the case of ZM with limited memory. These constraints provide a powerful check capability for bit-detection errors and synchronization errors at the receiver.

ZM patterns must possess run-lengths of one, two, or three zeros between two ones in the ZM pattern. Two consecutive ones or four or more consecutive zeros in the ZM pattern indicate a "pick up" or a "drop out" error, respectively, in the corresponding waveform. Let a_0b_0 denote the pair of ZM digits to be decoded in the sequential decoding process; $a_{-1}b_{-1}$ denote the preceding pair of ZM digits; and a_1b_1 denote the following pair of ZM digits. The error functions for the minimum run-length constraint and the maximum run length constraint are given by E_1 and E_2 as

$$E_1 = a_0b_0 + b_0a_1,$$

$$E_2 = \bar{b}_{-1}\bar{a}_0\bar{b}_0\bar{a}_1 + \bar{a}_0\bar{b}_0\bar{a}_1\bar{b}_1.$$

The violation of charge constraint can be checked by keeping a continuous count of the accumulated charge, which must be within ± 3 units. A simple and convenient check of this constraint, however, can be obtained in terms of the functions $P(B)$ and $P(B1)$. Table 5 shows that the charge constraint violation can occur without a violation in the run-length constraint. This happens if and only if the sequence received at d_0 is in state A or X, with the next transition $a_1b_1 = 00$. Consider these two transitions separately:

In state A at d_0 with $a_1b_1 = 00$ The pattern in question here is $a_0b_0a_1b_1 = 1000$. The decoder operates correctly up to d_{-1} with $P(B) = 0$ at d_{-1}, indicating state β on the data state diagram. The pattern $a_0b_0a_1b_1 = 1000$ is decoded as $d_0d_1 = 11$, leaving $P(B) = 0$ at d_0. This indi-

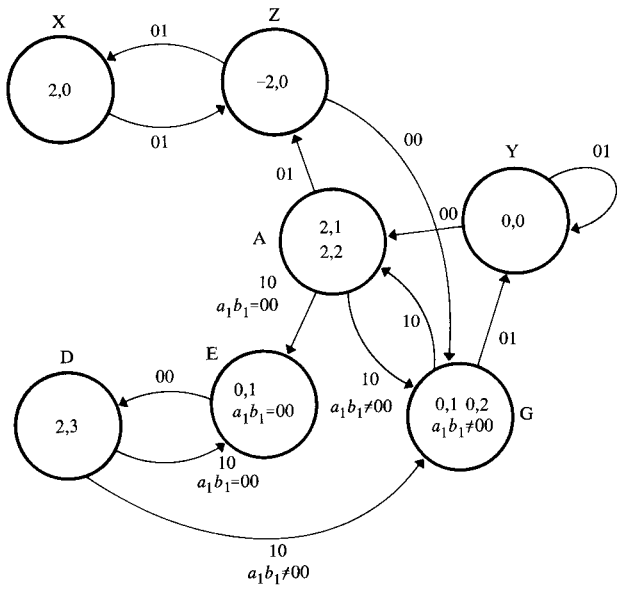


Figure 6 Isomorphic ZM state diagram, in which states C and F of Figure 5 are combined into the new state G.

Table 7 All state transitions in the data sequences.

	Present state when		
Previous state	$d_0 = 0$	$d_0 = 1, P(A) = 0$	$d_0 = 1, P(A) = 1$
α	β	μ_2	μ_1
β	α	γ	γ
γ	α	γ	γ
ψ_1			μ_1
ψ_2	β	μ_2	
μ_1	β	ψ_1	
μ_2			ψ_2

cates an error, since 1000 occurs only when $P(B) = 1$ for state transitions A → E → D. The error function corresponding to this case is thus given by

