P. A. Franaszek

Sequence-state Methods for Run-length-limited Coding

Abstract: Methods are presented for the encoding of information into binary sequences in which the number of ZEROS occurring between each pair of successive ONES has both an upper and a lower bound. The techniques, based on the state structure of the constraints, permit the construction of short, efficient codes with favorable error-propagation-limiting properties.

1. Introduction

This paper presents a study of a class of codes that are of interest in connection with a number of digital recording and communication techniques. The codes are such that coder output sequences are binary, and have the property that two consecutive ONES are separated by at least d but no more than k ZEROS. The parameter d may be used to control interference between recorded transitions in saturation recording, or to limit spectrum spread in frequency-shift keying. The parameter k imposes a bound on the maximum transition spacing, a dimension that must be specified in most systems that employ self-clocking.

Problems connected with such run-length-limited codes have received considerable attention.[1-6] Application of the codes, particularly to magnetic recording, was discussed by Melas,[1] Kautz,[2] Tang[3] and Gabor.[4] Properties of the output sequences were studied by Melas.[1] Asymptotically optimal coding techniques whose complexity grows linearly with the code word length were described by Kautz[2] and Tang.[3] In addition, a number of short codes and state-oriented coding techniques were presented by Freiman and Wyner,[6] Gabor[4] and Tang.[5]

The approach taken in this paper is based on the use of finite-state machines as models of the run-length-limited sequence constraints. The analysis is thus applicable to any constraints that can be described in this form. Algorithms presented in a recent report[7] are used to construct synchronous (fixed-rate) codes that are optimal in the sense that the maximum word length is minimized for a given bit-per-symbol value. Word lengths of fixed- and of variable-length codes in this class are compiled for a number of (d, k) constraints. The results indicate that varying the word length frequently yields codes that are shorter and easier to implement. The problem of error propagation, which arises in state-dependent and variable-length coding,[8] is studied. It is shown that one can always limit error propagation in fixed-length (d, k) codes by a proper assignment of message digits to code words. A method for constructing error-propagation-limiting, variable-length (d, k) codes is described, the method being valid for the more general case of constrained sequences with finite memory. Finally, some examples of code construction are included to illustrate the methods.

2. Run-length-limited sequence constraints

In some applications it is desirable to impose minimum and maximum distances between transitions or pulses in a signal. Digital magnetic recording is an example. The recording medium is typically partitioned into intervals of length T. Information is stored in these intervals by means of the presence or absence of transitions between saturation levels. Intervals in which transitions occur are assigned the value ONE. If no transition occurs, the interval is assigned the value ZERO. The separation between such transitions must be sufficiently great to limit interference to acceptable levels. If clocking is to be derived from the recorded data, an additional requirement is that transitions occur frequently enough to provide adequate energy for the timing circuits.

Suppose that Q and V are respectively the minimum and maximum tolerable distances between transitions. A possible approach to the signal design problem is to require that the recorded sequence be (d, k) *run-length-limited* with $(d + 1)T \geq Q$ and $(k + 1)T \leq V$. That is, at least $(d + 1)$ but no more than $(k + 1)$ intervals occur between each pair of successive transitions. If Q and V are fixed, an increase in $(d + 1)$ (i.e., a decrease in T)

The author is located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York.

will produce an increase in the amount of information that can be stored, and a decrease in the level of tolerable degradation. Thus, the choice of d and k is a function of the interference and of the circuit quality in a given system.

The (d, k) sequence constraints may be represented by a finite-state sequential machine. Figure 1 illustrates a possible state-transition diagram. There are $(k + 1)$ states, $(\sigma_1, \sigma_2, \cdots, \sigma_{k+1})$. Transmission of a ZERO takes the sequence from state σ_i to state σ_{i+1}. Transmission of a ONE takes the sequence to σ_1. That is, state σ_r, $r = 1, 2, \cdots, k + 1$, indicates that $(r - 1)$ ZEROS have occurred since the last ONE. A ONE may be transmitted only when the sequence occupies states $\sigma_{d+1}, \cdots, \sigma_{k+1}$. If the sequence occupies state σ_{k+1}, then only a ONE may be transmitted.

