Reprinted by permission from
IEEE TRANSACTIONS ON COMMUNICATION TECHNOLOGY
Vol. COM-19, No. 5, October 1971

Nonsystematic Convolutional Codes for Sequential Decoding in Space Applications

JAMES L. MASSEY, MEMBER, IEEE, AND DANIEL J. COSTELLO, JR., MEMBER, IEEE

***Abstract*—Previous space applications of sequential decoding have all employed convolutional codes of the systematic type where the information sequence itself is used as one of the encoded sequences. This paper describes a class of rate 1/2 nonsystematic convolutional codes with the following desirable properties: 1) an undetected decoding error probability verified by simulation to be much smaller than for the best systematic codes of the same constraint length; 2) computation behavior with sequential decoding verified by simulation to be virtually identical to that of the best systematic codes; 3) a "quick-look-in" feature that permits recovery of the information sequence from the hard-decisioned received data without decoding simply by modulo-two addition of the received sequences; and 4) suitability for encoding by simple circuitry requiring less hardware than encoders for the best systematic codes of the same constraint length. Theoretical analyses are given to show 1) that with these codes the information sequence is extracted as reliably as possible without decoding for nonsystematic codes and 2) that the constraints imposed to achieve the quick-look-in feature do not significantly limit the error-correcting ability of the codes in the sense that the Gilbert bound on minimum distance can still be attained under these constraints. These codes have been adopted for use in several forthcoming space missions.**

Paper approved by the Communication Theory Committee of the IEEE Communication Technology Group for publication without oral presentation. This work was supported by NASA under Grant NGL15-004-026. Manuscript received May 5, 1971.
J. L. Massey is with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, Ind. 46556.
D. J. Costello, Jr., is with the Department of Electrical Engineering, Illinois Institute of Technology, Chicago, Ill. 60616.

I. Introduction

THE DEEP-SPACE channel can be accurately modeled as the classical additive white Gaussian noise channel. Sequential decoding of convolutional codes is the best presently known technique for making

efficient use of the deep-space channel [1]. In this paper, we describe a class of convolutional codes developed especially for use with sequential decoding on the deep-space channel that offers advantages in undetected error probability and in ease of implementation over the convolutional codes previously used on the deep-space channel. In the remainder of this section, we present the background material in convolutional coding needed for the description and analysis of these new codes.

Let $i_0, i_1, i_2, \cdots$ be a sequence of binary information digits which are to be encoded into a convolutional code. It is convenient to represent such a sequence by its D transform

$$I(D) = i_0 + i_1 D + i_2 D^2 + \cdots .$$

A rate $R = 1/N$ convolutional encoder of memory order m is simply a single-input N-output linear sequential circuit whose outputs at any time depend on the present input and the m previous inputs. The N output sequences are the encoded sequences for the input information sequence and are interleaved into a single stream for transmission over the noisy communications channel. Letting $T^{(j)}(D)$ be the D transform of the jth encoded sequence, that is

$$T^{(j)}(D) = t_0^{(j)} + t_1^{(j)} D + t_2^{(j)} D^2 + \cdots$$

we can represent the encoder by N transfer functions

$$G^{(j)}(D) = g_0^{(j)} + g_1^{(j)} D + \cdots + g_m^{(j)} D^m$$

in the manner that

$$T^{(j)}(D) = I(D) G^{(j)}(D) \tag{1}$$

where all additions and multiplications in the evaluation of (1) are carried out in modulo-two arithmetic (i.e., in the finite field of two elements). In coding terminology, the transfer functions $G^{(j)}(D)$ are called the "code generating polynomials" of the convolutional code.

The convolutional code is said to be "systematic" if

$$T^{(1)}(D) = I(D) \tag{2}$$

or equivalently if

$$G^{(1)}(D) = 1. \tag{3}$$

(If $T^{(j)}(D) = I(D)$, for some $j \neq 1$, we exchange the labels on the first and jth output terminals and still consider the code to be systematic.) A systematic convolutional code is thus just simply one in which the first transmitted sequence is the information sequence itself.

The span of $(m + 1)N$ encoded digits generated by the encoder at times $0, 1, \cdots, m$ are all of the encoded digits affected by the first information digit i_0 and are called the "initial code word" [2], and the number $n_A = (m + 1)N$ is called the "constraint length" of the code. The "minimum distance" d_m of the code is the fewest number of positions in which two initial code words arising from different values of i_0 are found to differ, and it is easily shown that d_m is equal to the fewest number of nonzero digits in any initial code word with $i_0 = 1$.