$$E_3 = a_0\bar{b}_0\bar{a}_1\bar{b}_1\overline{P(B)}.$$

In state X at d_0 with $a_1b_1 = 00$ The pattern in question here is $a_0b_0a_1b_1 = 0100$. The decoder correctly decodes $a_0b_0 = 01$ into $d_0 = 1$, with $P(B) = 1$ and $P(B_1) = 0$, corresponding to the state ψ_1 on the data state diagram. The pattern $a_1b_1 = 00$, however, indicates an error, since $P(B) = 1$ and $P(B1) = 0$ occur only on states X and D and a_1b_1 cannot be 00 from those states. Thus, the error function for this case is

$$E_4 = \bar{a}_1\bar{b}_1P(B)\overline{P(B1)}.$$

This exhausts all possible violations of the charge constraint and run-length constraints.

Next is a simple but effective check on synchronization errors as well as random errors. The value of $P(B)$ is zero at every memory boundary of $f + 1$ data bits. The charge value is also zero at the memory boundary. These two checks are equivalent since they check for the isomorphic states β and γ on the data state diagram or G and Y on the ZM state diagram, respectively. The check is then given by the function

$$E_5 = P(B)\,(\text{count} = f + 1).$$

The counter is set to zero at the beginning of every section.

The complete check function E is obtained by combining all error functions as

$$E = E_1 + E_2 + E_3 + E_4 + E_5.$$

Synchronization signal for ZM waveforms

The zero-modulated waveform at the receiver is decoded into a data sequence with the help of a clock, which is usually derived from the waveform. A synchronizing signal of sufficient length and recognizable ending is required for the purpose of starting and synchronizing the clock and marking the beginning of data. A synchronizing signal may also be inserted at predetermined intervals in the waveform (preferably at ZM memory boundaries) to provide "resyncability" in case of temporary loss of synchronization.

Following are some specifications for a synchronization signal in a ZM waveform.

1. It must be distinct such that it may not be confused with the normal data waveform in its original or shifted position.
2. It must satisfy the ZM constraints of maximum and minimum pulse widths.
3. Accumulated charge at the end must be zero (zero dc) although the maximum accumulated charge at any point may be more than three, say four, five, or six units. (Note that this specification can even be waived if the synchronization signal is short and infrequent.)
4. The basic synchronization signal should be reasonably short and the endings compatible with the ZM algorithm for insertion at the memory boundary without modification.

These specifications can be easily satisfied if a known sequence of binary data is inadmissible. This, however, is impossible for serial data. Alternatively, one can choose from the inadmissible sequences in ZM-coded patterns. In that case, the ZM error indicator must be modified to recognize the chosen inadmissible sequence as a synchronization pattern and to exclude the occurrence of that exact pattern from indicating an error in the data.

Theorem Among the sequences that satisfy the ZM run-length constraints, the sequence

$$w = 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0$$

and its reciprocal

$$w^* = 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0$$

are the shortest sequences that do not occur in any ZM pattern.

Proof Consider the sequence w in relation with the clock such that it forms seven pairs in the following manner:

	1	2	3	4	5	6	7
Pattern =	0 0	1 0	1 0	0 0	1 0	1 0	0 0
	a b	*a b*	*a b*	*a b*	*a b*	*a b*	*a b*
Decoded data =	0	0	1	1	0	1	1.

The charge constraint is violated at either position 3 or position 6. This can be checked by the error function E_3 of the previous section. Suppose it is not violated at position 3, i.e., $P(B) = 1$ at position 3. This means that $P(B) = 0$ at position 6 and, hence, is a violation according to function E_3.

Now consider the sequence w in a shifted position in relation with the clock in the following manner:

	1	2	3	4	5	6	7	
Pattern =	0	0 1	0 1	0 0	0 1	0 1	0 0	0
	b	*a b*	*a b*	*a b*	*a b*	*a b*	*a b*	*a*
Decoded data =	0	1	1	0	1	1	0	.