The skeleton transition matrix, which gives the number of ways of going from state σ_i to state σ_j, is given by the $(k + 1) \times (k + 1)$ array:

$$\mathbf{D} = \{d_{ij}\}, \tag{1}$$

where

$$d_{i1} = 1, \quad i \geq d + 1; \tag{2}$$

$$d_{ij} = 1, \quad j = i + 1; \tag{3}$$

$$d_{ij} = 0, \quad \text{otherwise.} \tag{4}$$

For example, the $\mathbf{D}$ matrix for $(d, k) = (2, 4)$ is

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{5}$$

The above representation is related to the input-restricted channels first studied by Shannon.[9] The finite-state machine model permits the computation of the channel capacity, defined as the number of bits per symbol that may be carried by the sequence. This quantity is given[9] by the base-two logarithm of the largest real root of

$$\det [d_{ij} Z^{-1} - \delta_{ij}] = 0. \tag{6}$$

For large values of k it may be more convenient to obtain the channel capacity from[8]

$$C \approx \frac{1}{n} \log_2 \sum_{ij} d_{ij}^n, \tag{7}$$

where

$$d_{ij}^n \equiv [\mathbf{D}^n]_{ij}. \tag{8}$$

If the value of n is a power of two, then $\mathbf{D}^n$ may be obtained from $\mathbf{D}$ by a total of $\log_2 n$ matrix multiplications.

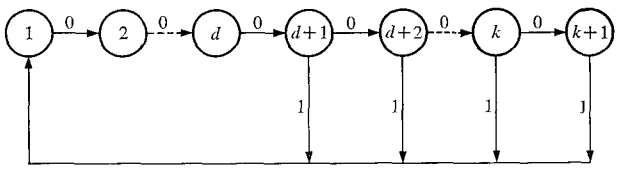

Figure 1 State-transition diagram for a (d, k) sequence.

Table 1: Channel capacities for a selection of (d, k) constraints.

d	k	*Channel capacity in bits/symbol*
0	1	0.694
0	2	0.879
0	3	0.947
1	3	0.552
1	4	0.617
1	5	0.651
1	7	0.679
2	5	0.465
2	8	0.529
2	11	0.545
3	7	0.406
3	11	0.452
3	15	0.462
4	9	0.362
4	14	0.397
5	12	0.337
5	17	0.356
6	13	0.301
6	17	0.318
6	20	0.324
7	15	0.279
7	23	0.298
8	17	0.260
8	26	0.276

The required value of n may be estimated from the following bound for the capacity[7]:

$$\frac{1}{n + k} [\log_2 \sum_{ij} d_{ij}^n - 2 \log_2 (k + 1)] \leq C \leq \frac{1}{n} \log_2 \sum_{ij} d_{ij}^n. \tag{9}$$

Values of C for a number of (d, k) constraints were obtained by computer calculation. These values are given in Table 1.

A property of the above sequence constraints is that the channel is of finite memory. That is, the channel state may be uniquely determined from a finite number of previously transmitted symbols. More precisely, if the channel occupies state σ_r, $r = 1, 2, \cdots, k + 1$, (we use the numbering of Fig. 1), then the channel state may

Table 2: Shortest fixed-length codes of given bit-per-symbol values for a selection of (d, k) constraints.

d	k	α	N	E
0	1	3	5	0.865
0	2	4	5	0.910
0	3	9	10	0.950
1	3	1	2	0.906
1	5	6	10	0.922
1	7	6	10	0.884
2	5	4	10	0.860
2	8	11	22	0.945
2	11	8	16	0.917
3	7	46	115	0.985
3	11	8	20	0.885
3	15	8	20	0.866
4	9	9	27	0.921
4	14	12	33	0.917
5	12	9	30	0.890
5	17	15	45	0.843
6	13	12	44	0.906
6	17	9	33	0.858
6	20	15	50	0.926
7	15	9	36	0.896
7	23	7	28	0.839
8	17	27	108	0.962
8	26	12	48	0.906

α = Number of bits per word
N = Word length
E = $[\alpha/N]/C$, the code efficiency

be identified through knowledge of the r previously transmitted symbols. This property is used to limit error propagation in variable-length codes.