II. Systematic Versus Nonsystematic Codes

The principal advantage of systematic codes is that the information sequence is simply recoverable from the encoded sequences by virtue of (2). Suppose that, as is the case in many applications, hard decisions are made at the receiver on each transmitted digit. The jth hard-decisioned received sequence

$$R^{(j)}(D) = r_0^{(j)} + r_1^{(j)} D + r_2^{(j)} D^2 + \cdots$$

may then be expressed as

$$R^{(j)}(D) = T^{(j)}(D) + E^{(j)}(D) \tag{4}$$

where the addition is again modulo-two, and

$$E^{(j)}(D) = e_0^{(j)} + e_1^{(j)} D + e_2^{(j)} D^2 + \cdots$$

is the sequence of errors in the hard decisions, that is $e_u^{(j)} = 1$ if and only if $r_u^{(j)} \neq t_u^{(j)}$. For many engineering purposes such as synchronization and monitoring, it is desirable to get reasonably good estimates of the information digits directly from the hard-decisioned received sequences without employing the lengthy decoding process which is often carried out at a remote site much later in time. In systematic codes, since by (2) and (4)

$$R^{(1)}(D) = I(D) + E^{(1)}(D) \tag{5}$$

the digits in $R^{(1)}(D)$ can be taken directly as the estimates of the information digits with the same reliability as the hard decisions themselves. In nonsystematic codes where (2) does not hold, this convenience is lost, a fact which has been a major deterrent in the use of nonsystematic codes.

Wozencraft and Reiffen [3] have shown that for any nonsystematic code there is a systematic code with precisely the same set of initial code words for $i_0 = 1$ and thus with the same minimum distance d_m. This result showed that there was no advantage for nonsystematic codes when used with decoders such as threshold decoders [2] which make their decoding decision for i_0 on the basis of the initial code word only. This fact seems to have caused the possible advantages of nonsystematic codes with other type decoders to be often overlooked.

A final factor militating against the use of nonsystematic codes has been a psychological reluctance to entrust data to any code where the data do not appear explicitly in the encoded output based on the twin fears that such encoders would be more difficult to implement, and that catastrophic errors might occur in the recovery of the data from the hard-decisioned received sequences.

On the other hand, recent results [4]–[6] have confirmed the inherent superiority in undetected decoding error probability of nonsystematic codes over systematic codes when used with sequential decoders [7], [8] and maximum likelihood decoders [9] which may examine received digits well beyond the initial code word before

making the decoding decision on i_0. The error probability of such decoders is most closely related to the code parameter "free distance," d_{free}, which is defined as the fewest number of positions in which the entire encoded sequences can differ for $i_0 = 0$ and $i_0 = 1$. It is easily shown that d_{free} is equal to the fewest number of nonzero digits in any entire encoded output for $i_0 = 1$. Costello [5] has shown that for a given memory order m more free distance can be obtained with nonsystematic codes than with systematic codes.

The preceding considerations led the authors to look for nonsystematic convolutional codes which would exhibit good performance with sequential decoders but would also retain as much as possible the ease of extracting the information digits from the hard-decisioned received sequences characteristic of systematic codes. The remainder of this paper describes the successful outcome of such a search for rate $R = 1/2$ codes. The rate $R = 1/2$ was chosen as the rate of greatest practical interest in deep-space applications, since the resultant doubling of bandwith compared to no coding suffices to attain within 1 dB the total gain possible by coding [1] but does not lower the energy per transmitted bit unacceptably for the operation of the receiver tracking loops. The extension to lower rates of the form $R = 1/N$ should be evident to the reader.

III. Quick-Look-In Nonsystematic Codes

It has been shown [10] that in order to avoid "catastrophic error propagation," i.e., to avoid a finite number of errors in estimating the encoded sequences from being converted to infinitely many errors in estimating the information digits, the necessary and sufficient condition is that the encoder should possess a feedforward (FF) inverse. An FF inverse for an $R = 1/N$ encoder is simply an N-input single-output linear sequential circuit with polynomial transfer functions $P_j(D)$, $j = 1, 2, \cdots, N$, such that

$$\sum_{j=1}^{N} P_j(D)T^{(j)}(D) = D^L I(D). \tag{6}$$

That is, passing the encoded sequences through the FF inverse, results in recovering the information sequence except for a possible delay of L time units. With the use of (1), (6) may be rewritten as

$$\sum_{j=1}^{N} P_j(D)G^{(j)}(D) = D^L. \tag{7}$$

For the special case $N = 2$, i.e., $R = 1/2$, (6) and (7) become

$$P_1(D)T^{(1)}(D) + P_2(D)T^{(2)}(D) = D^L I(D) \tag{8}$$

and

$$P_1(D)G^{(1)}(D) + P_2(D)G^{(2)}(D) = D^L. \tag{9}$$

(Note that for the special case of a systematic code (3) implies that (9) is satisfied by the simple choice $P_1(D) = 1$, $P_2(D) = 0$, i.e., the FF inverse is entirely trivial.)