Again, the charge constraint is violated at position 3 or 6. This can be checked by the error function E_4 of the previous section. Suppose it is not violated at position 3. Then $P(B) = 1$, since $a_1b_1 = 00$ and $P(B1) = 0$ at position 3. However, this implies that $P(B) = 0$ at

position 6 and, hence, is a violation according to function E_4, since $a_1 b_1 = 00$ and $P(B1) = 0$ at position 6.

This proves that the sequence w cannot occur in any ZM pattern. The proof for w^* follows in the same manner. That w and w^* are the shortest such sequences can be proved by lengthy analytic arguments involving E_3 and E_4. Alternatively, an exhaustive check can be made on all 13-digit sequences that satisfy ZM run-length constraints. They are all valid ZM patterns.

Thus, any pattern containing the sequence w (or w^*) can be used as a synchronizing pattern. Following are two examples:

$$W_1 = 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1,$$
(w spans digits 6–18)

$$W_2 = 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1.$$
(w^* spans digits 6–18)

Note that W_1 and W_2 both contain w or w^*. However, specification 3 is satisfied by W_1 but not by W_2. The endings on both sides of W_1 and W_2 are 01 and can be padded by any number of 01 digit pairs if desired for clocking. These endings also allow placement of a synchronization signal at the ZM memory boundary without modification.

In actual application, the synchronization pattern is placed at predetermined intervals at ZM memory boundaries. In case of loss of synchronization, the clock generated by means of the read waveform as soon as the defect (or any other cause) has passed. This clock enables the signal detector to produce the binary pattern, i.e., the ZM pattern. The decoding of the pattern however, cannot be started until the synchronization pattern arrives and establishes the ZM pair relation with respect to the clock by means of the sequence w or w^*. If the clock is found to be out of synchronization (which is equivalent to a one digit shift), the complement of the clock may be used for decoding. The start of the data, then, occurs at the end of the synchronization signal from which, once again, the ZM pattern can be decoded into "good" data.

Directions for further work

The theory and results reported in the Appendix provide a formalized method of investigation for a more generalized theory of constrained sequences and waveform design. The method is applicable to any rate codes and to nonbinary codes as well. Here we summarize the main points of this theory.

1. Constrained sequences and data sequences of indefinite length can be represented by state diagrams having a finite number of states.
2. Isomorphism of state diagrams is an effective method of establishing uniqueness of mappings, particularly in case of nonlinear and/or convolutional codes in which the theory of linear algebra cannot be used.
3. The equality of growth rates of state transition matrices is a necessary condition for isomorphism.
4. The eigenvalues and, in particular, the spectral radius [8] of the state-transition matrix determines its growth rate.
5. The theory of non-negative matrices [9] is applicable in a constructive manner in modifying the state diagram to alter its growth rate by small amounts in the desired direction.
6. The state diagrams can be modified for equality of growth rates and isomorphism by eliminating, merging, or conditional splitting of various states or state-transitions and adjusting for the relative growth at each state by means of look-ahead or look-back conditions.
7. The relation between convolutional coding and block coding (as is evident in the case of ZM with limited memory) signifies an interesting inherent structure of constrained sequences.

In the Appendix we give growth rates of constrained sequences with other run-length constraints, namely $d = 2$ and $k = 7$ or $k = 8$. The codes with these parameters are potentially "good" codes.

Appendix: Growth rate of constrained sequences

• *ZM sequences*

Consider the growth, i.e., the number of constrained sequences as a function of their length, starting from a given initial state. Using the state diagram of Fig. 3, determine the sequences starting from the state Y after 1, 2, 3, etc. catenations of two-digit symbols, as shown in Table A1. This table shows that the total number of sequences approximately doubles at every step as the length increases. This process is now formalized mathematically.

