P. A. Franaszek

Construction of Bounded Delay Codes for Discrete Noiseless Channels

Algorithms are described for constructing synchronous (fixed rate) codes for discrete noiseless channels where the constraints can be modeled by finite state machines. The methods yield two classes of codes with minimum delay or look-ahead.

1. Introduction

Consider the problem of fixed rate (synchronous) coding of binary data for a discrete noiseless channel whose restrictions may be modeled by a finite state machine. Codes for such channels arise in a wide variety of applications, which include digital transmission, magnetic or optical recording, and protocol formulation. The constraints are employed to obtain signal sequences with such properties as reduced intersymbol interference, sufficient transitions for self timing, or spectral nulls at prescribed frequencies. The literature in this area is quite extensive; references [1–12] provide an entry.

Magnetic recording is a good example of an application area. Codes for saturated digital recording produce binary symbols 1, 0 used to denote the presence and absence, respectively, of transitions between saturation levels. A lower bound d and upper bound k are generally placed on the number of 0's separating consecutive 1's in the coded stream. Examples of modern codes include a $(d, k) = (2, 7)$ variable length code with a bit per symbol ratio (rate) of 1/2 [13, 6], various $(1, n)$, $n \geq 6$, rate 2/3 codes [13, 10], and Zero Modulation [9], a code designed to control dc content in addition to run lengths. An interesting property of magnetic recording is that such codes yield substantial advantages in recording density and/or coder complexity [6, 10, 14].

A variety of approaches to formulating such channel codes is known. These include methods for constructing fixed and variable length state dependent codes [4–6, 10] and techniques which employ look-ahead. The latter codes in their most general form include the former as special cases [11]. Suppose the coder is constrained to utilize no more than M source characters, for a look-ahead of $M - 1$. The parameter M may be regarded as a coding delay, hence the term bounded-delay coding. References [11, 12] describe a number of properties of such codes, including necessary and sufficient conditions for existence. Also discussed is a general construction procedure [12], which is, however, given in detail only for $M \leq 2$. This paper treats the general case.

Suppose a code is to be constructed with a fixed rate of α bits per W channel symbols, where $(\alpha/W) \leq C$, the channel capacity [1]. The coder may be viewed as accepting a sequence $\{V_i\}$ of characters, each consisting of α bits, and emitting a sequence $\{w_j\}$ of words, each consisting of W channel symbols. The delay parameter M indicates that the word w_i corresponding to the character V_i is chosen with knowledge of no more than V_i and the next $M - 1$ characters $V_{i+1}, V_{i+2}, \cdots, V_{i+M-1}$, along with some information on the current channel state and previously emitted code words.

Let $\{\sigma_i\}$, $i = 1, 2, \cdots, S$, be the set of states in the finite state channel model Q. Transitions between neighboring states produce channel symbols. Let $\mathbf{D} = \{d_{ij}\}$ be the skeleton transition matrix for the channel; d_{ij} gives the number of ways of going from σ_i to σ_j without entering any intermediate states. Let $\mathbf{B} = \mathbf{D}^W$; $\{b_{ij}\}$ is the number of distinct paths of length W connecting σ_i to σ_j.

The matrix **B** may be considered as a skeleton transition matrix for a new finite state machine Q'. Transitions of length 1 in Q' correspond to paths of length W in Q.

It was shown [11, 12] that construction of a bounded delay code with parameter M requires the existence of a *state weight* vector Φ such that

1. $\sum_{j=1}^{S} B_{ij}\phi_j 2^{-\alpha} \geq \phi_i \qquad i = 1, 2, \cdots, S,$ (1)
2. $\phi_i \leq 1$, and (2)
3. Each ϕ_i is an integer multiple of $2^{-\alpha(M-1)}$; that is,

$$\phi_i = n_i 2^{-\alpha(M-1)}. \tag{3}$$

A state weight vector Φ for a given M not less than the minimum feasible value may be found by a simple recursive algorithm [12], which is reproduced in the appendix. A variety of vectors may be obtained by imposing additional constraints, for example, that each of the ϕ_i be less than some given value, by increasing M, or by increasing the value of α for a fixed ratio of α/W.

Once a set of state weights ϕ_i has been obtained, a code may be constructed. Procedures are described below for obtaining two classes of codes. The first type may be regarded as periodic. Each state σ_i has associated with it a path tree of depth $M - 1$ code words. A tree is entered once every $M - 1$ characters and traversed to the end with delay M (*i.e.*, look-ahead of $M - 1$ characters). The second class, termed stationary codes, may be regarded as a special case of the periodic codes. Here the path trees are constructed so that the coder need traverse each tree only to depth one. The result is a mapping between the sequence of characters $\{V_i\}$ and the sequence of code words $\{w_j\}$ such that a word w_i transmitted at time i is a function of at most [12]

1. The preceding $(M - 1)$ words,
2. The state occupied at time $i - M + 1$,
3. The next $M - 1$ characters.

Stationary codes yield simpler implementations, but the periodic type permit more flexibility in construction, which can be advantageous in applications such as those requiring state independent decoding [4]. If $M \leq 2$, the case treated in [12], there is no difference between stationary and periodic codes.