Equation (8) suggests forming an estimate of the information sequence $I(D)$ by passing the hard-decisioned estimates $R^{(1)}(D)$ and $R^{(2)}(D)$ of $T^{(1)}(D)$ and $T^{(2)}(D)$ through the FF inverse. The resulting esimate is given by

$$P_1(D)R^{(1)}(D) + P_2(D)R^{(2)}(D) = D^L[I(D) + \Delta(D)] \tag{10}$$

where

$$\Delta(D) = \delta_0 + \delta_1 D + \delta_2 D^2 + \cdots$$

is the sequence of errors in the estimated information digits. The use of (4) and (8) in (10) then gives

$$P_1(D)E^{(1)}(D) + P_2(D)E^{(2)}(D) = D^L\,\Delta(D) \tag{11}$$

which is the basic equation relating the errors in the hard-decisioned received sequences to the errors in the estimated information digits. Suppose, as would be the case for the deep-space channel, that each hard-decisioned received digit has probability p of being in error independently of the other digits, i.e., $e_u^{(j)} = 1$ with probability p independently of the value of the other error digits. It follows from (11) that if $p \ll 1$ then $\delta_i = 1$ with probability p times the total number of nonzero terms in the polynomials $P_1(D)$ and $P_2(D)$. Letting $W[P_j(D)]$ denote the number of nonzero terms in $P_j(D)$, i.e., the "Hamming weight" of $P_j(D)$, we may write the probability of error p_δ in the estimated information digits for $p \ll 1$ as

$$p_\delta = \{W[P_1(D)] + W[P_2(D)]\}p. \tag{12}$$

We call the quantity

$$A = W[P_1(D)] + W[P_2(D)] \tag{13}$$

appearing in (12) the "error amplification factor" since it relates the increased error probability at the output of the FF inverse to the input error probability. We note that A takes on its minimum possible value of 1 for systematic codes where we may choose $P_1(D) = 1$ and $P_2(D) = 0$.

For nonsystematic codes, the minimum possible error amplification factor is $A = 2$ and is attained for codes which permit $P_1(D) = P_2(D) = 1$. For such codes, we note from (10) that

$$R^{(1)}(D) + R^{(2)}(D) = D^L[I(D) + \Delta(D)] \tag{14}$$

so that the FF inverse which forms the estimates of the information digits from the hard-decisioned received sequences is instrumented simply by a single modulo-two adder which adds these sequences together. We note also from (9) that the choice $P_1(D) = P_2(D) = 1$ is possible if and only if

$$G^{(1)}(D) + G^{(2)}(D) = D^L \tag{15}$$

that is if and only if the two code generating polynomials differ only in a single term. We call any $R = 1/2$ nonsystematic convolutional code satisfying (15) a quick-look-in code, and note that such codes permit recovery of the information sequence $I(D)$ from the hard-deci-

sioned received sequences using a single modulo-two adder and with the minimum error amplification factor of 2 for nonsystematic codes.

Quick-look-in codes allow the information sequence to be recovered from the hard-decisioned received sequences almost as simply and as reliably as do systematic codes. It remains to show that there are quick-look-in codes which give better performance with sequential decoding than the best systematic codes with the same constraint length.

IV. Sequential Decoding Considerations and Code Construction

There are two important characteristics of a convolutional code when used with a sequential decoder, namely the undetected error probability in the decoder output and the distribution of computation. The latter arises from the fact that the amount of computation with sequential decoding is a random variable. The code parameters most closely related to computational performance are the "column distances" d_k, $k = 0, 1, \cdots, m$ where d_k is defined [11] as the minimum distance of the code of memory order k obtained by dropping all terms of degree greater than k from the original code generating polynomials. For instance, d_1 is the minimum distance of the code of memory order 1 with code generating polynomials $g_0^{(1)} + g_1^{(1)}D$ and $g_0^{(2)} + g_1^{(2)}D$. It is readily checked that $d_0 = 2$ is obtained if and only if $g_0^{(1)} = g_0^{(2)} = 1$, and that $d_1 = 3$ if and only if $d_0 = 2$ and the values of $g_1^{(1)}$ and $g_1^{(2)}$ are different. Simulations have shown that the amount of computation is unacceptably large if $d_1 < 3$, essentially because the distance between the upper and lower halves of the encoding tree [8] is not increasing rapidly enough to permit early rejection by the decoder of an incorrect hypothesis of i_0. These considerations suggest restricting the search for codes to be used with sequential decoding to those for which