Let V_t denote a column vector in which the elements $V_t(j)$ denote the number of sequences in the jth state after t catenations of two-digit symbols from initial state. The vector V_{t+1} can be obtained from V_t as

$$V_{t+1}(i) = \sum_i a_{ij} V_t(j),$$

where a_{ij} denotes the number of distinct transitions from state j to state i. The matrix A of elements a_{ij} is termed the transition matrix. Then $V_{t+1} = [A]V_t$. Using this equation iteratively, any state vector V_n can be computed as $V_n = [A]^n V_0$, where $[A]^i$ represents the ith power of matrix A, and V_0 is a vector with zeros in all positions except a single one in the position corresponding to the starting state. The total number of sequences of n two-digit symbols is then

$$N = \sum_i V_n(i).$$

The growth of N with n may be examined by studying the eigenvalues and eigenvectors of the matrix A. Consider the example of the ZM state diagram of Fig. 3. The states are arbitrarily ordered X, Y, Z, A, B, C, and D from 1 to 7. Then, matrix A can be written as

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

The eigenvalues of A can be obtained by solving the characteristic equation $|A - \lambda I| = 0$. The vector V_0 can be given as a linear combination of the basis vectors forming eigenspaces [10] corresponding to the eigenvalues. Table A2 gives these basis vectors corresponding to each eigenvalue and their relationship with A. If Y is the initial state, the initial state vector V_0 is [0, 1, 0, 0, 0, 0, 0]$^\tau$, which can be written in terms of the basis vectors of Table A2 as

$$V_0 = \tfrac{1}{12}\xi_1 - \tfrac{1}{3}\xi_3 + \tfrac{1}{4}\xi_5 - \tfrac{1}{2}\gamma_5 = [0\ 1\ 0\ 0\ 0\ 0\ 0]^\tau.$$

It is now easy to observe the growth of the vector V_0. We have $A\xi_1 = 2\xi_1$, $A\xi_3 = -\xi_3$, and $A\xi_5 = 0$, $A\gamma_5 = \xi_5$. Thus,

$$V_1 = AV_0 = 2(\tfrac{1}{12}\xi_1) + \tfrac{1}{3}\xi_3 + 0 - \tfrac{1}{2}\xi_5$$

and

$$V_n = A^n V_0 = 2^n(\tfrac{1}{12}\xi_1) - (-1)^n\tfrac{1}{3}\xi_3, \qquad \text{for } n \geq 2.$$

The elements of the vector V_n represent the number of sequences of n two-digit symbols in each state, as shown in Table A1. Using ξ_1, and ξ_3 of Table A2, we find

$$V_n = [P, 2^{n-2}, Q, 2^{n-2}, Q, P, P]^\tau,$$

where

$$P = (\tfrac{1}{3})\ [2^{n-1} - (-1)^n],$$

and

$$Q = (\tfrac{1}{3})\ [2^{n-1} + (-1)^n].$$

The total number of sequences N of n two-digit symbols is then

$$N = \sum_i V_n(i) = 2^n + (\tfrac{1}{3})\ [2^{n-2} - (-1)^n].$$

The value of N depends on the starting state. Table A3 lists values of N corresponding to each state as a starting state.

The growth rate GR of the constrained sequences may be defined as

$$GR = \lim_{n \to \infty} N^{(1/n)}.$$

The growth rate of ZM sequences, then, is equal to two. The logarithm of the growth rate is similar to the channel capacity [11]. In the limit, the largest positive eigenvalue of A dominates in determining the value of $N^{(1/n)}$; and, in general, the growth rate is equal to this eigenvalue of the matrix A. This growth rate is independent of the starting state, although the total number of sequences depends on the starting state.

In the next section, we apply some of these ideas to explore the growth rate of other run-length sequences.

• *Other run-length sequences with charge constraint*

All elements of the state transition matrix A for constrained sequences are non-negative, since they represent the number of transitions from one state to another. Such non-negative matrices possess a well defined growth rate given by their spectral radii [9]. The following theorem

Table A1 Number of sequences in various states starting from state Y.