3. Fixed-length binary (d, k) codes

• *Discussion*

Suppose one wishes to map binary information onto a (d, k) sequence with a code of fixed length. The message sequence is partitioned into blocks of length α, and such blocks mapped by the code onto words composed of N channel symbols. A code may be state dependent, in which case the choice of the word used to represent a given binary block is a function of the channel state, or the code may be state independent. State independence implies that code words can be freely concatenated without violating the sequence constraints.[6] It is an additional restriction that, in general, leads to codes that are longer than state-dependent codes for a given bit-per-symbol value. In some cases state independence may yield advantages in error propagation limitation, since channel words may be decoded without knowledge of the state. However, state-independent decoding may be achieved for any fixed-length (d, k) code, as is indicated by the theorem below. Since coder and decoder complexity tends to increase exponentially with the code word length, it is usually advantageous to search for the shortest existing code without regard to state independence, and to achieve state-independent decodability by a proper assignment of message digits to code words.

Let S denote the states of a finite-state machine that represents a (d, k) sequence. Let $W(\sigma_i)$ denote the set of words that may be transmitted when the sequence occupies the initial state σ_i.

Theorem 1: Given a set of states $S' = \{\sigma_i\} \subset S$ and a class of associated word sets $\{W(\sigma_i)\}$ such that each $W(\sigma_i)$ contains at least 2^α words, it is possible to assign binary blocks b_r, $r = 1, 2, \cdots, 2^\alpha$, to the word sets $\{W(\sigma_i)\}$ so that there is a unique inverse mapping.

Proof: A word w that is allowable from at least one state (i.e., it may be obtained from that state by an appropriate path through the state transition diagram) is not allowable from another state either because there is a ONE too close to the beginning of the word, or because there are too many ZEROS before the first ONE. Suppose w is allowable from state σ_n and σ_{n+m}, where $m > 0$. Then the allowability from σ_n implies that the first ONE is not too close to the beginning for w to be allowable from $\sigma_{n+1}, \sigma_{n+2}, \cdots, \sigma_{n+m-1}$. The allowability of w from σ_{n+m} implies that it does not have too many ZEROS in the prefix to be allowable from $\sigma_{n+1}, \sigma_{n+2}, \cdots, \sigma_{n+m-1}$. Thus,

$$w \in W(\sigma_n) \cap W(\sigma_{n+m}) \Rightarrow w \in W(\sigma_{n+j}),$$
$$j = 1, 2, \cdots, m - 1, \tag{10}$$

which is a sufficient condition[8] for the theorem to hold.

• *Fixed-length codes of minimum length*

A quick method for determining whether there exists a code with word length N and α bits per word is given in Ref. 7. It is a recursive search technique for determining the existence of a set of *principal states* through operations on the **D** matrix. These are the states from each of which there exists a sufficient number, 2^α, of paths terminating at other principal states. The existence of a set of principal states is a necessary and sufficient condition for the existence of a code with the given values of α and N, and in the case of (d, k) sequences, is a necessary and sufficient condition for the existence of a state-independently decodable code. A number of (d, k) combinations and the length of the shortest existing fixed-length codes for a given ratio of α/N were computed. These are listed in Table 2, which may be used as an aid in construction of both state-dependent and state-independent codes. In the latter case the Table indicates a lower bound on the code-word length for a given bit-per-symbol value. The systematic construction of a state-dependent code is illustrated next.