$$g_0^{(1)} + g_1^{(1)}D = 1 \tag{16a}$$

and

$$g_0^{(2)} + g_1^{(2)}D = 1 + D. \tag{16b}$$

From (15), we see that the only quick-look-in codes satisfying (16) are those for $L = 1$, namely those for which

$$G^{(1)}(D) + G^{(2)}(D) = D. \tag{17}$$

Simulations of $R = 1/2$ convolutional codes have shown that while d_1 is the main determiner of good computational performance, the further column distances $d_2, d_3, \cdots$ should grow as rapidly as possible to minimize the need for long searches into the encoding tree before an incorrect hypothesis of i_0 is rejected. This suggests the desirablity of those quick-look-in codes satisfying (16) and (17) and constructed by choosing $g_2^{(1)}, g_3^{(1)}, \cdots, g_m^{(1)}$ (which by (17) coincide with $g_2^{(2)}, g_3^{(2)}, \cdots, g_m^{(2)}$) in order so that the choice of $g_k^{(1)}$ maximizes d_k. This does not yet uniquely specify the code, since either choice of $g_k^{(1)}$ will occasionally give the same d_k.

The parameter of the code most affecting the undetected error probability of the decoder output is the free distance d_{free}. It has been empirically observed that for a given d_m, a large d_{free} is associated with a high density of "ones" in the code generating polynomials. This suggests that in the preceding procedure the choice $g_k^{(1)} = 1$ should be made whenever either choice gives the same d_k. Note that this rule now uniquely specifies a quick-look-in code of memory order m which can be constructed by the following algorithm.

Algorithm 1:

Step 1: Choose $g_0^{(1)} = g_0^{(2)} = g_1^{(2)} = 1$, and $g_1^{(1)} = 0$. Set $d_1 = 3$ and set $k = 2$.

Step 2: Set $g_k^{(1)} = g_k^{(2)} = 0$ and compute d_k. If $d_k > d_{k-1}$, go to step 4.

Step 3: Set $g_k^{(1)} = g_k^{(2)} = 1$.

Step 4: If $k = m$, stop. Otherwise increase k by 1 and go to step 2.

Three comments about this algorithm are in order. First, there is no necessity to recompute d_k in step 3 since it can be shown if setting $g_k^{(1)} = g_k^{(2)} = 0$ does not cause d_k to exceed d_{k-1} then neither does setting $g_k^{(1)} = g_k^{(2)} = 1$. Secondly, if $d_k > d_{k-1}$ in step 2, then it must be that $d_k = d_{k-1} + 1$. This implies that d_k will be given exactly by adding 3 (the value of d_{k-1} the first time step 2 is executed) to the number of zeros among $g_2^{(1)}, g_3^{(1)}, \cdots, g_k^{(1)}$. These facts can be proved easily from consideration of the generator matrix [3] of the convolutional code. Thirdly, the only computationally involved part of the algorithm is the calculation of d_k in step 2. The simplest way to compute d_k is by a sequential decoding type search as suggested by Forney [12].

Algorithm 1 was programmed for the UNIVAC 1107 computer in the University of Notre Dame Computing Center with $m = 47$ (code constraint length $n_A = (m + 1)N = 96$ digits). The algorithm yielded the following generators:

$$G^{(1)} = [533,533,676,737,355,3]_8 \tag{18a}$$

and

$$G^{(2)} = [733,533,676,737,355,3]_8 \tag{18b}$$

where we have adopted the convention of specifying a polynomial by its sequence of binary coefficients written in octal. For example, $G^{(1)} = [53]_8$ denotes the polynomial $G^{(1)}(D) = 1 + D^2 + D^4 + D^5$ since 53 is octal for the binary sequence 101011.

It should be clear that the codes obtained from Algorithm 1 are "nested" in the sense that the code of memory order m', $m' < m$, can be obtained by dropping the terms of degree greater than m' from the code generating polynomials of the latter code. Thus (18) serves to specify all the codes with memory order $m = 47$ or less given

by Algorithm 1. For instance, the code with memory order $m = 35$ has the code generating polynomials

$$G^{(1)} = [533,533,676,737]_8 \tag{19a}$$

and

$$G^{(2)} = [733,533,676,737]_8. \tag{19b}$$

The code with $m = 31$ has been selected by the National Aeronautics and Space Administration (NASA) for use in the Pioneer F/G Jupiter fly-by mission. Since there are 8 zeros among $g_2^{(1)}, g_3^{(1)}, \cdots, g_{32}^{(1)}$, it follows that this code has minimum distance $d_{32} = 3 + 8 = 11$. Using the Jet Propulsion Laboratory (JPL) hardware sequential decoder to search for a minimum weight encoded sequence, Layland [13] verified that this code had free distance $d_{\text{free}} = 23$. This large value of free distance ensures extremely low undetected error probability when the code is sequentially decoded. This same code has also been selected by the German Institute for Space Research (GFW) for use in its HELIOS probe. The code with $m = 23$ is being used by NASA in the study phase of the Planetary Explorer program.