The techniques given here may be regarded as a means of code optimization, since a construction method for $M \leq 2$ is in fact sufficient for obtaining any fixed rate up to the channel capacity. To see why this is so, consider a weight vector Φ with parameter $M > 2$. Setting $W_1 = (M - 1)W$ and $\alpha_1 = (M - 1)\alpha$ results in a corresponding vector $\Phi_1 = \Phi$ for $\mathbf{B}_1 = \mathbf{B}^{M-1}$ whose components are integer multiples of $2^{-\alpha_1}$, that is, a vector suitable for coding with $M_1 \leq 2$. The resulting coder does not, however, have minimal look-ahead.

Not discussed are codes resulting from renormalization of the state weights [12] and code construction to ensure state independent decoding [4].

2. Periodic codes

The state weights ϕ_i may be regarded as representing an integral number of possibilities for the $(M - 1)$-tuple of characters $V_1, V_2, \cdots, V_{M-1}$ to be transmitted from σ_i. That is, n_i is the number of such distinct $(M - 1)$-tuples, where $n_i = \phi_i 2^{\alpha(M-1)}$. Code construction requires an assignment of such tuples to channel states so that an appropriate state is occupied when a particular vector is to be transmitted. The $(M - 1)$-tuples of characters are termed *messages*. State σ_i, $i = 1, 2, \cdots, N$, is then associated with n_i messages. Note that a state weight of 1 permits transmission of all possible messages.

- *Definition*

The r-successor tree $T(\sigma_i, r)$ is the set of paths of length r (in the Q' machine) from state σ_i. □

Figure 1 shows an example of a constrained channel and associated successor trees $T(\sigma_i, M - 1)$. Here $\alpha = 1$, $W = 1$, and $M = 3$. States σ_1, σ_2, and σ_3 have weights 1/4, 2/4, and 3/4, respectively.

Each state σ_i may be regarded as being associated with n_i units f_{ij} each of weight $2^{-\alpha(M-1)}$. A set of n_i subtrees of $T(\sigma_i, M - 1)$, termed *independent paths* or *IP*'s [11, 12], may be formed for each $T(\sigma_i, r)$. The *IP*'s correspond to a partition of a subset of state weight units associated with $T(\sigma_i, r)$ such that each IP_{ij}, $j = 1, 2, \cdots, n_i$, permits the encoding from σ_i of a message G_{ij} consisting of $M - 1$ specific characters, followed by anything else.

- *Definition*

Let $F_{ij}(M - 1)$, $j = 1, 2, \cdots, n_i$, be a set of disjoint weight trees whose nodes are weight units f_{pq} associated with states σ_p in $T(\sigma_i, M - 1)$. Each unit f_{pq} at depth d, $d < M - 1$, in $F_{ij}(M - 1)$ has 2^{α} successors f_{km} drawn from states σ_k which are successors to σ_p in $T(\sigma_i, M - 1)$. □

Each $F_{ij}(M - 1)$ contains weights drawn from a succession of states in $T(\sigma_i, M - 1)$. Let that subset of $T(\sigma_i, M - 1)$ to which it corresponds be termed IP_{ij}. Note that the *IP*'s associated with a given state σ_i are not necessarily disjoint. Figure 2 shows a weight tree F_{11} and IP_{11}, the corresponding subtree of $T(\sigma_1, 2)$.

Construction of a code entails the assignment of messages to *IP*'s. The methods described here are based on the observation that each IP_{ij} leads to $2^{\alpha(M-1)}$ succeeding *IP*'s, that is, sufficient *IP*'s to encode all messages. These succeed-

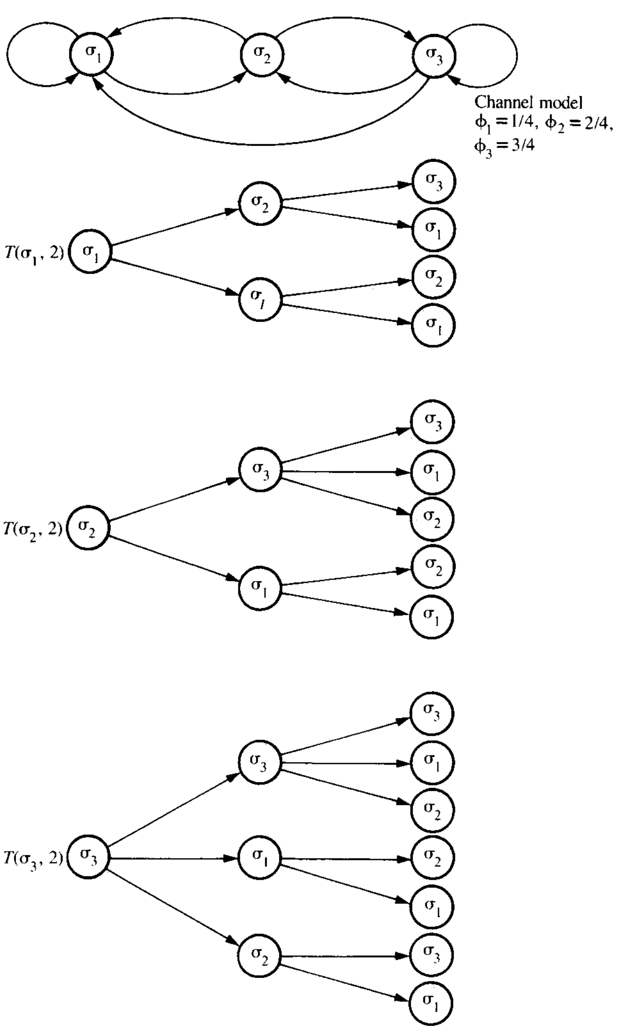

Figure 1 Construction of successor trees.