• *Code construction*

The words available for encoding are the paths of length N connecting the principal states. These may be obtained from the Nth power of the $(k+1) \times (k+1)$ channel-transition matrix

$$\mathbf{A} = \{a_{ij}\}, \tag{11}$$

where a_{ij} represents the sequence digit corresponding to a transition from state σ_i to σ_j. If it is not possible to go from σ_i to σ_j with one digit (i.e., if $d_{ij} = 0$), then $a_{ij} = \emptyset$, where $\emptyset$ denotes the null symbol. Powers of $\mathbf{A}$ are formed by the operations of disjunction, $+$, and concatenation. The concatenation of the null symbol $\emptyset$ with any symbol results in $\emptyset$.

As an example, consider the transition matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 1 & \emptyset \end{bmatrix}, \tag{12}$$

which corresponds to $(d, k) = (0, 1)$.

The outputs of length two are given by

$$\mathbf{A}^2 = \begin{bmatrix} 11 + 01 & 10 \\ 11 & 10 \end{bmatrix}. \tag{13}$$

Code words in $W(\sigma_i)$ may be obtained from $\sum_j [\mathbf{A}^N]_{ij}$, where the operation, $+$, is taken over the principal states.

A state-independently decodable code may be obtained as follows[8]:

1) List the principal states $\sigma_{m_1}, \sigma_{m_2}, \cdots, \sigma_{m_q}$, where $m_1 < m_2 < \cdots < m_q$.
2) Assign a binary word of length α to each of the 2^α words from $W(\sigma_{m_1})$.
3) If $w_x \in W(\sigma_{m_i}) \cap W(\sigma_{m_j})$, where $i < j$, assign the same binary word to w_x for coding from σ_{m_i} as from σ_{m_j}. If $w_x \in W(\sigma_{m_i})$ but $w_x \notin W(\sigma_{m_j})$, assign the binary word to a $w_y \in W(\sigma_{m_j})$ such that $w_y \notin W(\sigma_{m_i})$.

Theorem 1 assures the success of the procedure.

• *Example*

Consider $(d, k) = (1, 3)$. $\mathbf{D}$ and $\mathbf{D}^2$ are given by

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \tag{14}$$

$$\mathbf{D}^2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{15}$$

Table 2 shows that a bit-per-symbol value of $\frac{1}{2}$ represents over 90 percent of the channel capacity.

Use of the search algorithm described in the subsection on fixed-length codes of minimum length indicates that the shortest existing code with this bit-per-symbol value has the parameters $\alpha = 1$, $N = 2$. The principal states are $\sigma_1, \sigma_2, \sigma_3$. Code words may be obtained from

$$\mathbf{A}^2 = \begin{bmatrix} 01 & \emptyset & 00 & \emptyset \\ 01 & 10 & \emptyset & 00 \\ 01 & 10 & \emptyset & \emptyset \\ \emptyset & 10 & \emptyset & \emptyset \end{bmatrix}. \tag{16}$$

The alphabets available for encoding are the word sets associated with the principal states:

$$W(\sigma_1) = \begin{cases} 01 \\ 00, \end{cases} \tag{17}$$

$$W(\sigma_2) = \begin{cases} 01 \\ 10, \end{cases} \tag{18}$$

$$W(\sigma_3) = \begin{cases} 01 \\ 10. \end{cases} \tag{19}$$

A state-independently decodable code may be constructed with the following binary assignments:

Binary symbols	*State*	*Channel code words*
1	σ_1	01
	σ_2	01
	σ_3	01
0	σ_1	00
	σ_2	10
	σ_3	10

This code is well known and is usually referred to as MFM (modified frequency modulation). It is of interest to note that it results almost automatically from the application of the above techniques to the $(d, k) = (1, 3)$ sequence.