V. Performance With Sequential Decoding

To verify the effectiveness with sequential decoding of the quick-look-in codes of Algorithm 1, the code with memory order $m = 35$ was tested on several simulated channels together with the best known $m = 35$, $R = 1/2$, systematic code, namely the adjoint [14] of Forney's extension [12] of one of Bussgang's optimal codes. This systematic code has the code generating polynomials

$$G^{(1)} = [400,000,000,000]_8 \tag{20a}$$

and

$$G^{(2)} = [715,473,701,317]_8. \tag{20b}$$

The two codes were tested on both the binary symmetric channel and the additive white Gaussian noise (deep-space) channel as simulated on the UNIVAC 1107 computer which was also used to do the sequential decoding. Data were encoded into frames of 256 information bits followed by 35 zero bits to truncate the memory of the code. For the Gaussian channel, the output digits were quantized to 8 levels (3 bit quantization) in the manner suggested by Jacobs [1]. The Fano sequential decoding algorithm [7] was used for decoding and up to 50 000 computations were allowed to decode each frame. Frames requiring more than 50 000 computations were considered "erased" by the decoder. A computation was defined to be any "forward look" in the Fano algorithm. One thousand frames were decoded on each channel that was simulated.

Binary symmetric channels (BSC) with channel error probability p of 0.045 and 0.057 were simulated. For these values of p, the code rate $R = 1/2$ is equal to R_{comp} and $1.1 \times R_{\text{comp}}$, respectively, where R_{comp} is the computational cutoff rate of the sequential decoder [8]. The results of this simulation are given in Tables I and II. These results show little difference in the distribution of computation between the nonsystematic and the systematic code but show a dramatic difference in undetected error probability in the decoder output. In fact, no decoding errors whatsoever were committed with the nonsystematic code. For the noisier ($p = 0.057$) binary symmetric channel, it appears at first glance that the systematic code gives a reduced probability of large computational loads so that only 10 percent of the frames were erased compared to 25 percent for the nonsystematic code. It must be noted, however, that the undetected error probability was 10 percent for the systematic code. Thus, 15 percent of the frames are erased with the nonsystematic code but not with the systematic code, but the latter code gives incorrect decoding two-thirds of the time for these frames. In other words, the apparent improvement in computation is the result of decoding er-

TABLE I

Simulation Results for Decoding 1000 Frames of 256 Bits Each for the Nonsystematic Code of (19)

N	Fraction of Frames With Computation N or More: BSC $p = 0.057$	BSC $p = 0.045$	Gaussian $E_b/N_0 = 2$
400	1.000	0.991	0.968
550	0.949	0.785	...
600	...	...	0.676
850	0.802	0.477	0.445
1000	0.753	0.382	0.358
4000	...	...	0.070
5000	0.440	0.063	...
10 000	0.358	0.036	0.017
	Fraction of frames erased (50 000 or more computations)		
	0.249	0.008	0.005
	Fraction of frames decoded in error		
	0.000	0.000	0.000

TABLE II

Simulation Results for Decoding 1000 Frames of 256 Bits Each for the Systematic Code of (20)

N	Fraction of Frames with Computation N or More: BSC $p = 0.057$	BSC $p = 0.045$	Gaussian $E_b/N_0 = 2$
400	1.000	0.991	0.969
550	0.932	0.756	...
600	...	...	0.652
850	0.734	0.403	0.404
1000	0.673	0.320	0.327
1500	0.532	0.187	0.188
4000	...	...	0.060
5000	0.319	0.048	...
10 000	0.237	0.031	0.019
	Fraction of frames erased (50 000 or more computations)		
	0.108	0.004	0.004
	Fraction of frames decoded in error		
	0.087	0.002	0.000

roneously and is a "fools rush in where angels fear to tread" phenomenon.

Tables I and II also give the simulation results for the Gaussian channel with $E_b/N_0 = 2$ (3 dB) where E_b is the transmitted energy per information bit and N_0 is the one-sided noise power spectral density. No decoding errors were observed for either code, and the distributions of computation are virtually identical. The code rate $R = 1/2$ is equal to R_{comp} for the quantized channel. Comparison to the binary symmetric channel with $R = R_{\text{comp}}$ ($p = 0.045$) shows essentially identical performance on both channels.