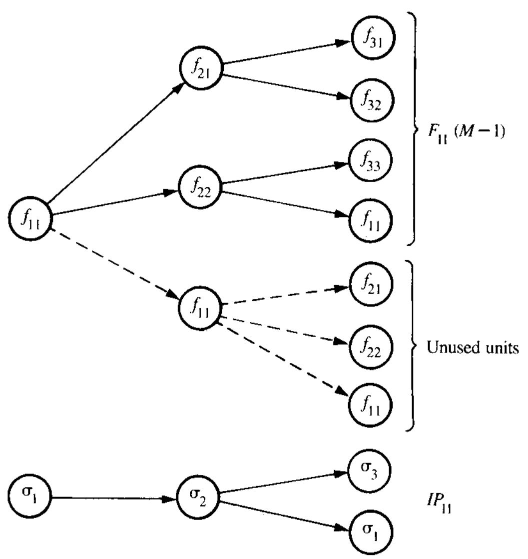

Figure 2 A weight tree and its associated *IP*.

ing *IP*'s, associated with the leaves of IP_{ij}, are assigned messages so as to produce codes with the properties described in Section 1.

Once a set of *IP*'s has been obtained, an assignment of messages $\{G_{ij}\}$ may be made to the leaves. Message assignments must be such that the tree can be traversed with delay M. This means that a branch at depth d in IP_{ij} requires that messages assigned to leaves corresponding to different paths from d must be distinguishable given knowledge of no more than the first $d + 1$ message characters to be transmitted starting at depth $M - 1$.

An assignment with this property can be obtained by associating a distinct message character in position d, $d = 1, 2, \cdots, M - 1$, with each branch from depth $d - 1$, $d = 1, 2, \cdots, M - 1$, in $F_{ij}(M - 1)$. Figure 3 shows the result for a tree $F_{22}(M - 1)$ in the example. The message to be transmitted from the leaf is then obtained by tracing the path from $d = 0$. Thus, for example, the leaf f_{21} is associated with the message 00. The $F_{ij}(M - 1)$ leaf assignments may then be mapped onto the IP_{ij}, as illustrated in Fig. 3. Note that the decision at $d = 0$ on whether to take the path $\sigma_2 \rightarrow \sigma_3$ or $\sigma_2 \rightarrow \sigma_1$ requires knowledge of only the first character of the message to be transmitted from the leaves, so that the *IP* can be traversed with delay M.

• *Definition*
$H(\sigma_i, M - 1)$ is the sequence of transmitted words $w_{-(M-1)}, \cdots, w_{-2}, w_{-1}$ leading to the current state σ_i, along with the state σ_k occupied when transmitting $w_{-(M-1)}$. □

The set of messages that can be received by the coder when occupying state σ_i at the end of an *IP* is a function of the *IP* and $H(\sigma_i, M - 1)$. Thus, in the example shown in Fig. 3, σ_2, if reached via $\sigma_2 \rightarrow \sigma_3 \rightarrow \sigma_2$ in IP_{22}, has 00 and 01 as possible messages. These may be assigned to IP_{21} and IP_{22}, respectively. It will be shown below that *IP*'s can be formed and messages assigned so that those associated with a given leaf σ_j of $T(\sigma_j, M - 1)$ are distinct. That is, any given message appears at most once at each state. This permits a code mapping between messages and *IP*'s which is only a function of the state σ_i occupied at the beginning of an *IP* and the history $H(\sigma_i, M - 1)$. Codes with this property will be termed *regular*.

• *Definition*
A set of weight units $\{f_{ij}\}$ will be said to be *lexicographically ordered* if

1. For any $f_{ij}, f_{pq} \in \{f_{ij}\}, f_{ij} < f_{pq}$ or $f_{pq} < f_{ij}$ if $(i, j) \neq (p, q)$.
2. If $i \neq p$ and $f_{ij} < f_{pq}$ for some j, q, then $f_{ij} < f_{pq}$ for all j, q. □

Suppose that, for each state σ_j at depth d, $0 \le d < M - 1$ in $T(\sigma_i, M - 1)$, the set of state weight units f_{pq} associated with successor states σ_r at depth $d + 1$ is given a lexicographic ordering. For example, in Fig. 2, the units associated with successors to σ_1 at depth 1 are f_{21}, f_{22}, and f_{11} with the ordering $f_{21} < f_{22} < f_{11}$ (*i.e.*, f_{21} is of highest order). If the trees $F_{ij}(M - 1)$ are formed so that $F_{ij}(M - 1)$ contains f_{ij} as a root and so that for $j = 1, 2, \cdots, n_i$, F_{ij} is formed by taking remaining successors at each depth d in decreasing order [that is, each chosen successor f_{pq} is of lower order than any previously chosen from this node σ_r in $T(\sigma_i, M - 1)$], the weight trees will be termed *completely ordered,* with $F_{ij}(M - 1) < F_{ik}(M - 1)$, $j < k$. Note that $F_{11}(M - 1)$, shown in Fig. 2, is completely ordered if $f_{31} < f_{32} < f_{33} < f_{11}$ for σ_3, σ_1 successor states to σ_2 from depth 1, and $f_{21} < f_{22} < f_{11}$ for σ_2, σ_1 successors to state σ_1 at depth 0.