4. Variable-length synchronous binary codes

• *General*

The codes discussed in the previous section have the property that the code may be a function of the state occupied by the sequence. This permits a minimization of word length over the class of fixed-length codes with a given bit-per-symbol value. In this section, an additional degree of freedom is introduced by allowing the word length to vary. The result is often a significant decrease in the required code word length. The rate of information

Table 3: Shortest variable-length codes of given bit-per-symbol values for a selection of (d, k) constraints.

d	k	α	N	M	E
2	8	1	2	4	0.945
2	11	1	2	4	0.917
3	7	2	5	8	0.985
3	11	2	5	3	0.885
3	15	2	5	3	0.866
4	9	1	3	3	0.921
4	14	4	11	3	0.917
5	12	3	10	3	0.890
5	17	1	3	6	0.843
6	13	3	11	3	0.906
6	17	3	11	2	0.858
6	20	3	10	4	0.926
7	15	1	4	3	0.896
7	23	1	4	3	0.839
8	17	1	4	8	0.962
8	26	1	4	5	0.906

α = number of bits per word
N = word length
NM = maximum word length

transmission is kept constant by fixing the bit-per-symbol value for each word. That is, words of length N carry half as many bits as those of length $2N$. Algorithms described in a recent report[7] are used to construct variable-length fixed-rate (synchronous) codes that are optimal in the sense that the maximum word length is minimized. Procedures are developed to find codes with error-propagation-limiting properties.

• *Variable-length codes of minimum length*

A recursive search and optimization method to determine whether there exists a variable-length code with a given bit-per-symbol value and maximum word length is described in Ref. 7. To apply the procedure, a *basic channel word length* N and number α of bits per N channel symbols are chosen, along with a maximal word length MN. Words may be of length jN, $j = 1, 2, \cdots, M$. The procedure involves operations on powers of the **D** matrix. If the search is successful, the results are a set of states from which coding may be performed (termed the *principal states*) and, for each such state σ_i, a set of paths that maximizes the quantity

$$\Psi(\sigma_i) = L_1(\sigma_i) + 2^{-\alpha}L_2(\sigma_i) + \cdots + 2^{-\alpha(M-1)}L_M(\sigma_i), \quad (20)$$

where $L_j(\sigma_i)$ is the number of available code words of length jN. It is required that $\Psi(\sigma_i) \geq 2^{\alpha}$ for a code to exist. Terminal states are specified for the paths. These terminal states are in turn principal states, so that channel words always lead to states from which coding is possible.

The shortest maximum word lengths for a number of (d, k) constraints and bit-per-symbol values were computed with the above technique and are shown in Table 3. A comparison of the variable-length codes with the fixed-length codes of Table 2 shows that the extra degree of freedom in variable-length coding often yields a very significant decrease in word length. For example, the $(d, k) = (4, 9)$ constraints, with a bit-per-symbol value of $\frac{1}{3}$, result in fixed-length codes with a minimum of 9 bits per word. Thus, at least $2^9 = 512$ words are required for fixed-length coding, while a variable-length code (illustrated in an example given later in this section) exists with a maximum word length corresponding to 3 bits, and which has a total of 6 words. Since coding is done by table look-up, the advantage of the variable-length code is clear. Further benefits resulting from the shorter word length are discussed below.

• *Error propagation limitation*

Consider the fixed-length codes of Section 3. It was shown that if a fixed-length code exists, then it is possible to code so that decoding can be performed without a knowledge of the state occupied by the channel at the beginning of the transmitted word. This property implies that an error in the detection of a symbol in a given word does not affect the decoding of the next word. However, the property is not sufficient to limit error propagation in codes of variable word length since an error in detection may cause the decoder to treat the received symbol as part of a word of length different from that which was transmitted, possibly resulting in serious error propagation. Thus, error propagation in variable-length codes may result from improper blocking of the received sequence into words (misframing), as well as from code state dependency.

Suppose it were possible to mark word endings with, for example, a special sequence of symbols. Misframing as a result of an error in detection could then be eliminated after the correct reception of at most one word. For channels with finite memory [of which the (d, k) constraints are an example] it is possible to achieve essentially this result by an appropriate choice of code paths, as shown below. Since the method depends in part on identification of the sequence state, it overcomes error propagation due to state dependence as well as to misframing.