Length in terms of two-digit symbols	X	Y	Z	A	B	C	D	*Total*
(Starting state →)	0	1	0	0	0	0	0	
1	0	1	0	1	0	0	0	2
2	0	1	1	1	1	0	0	4
3	1	2	1	2	1	1	1	8 + 1
4	1	4	3	4	3	1	1	16 + 1
5	3	8	5	8	5	3	3	32 + 3
6	5	16	11	16	11	5	5	64 + 5
7	11	32	21	32	21	11	11	128 + 11
8	21	64	43	64	43	21	21	256 + 21
9	43	128	85	128	85	43	43	512 + 43
10	85	256	171	256	171	85	85	1024 + 85
⋮								
n	P	2^{n-2}	Q	2^{n-2}	Q	P	P	$2^n + P$

$P = \frac{1}{3}[2^{n-2} - (-1)^n]$ and $Q = \frac{1}{3}[2^{n-1} + (-1)^n]$.

from the Perron-Frobenius theory [9, 10] of non-negative matrices is stated here without proof. These results are immediately useful in developing new codes and, in general, any nonlinear mappings by means of the method of isomorphic state diagrams.

Theorem 1 Let A be a non-negative square matrix. Then A has a non-negative real eigenvalue equal to its spectral radius $\rho(A)$. To $\rho(A)$ there corresponds a non-negative eigenvector.

Theorem 2 Let A and B be two square, non-negative matrices such that each element of matrix B is smaller than or equal to the corresponding element of matrix A. Then $\rho(B) \leq \rho(A)$.

The growth rate of the charge-constrained sequences is the spectral radius $\rho(A)$ of the state transition matrix A. This is the largest real eigenvalue of the matrix A. A necessary condition for the existence of the code is that the growth rate of the sequences is at least that of the binary data sequences. Thus, $\rho(A) < 2$ implies that a mapping is not possible. However, $\rho(A) > 2$ indicates that the state diagram must be modified for a spectral radius of two before the isomorphic state diagram of data sequences can be constructed. This may be done by eliminating some of the transitions or states which decrease the value of some elements of A. The new transition matrix, according to Theorem 2, may have a smaller but not larger spectral radius than that of the original matrix A.

The spectral radius $\rho(A)$ of the transition matrix A can be computed using iterative analysis [9]. Many dif-

Table A2 Eigenvalues and eigenspaces of matrix A.

Eigenvalues λ	*Basis vectors for the corresponding eigenspace*	*Relationship with A*
2	$\xi_1 = [1, 3, 2, 3, 2, 1, 1]^T$	$[A - 2I]\xi_1 = 0$
1	$\xi_2 = [1, 0, 1, 0, -1, 1, -1]^T$	$[A - I]\xi_2 = 0$
−1	$\xi_3 = [1, 0, -1, 0, -1, 1, 1,]^T$	$[A + I]\xi_3 = 0$
	$\gamma_3 = [1, -2, 0, -2, 3, 1, -2]^T$	$[A + I]\gamma_3 = \xi_3$
0	$\xi_4 = [1, 0, 0, -1, 0, 0, 1]^T$	$A\xi_4 = 0$
	$\xi_5 = [1, -1, 0, -1, 0, 1, 1,]^T$	$A\xi_5 = 0$
	$\gamma_5 = [0, -2, 1, 0, 1, 0, 0]^T$	$A\gamma_5 = \xi_5$

Table A3 Value of N from various starting states in ZM sequences.

Starting state	*Total number N of sequences of n two-digit symbols*
X	$\frac{3}{16}2^n - (-1)^n\left(\frac{6n-23}{36}\right) + \frac{1}{4}$
Y	$\frac{39}{36}2^n - (-1)^n\frac{1}{3}$
Z	$\frac{26}{36}2^n + (-1)^n\left(\frac{6n-17}{36}\right) + \frac{1}{4}$
A	$\frac{39}{36}2^n + (-1)^n\frac{2}{3}$
B	$\frac{52}{36}2^n - (-1)^n\left(\frac{6n+7}{36}\right) - \frac{1}{4}$
C	$\frac{39}{36}2^n - (-1)^n\frac{1}{3}$
D	$\frac{26}{36}2^n + (-1)^n\left(\frac{6n+1}{36}\right) - \frac{1}{4}$

Table A4 Growth rate of charge-constrained sequences.