A regular code may be obtained by first constructing, for each state σ_i, a completely ordered set of $F_{ij}(M - 1)$. Messages are then assigned to the leaves of the $F_{ij}(M - 1)$ as discussed above, with characters being associated with branches from each node f_{pq} in order $V_1, V_2, \cdots$. Note that a leaf σ_r of $T(\sigma_i, M - 1)$ is shared by no more than two *IP*'s, which must be consecutive. That is, if the *IP*'s are IP_{ij} and IP_{ik} with $j < k$, then $k = j + 1$. Moreover, no two *IP*'s share more than a single leaf, and each state σ_r has at most $2^{\alpha(M-1)}$ messages in the assignment. It follows that these messages are distinct. This is because messages are assigned in increasing order to a linearly ordered set of leaves. Thus a leaf associated with two *IP*'s cannot be assigned the same message from both.

In the treatment below, it is assumed that all codes being discussed are regular.

Figure 4 shows a completed regular code for the constrained channel of Fig. 1. To illustrate the operation of the coder, note that σ_3, if reached at the end of the path $\sigma_1 \rightarrow \sigma_2 \rightarrow \sigma_3$, is assigned the messages $\{G_{3j}\} = \{00, 01, 10\}$. IP_{31} is followed given reception of 00, IP_{32} given reception of 01, and IP_{33} if the message is 10. Suppose the message is 01, corresponding to IP_{32}. Then the path $\sigma_3 \rightarrow \sigma_3$ is followed if the first character of the next message is 0 and $\sigma_3 \rightarrow \sigma_1$ is followed otherwise. For this case, the second transition of the *IP* is not a function of the next message.

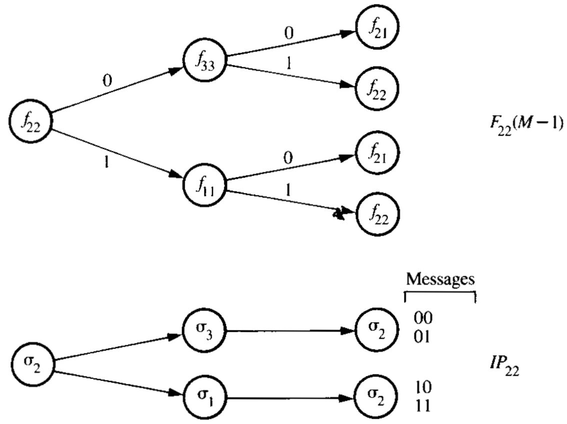

Figure 3 Message assignment.

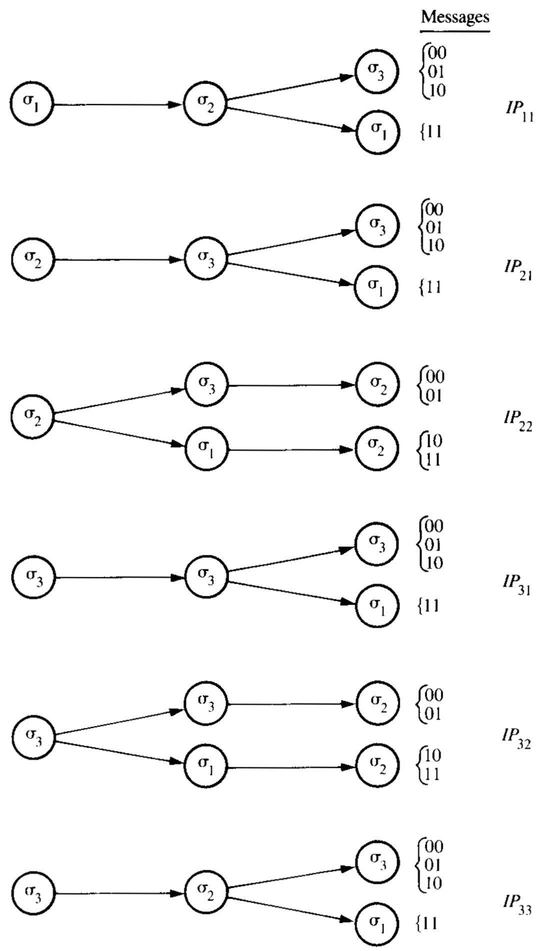

Figure 4 A set of *IP*'s and message assignments.

3. Decoding

Two significant parameters associated with decoding are the delay and the extent of error propagation.

• *Definition*

If a word w_i can be decoded given knowledge of no more than $w_i, w_{i+1}, \cdots, w_{i+\Delta}$, the decoding delay is Δ. □

If $\Delta = 0$, decoding will be termed instantaneous.

• *Definition*

Two *IP*'s associated with a state σ_i will be termed distinct from depth d if they share no transition past depth d. □

To illustrate this notion, consider Fig. 4. Note that IP_{31} and IP_{32} share the transition $\sigma_3 \rightarrow \sigma_3$. They are distinct past depth 1.

• *Definition*
Two *IP*'s will be termed *distinguishable* if they are distinct past depth $M - 2$. □

All *IP*'s in Fig. 4 are distinguishable. Note that if all *IP*'s are distinguishable, the decoding delay is $\Delta \leq M - 2$. To see why this is so, note that, for an *IP* starting at state σ_i, $i = 1, 2, \cdots, S$, the history $H(\sigma_i, M - 1)$ determines the set of messages associated with σ_i in this instance. Knowledge of the particular *IP* is then sufficient for decoding.