The conclusions of these simulations are that the nonsystematic quick-look-in code gives comparable computational performance but gives much lower undetected error probability than the best systematic code of the same memory order. In the next section, we show the rather surprising fact that the encoder for the nonsystematic code is also easier to implement.

VI. Encoder Implementation

The obvious realization for an $R = 1/2$ convolutional encoder is shown in Fig. 1 and is seen to require

$$W[G^{(1)}(D)] + W[G^{(2)}(D)] - 2$$

two-input modulo-two adders as well as the necessary delay cells for storing the past m information bits. This realization of the encoder for the systematic code of (20) requires 21 modulo-two adders and is the customary encoding circuit for this code. For the quick-look-in nonsystematic code of (19), this realization of the encoder would require 53 modulo-two adders, this large number resulting from the fact that about three-fourths of the coefficients in the generators are ones.

A simple "trick" can be used to reduce the number of modulo-two adders need to implement a generator whose density of ones exceeds 50 percent, namely implement the binary complemenet of the desired generator together with a circuit that does the necessary complementation of the output. Letting

$$\bar{G}(D) = \bar{g}_0 + \bar{g}_1 D + \cdots + \bar{g}_m D^m$$

where

$$\bar{g}_i = g_i + 1$$

we may write

$$G(D) = \bar{G}(D) + 1 + D + \cdots + D^m$$

or

$$G(D) = \bar{G}(D) + (1 + D^{m+1})/(1 + D). \tag{21}$$

It is readily verified that the circuit in Fig. 2 realizes the transfer function $G(D)$ in the form (21) and requires

$$W[\bar{G}(D)] + 2 = m + 3 - W[G(D)] \tag{22}$$

two-input modulo-two adders. For example, this circuit realizes the transfer function $G^{(2)}(D)$ of (19b) with only 10 modulo-two adders.

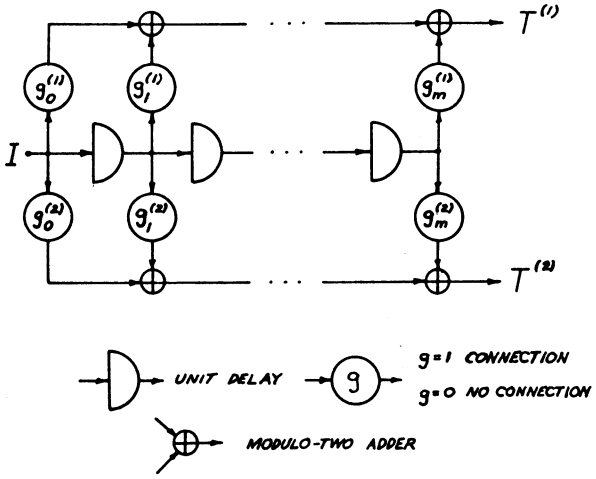

Fig. 1. Obvious realization of encoder for $R = ½$ convolutional code.

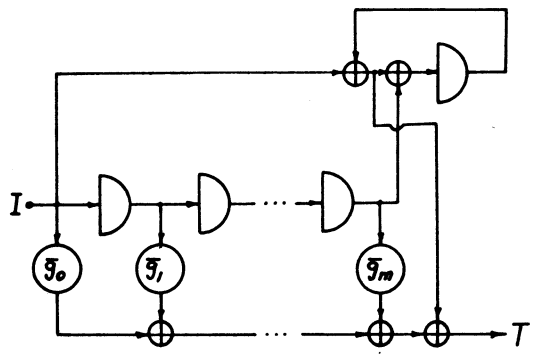

Fig. 2. Alternate realization of transfer function $G(D)$ of form suggested by (21).

By virtue of (1) and (17), we have for the code of (19) that

$$T^{(1)}(D) = DI(D) + T^{(2)}(D). \tag{23}$$

Hence if the circuit of Fig. 2 is used to realize $G^{(2)}(D)$ so that its input is $I(D)$ and its output $T^{(2)}(D)$, then $T^{(1)}(D)$ can be formed with one further modulo-two adder which adds the output of the first delay cell in the circuit of Fig. 2, namely $DI(D)$, to the circuit output $T^{(2)}(D)$. Hence the complete encoder for the quick-look-in nonsystematic code of (19) can be realized with only 11 two-input modulo-two adders.