Let S_e be the subset of the principal states in which code words actually terminate. Consider a code such that code words are *required* to terminate when a state that is a member of S_e is entered after rN symbols, where r is an integer. Since words may end only after an integer multiple of N symbols, this restriction is sufficient to permit the decoder to determine word endings by tracking

the state of the sequence. Knowledge of the state also resolves ambiguities resulting from code-state dependence.

Let L be the maximum number of channel symbols that are required to identify such a state. In general, L is less than the channel memory, $\mathfrak{M}$. The decoder may identify the first word termination occurring after at most L correctly received symbols. Note that this is equivalent to marking word endings with any one of a number of special sequences, some of which may be longer than code words. An advantage of this approach over that of using a single sequence to mark word terminations is that it may result in a larger number of available code words of a given length, and thus in a shorter code for a given bit-per-symbol value.

• *A search routine for a class of error-propagation-limiting codes*

In order to implement the error-propagation-limiting (EPL) procedure discussed in the previous section, it is necessary to find a set of states S_e (termed an EPL set) and paths connecting them that have the following properties:

1) The set S_e is a subset of S_p, the set of principal states. This is due to the necessity of starting words from principal states.

2) A word that enters a state $\sigma_i \in S_e$ after rN symbols, $r = 1, 2, \cdots, M$, terminates there. All words terminate in states that are members of S_e.

3) Associated with each $\sigma_i \in S_e$ there is a set of words of length rN, $r = 1, 2, \cdots, M$, such that

$$\Psi(\sigma_i) = L_1(\sigma_i) + 2^{-\alpha}L_2(\sigma_i) + \cdots + 2^{-\alpha(M-1)}L_M(\sigma_i) \geq 2^{\alpha}. \quad (21)$$

It follows from these properties that there may be instances when one or more states must be eliminated from S_p in order to form a set S_e, if it exists.

Let V_p^r be an r-tuple of states that may be elided from S_p while the possibility of encoding is maintained. That is, from each state $\sigma_i \in (S_p - V_p^r)$ there is available a sufficient number of words terminating in states belonging to $(S_p - V_p^r)$ to satisfy Eq. (21). Such subsets of the principal state set are termed *principal subsets*. Let Ω^r be the set of V_p^r r-tuples.

The set Ω^r can be found by using the optimization procedure of Ref. 7 to determine whether $(S_p - \sigma_i)$ is a principal subset of states for each $\sigma_i \in S_p$. The dimensionality of the search for all principal subsets of S_p may be reduced by noting that if an r-tuple of states is a member of Ω^r, then all combinations of $(r-1)$ members of the r-tuple are members of Ω^{r-1}.

Each principal subset S_p^a may be tested to determine whether it forms an EPL set. The subset S_p^a is an EPL set if, for each $\sigma_i \in S_p^a$, $\Psi_e(\sigma_i) \geq 2^{\alpha}$.

$$\Psi_e(\sigma_i) = \sum_{\sigma_k \in \mathcal{S}_1} d_{ik}^N + 2^{-\alpha} \sum_{\sigma_e \notin \mathcal{S}_1} \sum_{\sigma_m \in \mathcal{S}_2} d_{ie}^N d_{em}^N + \cdots + 2^{-\alpha(M-1)} \sum_{\sigma_e \notin \mathcal{S}_1} \cdots \sum_{\sigma_q \notin \mathcal{S}_{M-1}} \sum_{\sigma_r \in \mathcal{S}_M} d_{ie}^N \cdots d_{qr}^N, \quad (22)$$

where the $\mathcal{S}_r \equiv S_p^a[rN, \sigma_i]$ are those members of S_p^a that are reachable from σ_i with rN symbols.