If some *IP*'s are not distinguishable, knowledge of subsequent ones is sometimes required for decoding. If non-distinguishable *IP*'s lead to others with this property, decoding of some messages may be delayed indefinitely. Here formal decodability may be ensured via use of an appropriate message termination, which is in any case generally required if decoding is not instantaneous. Practical systems, however, must have short decoding delays to control implementational complexity and error propagation. The delay is in general a function of the channel restrictions, the vector ϕ, and the code design, including the parameter M. Increasing M beyond the minimum may in some cases result in a smaller value of Δ.

Consider the problem of error propagation. Suppose first that the channel is of finite memory L. This means that L correctly received channel symbols are sufficient to determine the current state. A given *IP* can be correctly decoded provided that the error occurred at least $L + 1$ channel symbols before the preceding $M - 1$ words. The total number of decoding errors, given a single error in detection, is also a function of the decoding delay, since an error in a given word w_i can affect Δ preceding ones.

If the channel constraints are not of finite memory, limiting error propagation requires that decoding be to some extent state independent [4]. An example of such constraints is the requirement that the sequence of channel symbols have a bounded running sum [4], which forces a null in the signal spectrum at zero frequency. In the present context, a sufficient condition for state independent decoding is that any two *IP*'s which share a sequence of $M - 1$ code words be assigned the same message.

4. Coding with no memory

Consider the operation of the coder from state σ_i at the start of an *IP*. The set of possible messages is determined by $H(\sigma_i, M - 1)$, as is their assignment to *IP*'s. There are n_i such messages, each consisting of $M - 1$ characters. Let the set of all messages be $\{q_i\}$, $i = 1, 2, \cdots, 2^{\alpha(M-1)}$. It will be shown here that certain assignments of messages to *IP*'s have the property that no previous state information need be kept by the coder.

• *Definition*
A map $A_i(q_j) \rightarrow IP_{ik}$ is an assignment, at each state σ_i, of each possible message onto an *IP* independent of previous path information. $A_i(q_j) \rightarrow \varnothing$, the empty set, for those $\{q_j\}$ which are not assigned to σ_i for any $H(\sigma_i, M - 1)$. □

Note that a map of the above form makes it unnecessary for the coder to maintain any information concerning $H(\sigma_i, M - 1)$.

• *Definition*
A complete ordering $<$ of the set of messages is one where $q_i < q_j$, or $q_j < q_i$ for $i \neq j$. □

• *Definition*
Suppose q_j and q_k are, respectively, the smallest and largest messages (according to the ordering) assigned to σ_i for some particular $H(\sigma_i, M - 1)$. The assignment will be termed *contiguous* if for any q_m not assigned to σ_i for this $H(\sigma_i, M - 1)$, $q_m < q_j$ or $q_m > q_k$. □

If the assignments are all contiguous, it is possible to construct a map $A_i(q_j) \rightarrow IP_{ik}$. The procedure is analogous to that for ensuring state independent decodability [4] for the case of $M = 1$. In the current instance the messages are history dependent and the *IP*'s are not. Any two identical messages resulting from different histories must be assigned the same *IP*; in the case of state independent decoding for $M = 1$ the *IP*'s (code words) are state dependent and not the messages. A simple procedure for constructing a code in the latter case is given in reference [4]. An analogous method may be used for assigning messages to *IP*'s.

Note that the assignments in Fig. 4 are contiguous if the orderings of messages are thought of as binary numbers with the implied ordering (*e.g.*, $00 < 01$).

5. Stationary codes

This section describes a class of codes with often simpler implementations than the periodic ones of Section 2. They are less general in that their construction imposes stronger relations between *IP*'s and messages associated with different states.

One way of viewing these codes is as follows. Suppose the first transition (code word) taken on an *IP* associated with σ_i brings the channel to σ_j. If the message assignment and last $M - 2$ transitions of the *IP* associated with σ_i are compatible with the message assignment and first $M - 2$ transitions from σ_j, the coder can switch to *IP*'s associated with σ_j. The result is that only the first transitions associated with each *IP* need be kept by the coder.

A construction procedure for such codes is given below. The result is a mapping

$$J(V_t, V_{t+1}, \cdots, V_{t+M-1}; H(\sigma_i, M-1)) \rightarrow w_t \tag{4}$$

between the current character V_t and code word w_t as a function of $H(\sigma_i, M-1)$ and the next $M-1$ characters. Decoding is done as described in Section 3.

- *Step 1*

For each state σ_i, partition the weights of the leaves σ_r of $T(\sigma_i, 1)$ into units f_{pq} of weight $2^{-\alpha(M-1)}$. A subset of $n_i 2^\alpha$ such units, taken in some lexicographic order, is then labeled as $u_{ij}, j = 1, 2, \cdots, n_i 2^\alpha$. □

Figure 5 illustrates the results of Step 1 for the $T(\sigma_i, 1)$ associated with the channel shown in Fig. 2. Note that no labels are associated with σ_1 as a leaf of $T(\sigma_1, 1)$, since u_{11} and u_{12} are sufficient for $n_1 = 1$.