Run length $min = d$	$max = k$	*Max accumulated charge* $\pm c$	*States eliminated for reduction in rate (charge, end-zeros)*	*Growth rate*
1	3	3		2
1	4	3		2.1112
1	4	3	(2, 0) and (2, 4)	2
2	7	7		1.9879
2	7	8		2.0029
2	7	8	(0, 7)	2.0003
2	7	8	(8, 7)	2.0003
2	7	8	(8, 7) and (0, 7)	1.9981
2	8	6		1.9820
2	8	7		2.0099
2	8	7	(2, 8) and (6, 1)	2.0037
2	8	7	(2, 8) and (6, 8)	2.0032
2	8	7	(6, 8) and (6, 1)	2.0024
2	8	7	(6, 8), (6, 1) and (2, 8)	1.9998
2	9	6		1.9903
2	9	7		2.0214

ferent transition matrices were studied for other possible 0.5-rate codes. Among them, the most interesting and of immediate importance to magnetic recording are those with a minimum run-length of two, in particular, the (2, 7) and (2, 8) run-length-limited sequences. These codes can provide a higher density ratio of data to magnetic transitions.

The (2, 7) run-length sequences with various charge constraints were examined to determine their growth rate using two-digit symbols for a rate one-half code. The (2, 8) and (2, 9) run-length sequences were also examined in a similar manner. Table A4 presents the results of this study. The conclusion is that any binary rate one-half mapping into (2, 7) run-length sequences will have charge accumulation of at least eight units in either direction. Similarly, any binary rate one-half mapping into (2, 8) or (2, 9) run-length sequences will have charge accumulation of at least seven units in either direction. Table A4 also shows that the growth rate can be adjusted by elimination of some of the state transitions or states. This is a trial-and-error effort to achieve a growth rate of two, which may require an exhaustive search. There are other ways to modify the state diagrams and the state transition matrix, in particular, by means of conditional splitting and merging of some of the states. Such investigations may reveal more general structural properties of constrained sequences and related band-limited signaling waveforms.

References

1. M. Hecht and A. Guida, "Delay Modulation," *Proc. IEEE* **57,** 1314 (1969).
2. P. A. Franaszek, "Sequence-State Methods of Run-Length-Limited Coding," *IBM J. Res. Develop.* **14,** 376 (1970).
3. D. T. Tang and L. Bahl, "Block Codes for a Class of Constrained Noiseless Channels," *Information and Control* **17,** 436 (1970).
4. A. Gabor, "Adaptive Coding for Self-Clocking Recording," *IEEE Trans. Electronic Computers* **EC-16,** 866 (1967).
5. C. V. Freiman and A. D. Wyner, "Optimum Block Codes for Noiseless Input Restricted Channels," *Information and Control* **7,** 398 (1964).
6. P. A. Franaszek, "Sequence State Coding for Digital Transmission," *Bell System Tech. J.* **47,** 143 (1968).
7. A. Croisier, "Introduction to Pseudoternary Transmission Codes," *IBM J. Res. Develop.* **14,** 354 (1970).
8. R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1962.
9. G. Frobenius, "Über Matizen aus Nicht Negativen Elementen," S.-B. Preuss Akad. Wiss., Berlin, 1912, pp. 456-477.
10. E. D. Nering, *Linear Algebra and Matrix Theory*, John Wiley & Sons, Inc., New York, 1963.
11. C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Tech. J.* **27,** 379 (1948).

Received March 15, 1974; revised March 10, 1975

The author is located at the IBM General Products Division laboratory, Monterey and Cottle Roads, San Jose, California 95193.