If the circuit of Fig. 2 is used to realize $G^{(2)}(D)$ for the systematic code of (20), the complete encoder can be realized with only 16 modulo-two adders. This is a substantial reduction in the 21 adders needed for the encoder of Fig. 1, but is still surprisingly more than the 11 adders required in the encoder for the nonsystematic code. The obvious conclusion is that nonsystematic codes do not necessarily entail more complicated encoders than systematic codes of the same memory order.

VI. Theoretical Consideration: Gilbert Bound

Although the simulations reported previously confirm the practical value of quick-look-in nonsystematic codes, it may be reassuring to note that the constraints (16)

and (17), which define the quick-look-in codes suitable for sequential decoding, are compatible with the existence of codes with large minimum distance. We shall demonstrate this fact by showing that the class of codes satisfying (16) and (17) meet the same asymptotic Gilbert bound as the general class of rate $R = 1/2$ convolutional codes. We follow closely the derivation of the Gilbert bound given in [2].

For any D transform

$$P(D) = p_0 + p_1 D + p_2 D^2 + \cdots$$

we shall hereafter write $\{P(D)\}$ to denote the polynomial of degree m or less obtained by dropping all terms of degree greater than m from $P(D)$. With this convention, $\{T^{(1)}(D)\}$ and $\{T^{(2)}(D)\}$ are just the initial code word of a rate $R = 1/2$ convolutional code. From (23) it follows that

$$\{T^{(1)}(D)\} + \{T^{(2)}(D)\} = \{DI(D)\} \tag{24}$$

for quick-look-in codes, and hence that specification of an initial code word also specifies all digits in $\{I(D)\}$ except i_m. Thus only two choices of $\{I(D)\}$ are possible for a given initial code word. Next, it is easily shown [2, p. 14] that the specification of $\{T^{(j)}(D)\}$ and an $\{I(D)\}$ with $i_0 = 1$ uniquely determines $G^{(j)}(D)$. Hence only two quick-look-in codes can have a given initial code word in common and produced by information sequences with $i_0 = 1$.

Let d be the greatest minimum distance d_m for all codes of memory order m satisfying (16) and (17). Since an initial code word contains n_A positions, there are at most

$$N = \sum_{j=0}^{d} \binom{n_A}{j} \tag{25}$$

different initial code words of Hamming weight d or less. But no code has distance greater than d, and hence every code must have at least one initial code word of weight d or less resulting from some $\{I(D)\}$ with $i_0 = 1$. Noting that there are exactly 2^{m-1} codes which satisfy (16) and (17) since $g_2^{(1)}, g_3^{(1)}, \cdots, g_m^{(1)}$ may be selected arbitrarily, it must be true that

$$2N \geqq 2^{m-1} \tag{26}$$

since each initial code word appears in only two codes for $\{I(D)\}$ with $i_0 = 1$. Using the well-known [3] inequality

$$N \leqq 2^{n_A H(d/n_A)} \quad \text{for} \quad d/n_A \leqq 1/2 \tag{27}$$

where $H(x) = -x \log_2 x - (1 - x) \log_2 (1 - x)$ is the binary entropy function, the inequality (26) becomes

$$H(d/n_A) \geqq (m - 2)/n_A. \tag{28}$$

Noting that $n_A = 2(m + 1)$ for $R = 1/2$, it follows from (28) that

$$\lim_{n_A \to \infty} H(d/n_A) \geqq 1/2 = 1 - R \tag{29}$$

which is the usual asymptotic Gilbert bound on minimum distance for general convolutional codes of rate $R = 1/2$ [2].

We conclude that the class of quick-look-in codes satisfying (16) and (17) are "good" in the sense that there exist codes of this type with minimum distance at least as great as that guaranteed by the asymptotic Gilbert bound.

VIII. Conclusions

In this paper, it has been shown that the desirability of quick and reliable extraction of the information digits from the hard-decisioned received sequences of a convolutional code led naturally to the formulation of quick-look-in nonsystematic codes. An algorithm was given for the generation of a class of $R = 1/2$ quick-look-in codes designed especially for use with sequential decoding. It was noted that codes of this class have been chosen for several deep-space missions. The quality of these codes relative to the best systematic codes was verified by a simulation which showed comparable computational performance with sequential decoding but a much lower undetected error probability for the nonsystematic codes. It was shown further that the nonsystematic codes had simplier encoder realizations than the systematic codes. Finally, a proof was given that the quick-look-in constraint is compatible with the existence of codes whose minimum distance satisfies the asymptotic Gilbert bound.

Although not explicitly mentioned before, it should be noted that the small error amplification factor of quick-look-in codes is also advantageous in that it results in a small number of actual decoding errors during the "error events" when the sequential decoder has given incorrect estimates of the received sequence. For a full discussion of such error events, the reader is referred to [15, appendix II].