- *Definition*

If $m < n$, X_{im} will be termed of higher precedence than X_{in} for any object X. □

- *Step 2*

To each u_{im} at depth 1 in $T(\sigma_i, M-1)$, $m = 1, 2, \cdots, n_i 2^\alpha$, assign the 2^α highest precedence remaining successors $\{u_{kn}\}$ at depth 2, where u_{kn} is a successor to u_{im} if σ_k is a successor to σ_i. For each of the u_{kn}, $n = 1, 2, \cdots$, at depth d, assign successors at depth $d + 1$ in the same fashion. Continue this process until the $\{u_{ij}\}$ have been assigned at depth $M - 1$. □

Step 2 yields a set of trees $U_{ij}(M-1)$ whose roots are σ_i and nodes are $\{u_{km}\}$. The leaves at depth $M - 1$ have total weight $2^{-\alpha}$. $U_{ij}(M-1)$ may be regarded as capable of transmitting a specific message G_{ir}, followed by one character, followed by anything else. That is, 2^α of the $U_{ij}(M-1)$ may be used to form an *IP*. Figure 6 shows the set of $U_{ij}(M-1)$ associated with state σ_1 for the example of Fig. 5.

- *Step 3*

Assign the 2^α highest precedence unassigned trees $U_{ik}(M-1)$ to IP_{ij}, $j = 1, 2, \cdots, n_i$. □

- *Step 4*

For each leaf σ_k of $T(\sigma_i, M-1)$, assign the u_{rm} associated with σ_k with the IP_{kj} of the same precedence.

Step 4 completes the formation of the *IP*'s, which can now be assigned messages.

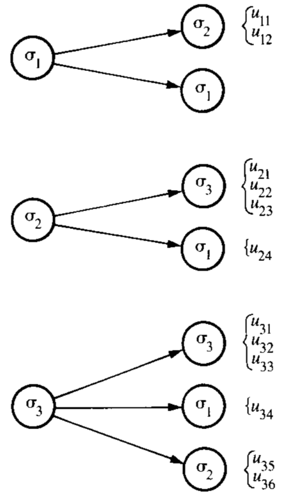

Figure 5 Ordering of unit weights.

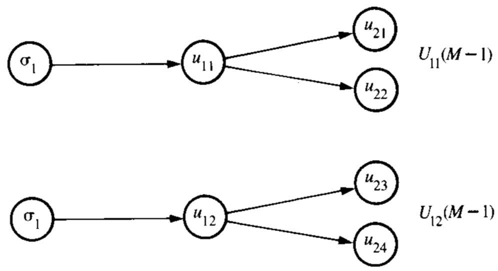

Figure 6 $U_{1j}(M-1)$.

- *Step 5*

Suppose the set of possible characters is V_r, $V = 1, 2, \cdots, 2^\alpha$. For each IP_{ik}, $k = 1, 2, \cdots, n_i$, assign the lowest numbered V_r to the leaves of the highest precedence unassigned $U_{ij}(M-1)$ associated with IP_{ik}. □

The above procedure differs from that of Section 2 for regular periodic codes in the stronger ordering conditions in *IP* construction. It follows that the observation that *IP*'s can be traversed with delay M holds here as well.

Figure 7 shows the character assignments for the example. Consider $T(\sigma_2, M-1)$. IP_{21} terminates in states σ_3 and σ_1. Messages associated with σ_3 where $H(\sigma_3, M-1) = \sigma_2 \rightarrow \sigma_3 \rightarrow \sigma_3$ start with 0 for IP_{31} and IP_{32}, and with 1 for IP_{33}.

The following example illustrates the coding operation. Consider Fig. 7 and suppose the channel occupies state σ_3. Let the sequence of characters $V_{i(1)} V_{i(2)} \cdots$ to be encoded

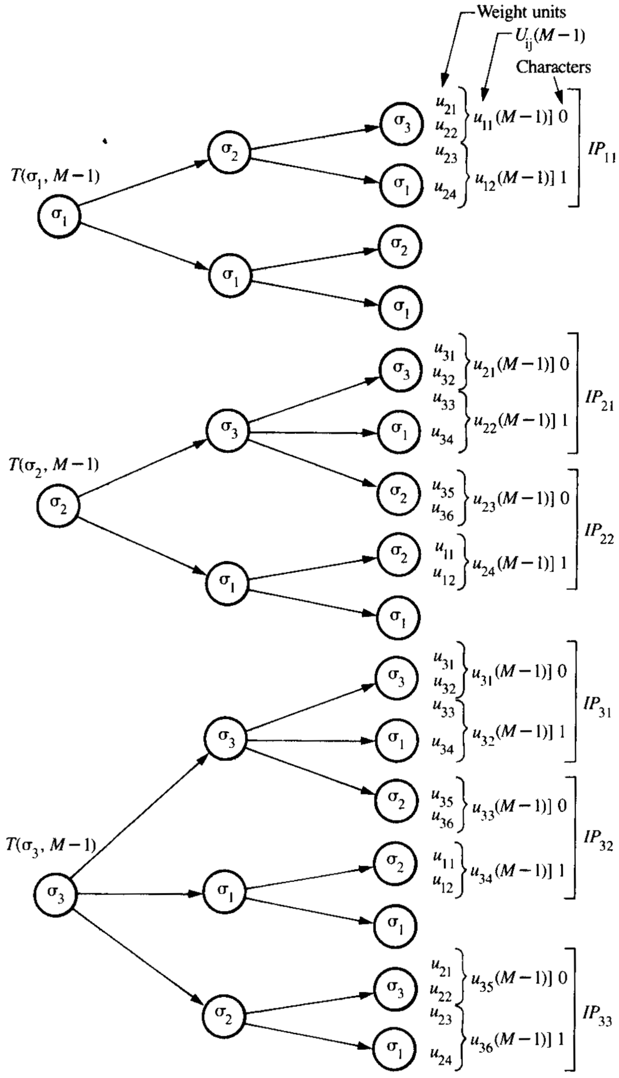

Figure 7 Character assignments.

be xy 010 · · ·, and suppose xy corresponds to IP_{32}. $V_{i(3)} = 0$ yields $\sigma_3 \rightarrow \sigma_3$ in IP_{32}. Note that the second code word in IP_{32} corresponds to the first in IP_{33} after the transition $\sigma_3 \rightarrow \sigma_3$. That is, IP_{33} is to be followed with y01, the characters seen by the coder. This produces the transition $\sigma_3 \rightarrow \sigma_2$. The characters seen by the coder are now 010. Note that 01 corresponds to IP_{22} from σ_2. The 0 which follows this message determines the code word, which is $\sigma_2 \rightarrow \sigma_3$.