The available code words can be obtained from the channel-transition matrix **A**. Let $W_r(\sigma_i)$ be the words of length rN available from state σ_i. Using the operations defined in Section 3 we construct the words

$$W_1(\sigma_i) = \sum_{\sigma_q \in S_p{}^a} a_{iq}^N$$

$$W_2(\sigma_i) = \sum_{\sigma_q \notin S_p{}^a} \sum_{\sigma_l \in S_p{}^a} a_{iq}^N a_{ql}^N$$

$$\vdots$$

$$W_{MN}(\sigma_i) = \sum_{\sigma_q \notin S_p{}^a} \sum_{\sigma_l \notin S_p{}^a} \cdots \sum_{\sigma_n \in S_p{}^a} a_{iq}^N a_{ql}^N \cdots a_{mn}^N. \quad (23)$$

Examples of codes for $(d, k) = (4, 9)$ and $(d, k) = (7, 15)$ are given below.

• *Example: a variable-length code for* $(d, k) = (4, 9)$

Consider the sequence with $(d, k) = (4, 9)$. There are ten states: $\sigma_1, \sigma_2, \cdots, \sigma_{10}$. Using the procedure discussed in the second subsection of Section 4 a set of principal states can be found for coding with $\alpha = 1$, $N = 3$ and $M = 3$. These are σ_4, σ_5, σ_6 and σ_7. In this case, S_p is an EPL set. Moreover, $\{\sigma_4, \sigma_5, \sigma_6, \sigma_7\}$ is the only principal subset of states; elimination of one state from this subset eliminates the possibility of coding with the above α, N and M parameters. Channel words corresponding to the EPL paths for the principal states, obtained from Eq. (23), are shown in Fig. 2.

Table 4 shows an assignment of binary digits to the channel words such that decoding may be performed without recourse to state information as long as there are no errors.

Words that enter principal states after 3, 6 or 9 symbols terminate there. Word endings may be found through state identification. Since n symbols are required to identify σ_n, the correct reception of any word except 000 is sufficient to determine word termination. In the latter case, this can be done after reception of the next word, so that at worst, code words corresponding to four bits must be correctly received in order to identify word termination after an error in detection.

Table 2 shows that fixed-length coding with the same bit-per-symbol value as in the current example would require channel words of minimum length 27, each representing 9 bits. Here table look-up coding and decoding

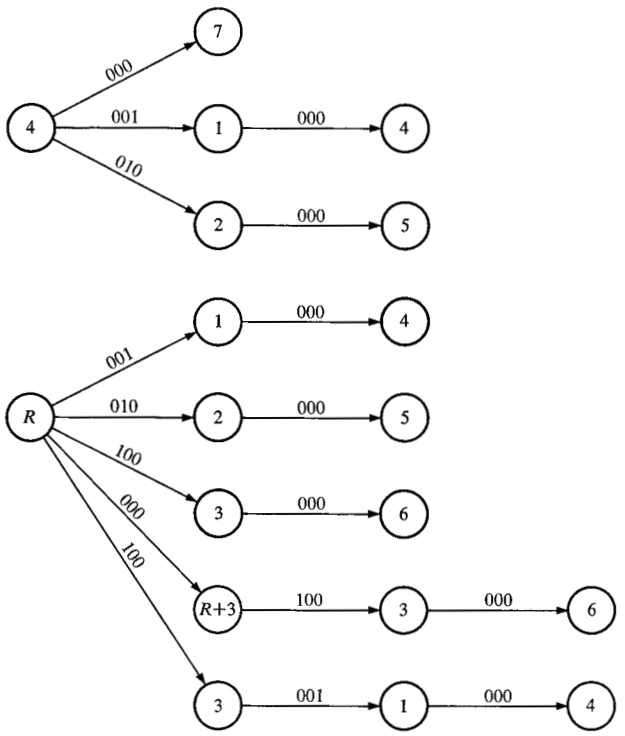

Figure 2 EPL coding paths for $(d, k) = (4, 9)$, $\alpha = 1$, $N = 3$ and $M = 3$. The principal states are σ_4, σ_5, σ_6, σ_7; $R = 5$, 6 and 7 for the paths corresponding to σ_5, σ_6 and σ_7, respectively.

might not be practical. Moreover, a single error in detection could result in the incorrect decoding of nine bits.