Acknowledgment

The authors wish to acknowledge that the idea for quick-look-in codes occurred independently to Dr. G. David Forney, Jr., of the Codex Corporation, Newton, Mass., although Dr. Forney formulated no specific codes of this type. The contributions of K. Vairavan, J. Brennan, J. Wruck, and J. Geist to the computer programs used in the simulations reported herein is gratefully acknowledged.

References

[1] I. M. Jacobs, "Sequential decoding for efficient communication from deep space," *IEEE Trans. Commun. Technol.* vol. COM-15, Aug. 1967, pp. 492–501.
[2] J. L. Massey, *Threshold Decoding.* Cambridge, Mass.: M. I. T. Press, 1963.
[3] J. M. Wozencraft and B. Reiffen, *Sequential Decoding.* Cambridge, Mass.: M. I. T. Press, 1961.
[4] E. A. Bucher and J. A. Heller, "Error probability bounds for systematic convolutional codes," *IEEE Trans. Inform. Theory,* vol. IT-16, Mar. 1970, pp. 219–224.
[5] D. J. Costello, Jr., "A strong lower bound on free distance for periodic convolutional codes," presented at the IEEE Int. Symp. Information Theory, Noordwijk, the Netherlands, June 1970.

[6] F. Jelinek and L. R. Bahl, "Maximum likelihood and sequential decoding of short constraint length convolutional codes," in *Proc. 7th Annu. Allerton Conf.*, Oct. 1969, pp. 130–139.
[7] R. M. Fano, "A heuristic discussion of probabilistic decoding," *IEEE Trans. Inform. Theory*, vol. IT-9, Apr. 1963, pp. 64–74.
[8] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. New York: Wiley, 1965, pp. 425–476.
[9] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, Apr. 1967, pp. 260–269.
[10] J. L. Massey and M. K. Sain, "Inverses of linear sequential circuits," *IEEE Trans. Comput.*, vol. C-17, Apr. 1968, pp. 330–337.
[11] D. J. Costello, Jr., "Construction of convolutional codes for sequential decoding," Ph.D. dissertation, Dep. Elec. Eng., Univ. Notre Dame, Notre Dame, Ind., Aug. 1969.
[12] G. D. Forney, Jr., "Use of a sequential decoder to analyze convolutional code structure," *IEEE Trans. Inform. Theory* (Corresp.), vol. IT-16, Nov. 1970, pp. 793–795.
[13] J. W. Layland, private communication, Mar. 17, 1970.
[14] J. J. Bussgang, "Some properties of binary convolutional code generators," *IEEE Trans. Inform. Theory*, vol. IT-11, Jan. 1965, pp. 90–100.
[15] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, Nov. 1970, pp. 720–738.

James L. Massey (S'54–M'55–F'71) was born in Wauseon, Ohio, on February 11, 1934. He received the B.S. degree in electrical engineering from the University of Notre Dame, Notre Dame, Ind., in 1956 and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1960 and 1962, respectively.

From 1956 to 1959 he was a Communications Officer in the United States Marine Corps. Since 1962 he has been on the faculty of the University of Notre Dame, where he is now Professor of Electrical Engineering; his research interests have been in the areas of coding and finite-state automata. He is a Consultant to the Codex Corporation, Newton, Mass., which markets error-correction equipment for which he has been granted patents. During 1966–1967 he was a Visiting Associate Professor of Electrical Engineering at the Massachusetts Institute of Technology.

Dr. Massey received the 1963 Best Tutorial Paper Award from the National Electronics Conference and the 1964 Paper award of the IEEE Group on Information Theory. He is a member of the American Society for Engineering Education, Eta Kappa Nu, Sigma Xi, and Tau Beta Pi.

Daniel J. Costello, Jr. (S'67–M'69) was born in Seattle, Wash., on August 9, 1942. He received the B.S.E.E. degree from Seattle University, Seattle, Wash., in 1964, and the M.S.E.E. and Ph.D. degrees from the University of Notre Dame, Notre Dame, Ind., in 1966 and 1970, respectively. His graduate work was supported by both the National Science Foundation and the National Aeronautics and Space Administration. His Ph.D. thesis, "Construction of convolutional codes for sequential decoding," was in the area of coding theory.

Since September 1969 he has been Assistant Professor of Electrical Engineering at the Illinois Institute of Technology, Chicago, teaching courses in linear systems, communications, information theory, and coding. His research interests are in the area of coding for digital communication systems.

Dr. Costello is a member of Pi Mu Epsilon, Tau Beta Pi, and Sigma Xi and is presently Secretary of the Midwest Chapter of the IEEE professional group on information theory. He also serves as a referee for several professional journals.