Consider a particular $U_{ik}(M-1)$, say $U_{i1}(M-1)$, the one of highest precedence. Suppose the first transition w_q takes the machine Q' to state σ_j [there is only one initial code word associated with $U_{ik}(M-1)$]. Note that the subtree of the $U_{i1}(M-1)$ tree, starting from σ_j, corresponds to the 2^α highest precedence $U_{jk}(M-1)$ taken to depth $M-2$. The leaves of $U_{i1}(M-1)$ are associated with some particular source character V_r. The leaves of the 2^α highest precedence $U_{jk}(M-1)$ are associated with the entire set of 2^α characters. Thus the encoding of the character V_r can be followed by the encoding of any other. This is true for the $U_{ik}(M-1)$ of highest precedence, from which it follows that it is true for the one next highest. The observation that this holds for all $U_{ik}(M-1)$ follows by induction.

Once the form shown in Fig. 7 has been obtained, characters can be mapped onto transitions from the leaves of $T(\sigma_i, M-1)$. Note, for example, that the leaves of $U_{21}(M-1)$ in Fig. 7 are assigned a 0, while those of $U_{22}(M-1)$ are assigned a one. Thus the first characters of the messages assigned to IP_{31}, IP_{32}, and IP_{33} when $H(\sigma_i, 2) = \sigma_2 \rightarrow \sigma_3 \rightarrow \sigma_3$ are 0, 0, and 1, respectively. Figure 8 shows the encoding of characters onto the first transitions of *IP*'s starting at the leaves of $T(\sigma_i, 2)$, $i = 1, 2, 3$.

At this point in the construction, one may delete extraneous history information. Note, for example, that the assignment for σ_3 in Fig. 8 is not a function of the previous path. Figure 9 shows the result. Trees to depth M may be obtained by concatenating the appropriate ones of depth 1. This process may be viewed as a means for solving the messages associated with each state as a function of its history. These may then be used to obtain a code mapping of the form (4). Table 1 is of this form. Note that the current and next state determine the code word for this example. Decoding may be performed with delay $\Delta = 0$.

It can be observed that for this example the assignment of messages to the *IP*'s is contiguous, permitting the elimination of $H(\sigma_i, M-1)$ in the coding process. One way to do so is to replace the entries for σ_2 in Table 1 with

Current state	*Input characters*	*Next state*
σ_2	$X0$	σ_3
σ_2	$X10$	σ_3
σ_2	$X11$	σ_1

where X, representing the current character, is either a 0 or a 1.

6. Conclusion

Algorithms were described for constructing two classes of minimum delay fixed rate codes for discrete noiseless channels. The first class, periodic codes, have the disadvantage of greater implementational complexity, but offer more freedom in *IP* construction and message assignment. This can be of significance for applications and constraints such that stationary codes yield long decoding delays or if state independent decoding is required. Stationary codes, the second class, although conceptually more complex, yield simple implementations and are thus the preferred mode for most applications.

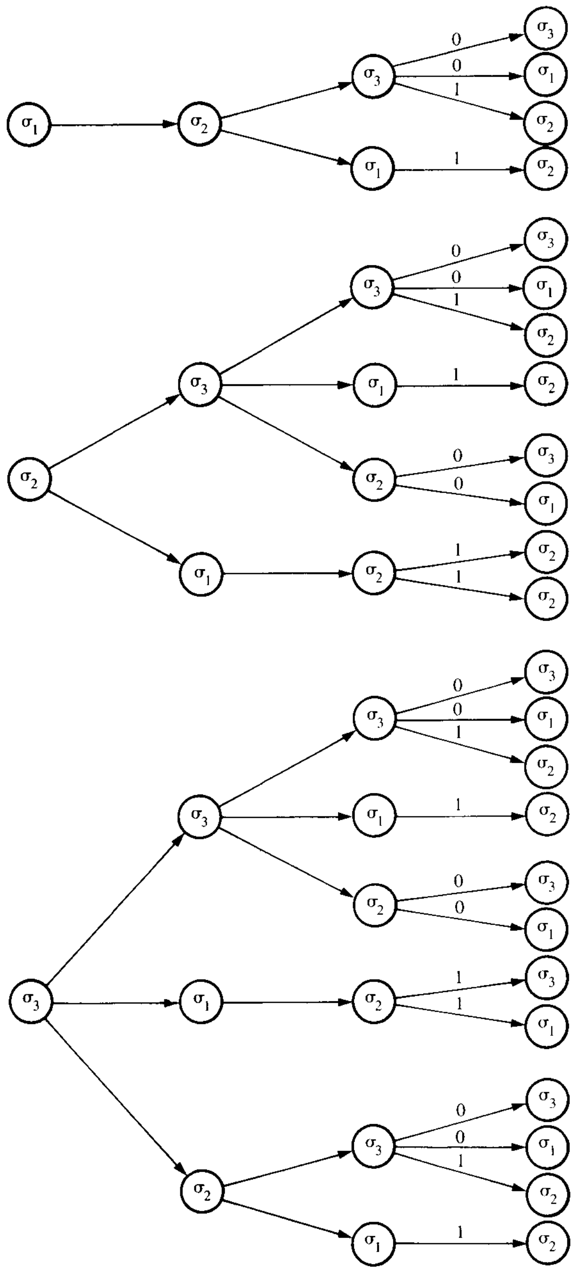

Figure 8 Map of characters onto code words.