• *Example: Variable-length code for* $(d, k) = (7, 15)$

The $(d, k) = (7, 15)$ channel has sixteen states. A variable-length code with $\alpha = 1$, $N = 4$ and $M = 3$ exists, as can be seen in Table 3. The principal states for such encoding are $\sigma_5, \sigma_6, \cdots, \sigma_{13}$. These do not form an EPL set. However, there exist principal subsets that have the EPL property. The smallest such set is $\{\sigma_5, \sigma_6, \sigma_7, \sigma_8, \sigma_{10}, \sigma_{11}, \sigma_{13}\}$. A code with these as terminal states can be formed as shown in Table 5.

Word termination may be detected by channel state identification. That is, words terminate when a state in the principal subset is entered after rN symbols, where r is an integer. Once the detected sequence has been partitioned into words, decoding may proceed without further recourse to state information. Misframing as a result of an error in detection is corrected after the detection of at most 16 channel symbols, which correspond to 4 bits.

5. Conclusions

Methods have been described for the systematic construction of run-length-limited codes. The techniques are based on modeling the sequence constraints by finite-state machines, and are thus applicable to any constraints that can be described in this form. It was shown that varying the code word length often yields codes with superior error-propagation-limiting properties that are simpler and easier to implement than the corresponding codes of fixed word length.

Table 4 A $(d, k) = (4, 9)$ code for $\alpha = 1$, $N = 3$ and $M = 3$.

State	*Binary symbols*	*Channel code words*
σ_4	0	000
	10	001 000
	11	010 000
$\sigma_5, \sigma_6, \sigma_7$	10	001 000
	11	010 000
	01	100 000
	001	000 100 000
	000	100 001 000

Table 5 A $(d, k) = (7, 15)$ code for $\alpha = 1$, $N = 4$ and $M = 3$.

State	*Binary symbols*	*Channel code words*
σ_5	00	0001 0000
	010	0000 0100 0000
	011	0000 1000 0000
	11	0000 0000
	100	0000 0001 0000
	101	0000 0010 0000
σ_6, σ_7	1	0000
	00	0001 0000
	01	0010 0000
$\sigma_8, \sigma_{10}, \sigma_{11}, \sigma_{13}$	00	0001 0000
	01	0010 0000
	10	0100 0000
	11	1000 0000

Acknowledgment

The programming support of G. Krajcsik is gratefully acknowledged.

References

1. C. M. Melas, "Quelques Proprietes des Codes Binaires avec Symbols Consecutifs," IEEE International Symposium on Information Theory, 1967.
2. W. H. Kautz, "Fibonacci Codes for Synchronization Control," *IEEE Trans. Info. Theory* **IT-11,** 284 (1965).

3. D. T. Tang, "Run Length Limited Codes," IEEE International Symposium on Information Theory, 1969.
4. A. Gabor, "Adaptive Coding for Self-clocking Recording," *IEEE Trans. Elect. Computers* **EC-16,** 866 (1967).
5. D. T. Tang, "Practical Coding Schemes with Run-length Constraints," IBM Research Report RC-2022, Thomas J. Watson Research Center, Yorktown Heights, New York, 1968.
6. C. V. Freiman and A. D. Wyner, "Optimum Block Codes for Noiseless Input Restricted Channels," *Info. and Control* **7,** 398 (1964).
7. P. A. Franaszek, "On Synchronous Variable Length Coding for Discrete Noiseless Channels," *Info. and Control* **15,** 155 (1969).
8. P. A. Franaszek, "Sequence-state Coding for Digital Transmission," *Bell System Tech. J.* **47,** 143 (1968).
9. C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Tech J.* **27,** 379 (1948).

Received October 9, 1969