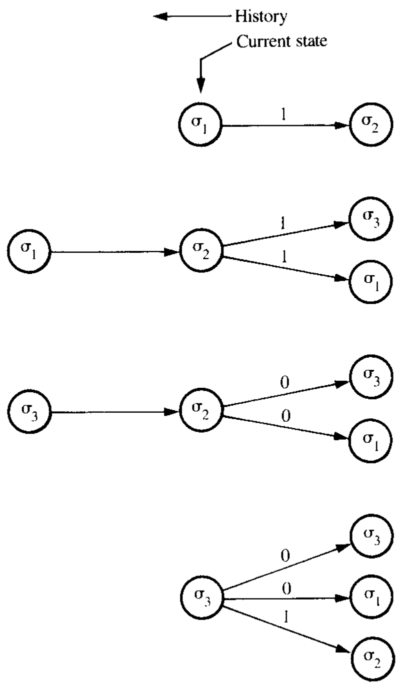

Figure 9 Simplified transition assignments.

Table 1 Code table.

Previous state	*Current state*	*Input characters*	*(Code word) Next state*
	σ_3	00	σ_3
	σ_3	010	σ_3
	σ_3	011	σ_1
	σ_3	10	σ_2
σ_3	σ_2	00	σ_3
σ_3	σ_2	010	σ_3
σ_3	σ_2	011	σ_1
σ_1	σ_2	10	σ_3
σ_1	σ_2	110	σ_3
σ_1	σ_2	111	σ_1
	σ_1	11	σ_2

Appendix

The following is a recursive algorithm for obtaining a set of state weights $\{\phi_i\}$ [11, 12].

Let $\phi_i^*(n)$ denote the state weights at the nth iteration. Then

$$\phi_i^V(n+1) = 2^{-\alpha}\left[\sum_{ij} B_{ij}\phi_j^*(n)\right],$$

$$\phi_i^*(n) = 2^{-(M-1)\alpha} \llcorner [2^{(M-1)\alpha}\phi_i^V(n)],$$

subject to the condition that

$$\phi_i^* \le 1, \forall_i,$$

$$\phi_i^*(0) = 1.$$

The procedure is continued until there is no change over one iteration [*i.e.*, $\phi_i^*(n+1) = \phi_i^*(n), \forall_i$]. The symbol $\llcorner$ denotes the floor function.

8. References

1. C. E. Shannon, "A Mathematical Theory of Communication," *Bell Syst. Tech. J.* **27,** 379 (1948).
2. C. V. Freiman and A. D. Wyner, "Optimum Block Codes for Noiseless Channels," *Info. Control* **7,** 398–415 (1964).
3. W. H. Kautz, "Fibonacci Codes for Synchronization Control," *IEEE Trans. Info. Theory* **IT-11,** 284–292 (1965).
4. P. A. Franaszek, "Sequence-State Coding for Digital Transmission," *Bell Syst. Tech. J.* **47,** 143–157 (1968).
5. P. A. Franaszek, "On Synchronous Variable-Length Coding for Discrete Noiseless Channels," *Info. Control* **15,** 155–164 (1969).
6. P. A. Franaszek, "Sequence-state Methods for Run-length-limited Coding," *IBM J. Res. Develop.* **14,** 376–383 (1970).
7. D. T. Tang and L. R. Bahl, "Block Codes for a Class of Constrained Noiseless Channels," *Info. Control* **17,** 436–461 (1970).
8. H. Kobayashi, "A Survey of Coding Schemes for Transmission or Recording of Digital Data," *IEEE Trans. Commun. Technol.* **COM-19,** 1087–1100 (1971).
9. A. M. Patel, "Zero-Modulation Encoding in Magnetic Recording," *IBM J. Res. Develop.* **19,** 366–378 (1975).
10. T. Horiguchi and K. Morita, "An Optimization of Modulation Codes in Digital Recording," *IEEE Trans. Magnetics* **MAG-12,** 740–742 (1976).
11. P. A. Franaszek, "Synchronous Bounded Delay Coding for Input Restricted Channels," *IBM J. Res. Develop.* **24,** 43–48 (1980).
12. Peter A. Franaszek, "A General Method for Channel Coding," *IBM J. Res. Develop.* **24,** 638–641 (1980).
13. P. A. Franaszek, "Run-Length-Limited Variable Length Coding with Error Propagation Limitation," U.S. Patent 3,689,899, 1972.
14. J. M. Harker, D. W. Brede, R. E. Pattison, G. R. Santana, and L. G. Taft, "A Quarter Century of Disk File Innovation," *IBM J. Res. Develop.* **25,** 677–689 (1981).

Received August 28, 1981; revised February 25, 1982

The author is located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.