June 5, 1969

QUARTERLY PROGRESS REPORT
CONVOLUTIONAL CODING TECHNIQUES
FOR DATA PROTECTION

NASA GRANT NGL-15-004-026

Submitted to: Flight Data Systems Branch
NASA Goddard Space Flight Center
Greenbelt, Md. 20771
ATT: Dr. Robert W. Rochelle (Code 710)

and: National Aeronautics and Space Administration
Washington, D. C. 20546
ATT: Miss Winnie M. Morgan (Code - USI)
Technical Reports Assistant

Principal Investigator: Dr. James L. Massey
Professor of Electrical Engineering
Univ. of Notre Dame
Notre Dame, Ind. 46556

Research Assistants: J. Chang
D. Colstello
J. Geist

Research Period Reported: Feb. 16, 1969 to May 15, 1969

1. Development of a Powerful, Easily-Implemented, Non-Systematic, Binary Convolutional Code of Rate 1/2 Suitable for Sequential Decoding

(a) Description of the Code

The requirements of the Flight Data Section at the NASA GSFC have indicated the need for short constraint length, R = 1/2, convolutional codes that will yield low error probability when decoded by sequential decoding. These considerations led to a search for a good non-systematic code since the "effective" constraint length of a non-systematic R = 1/2 code is about double that of the more usual systematic code. A further requirement is that the encoder should be simple--that is, that there should be a small number of inputs to the modulo-two adders used in the encoder. This requirement stems from the fact that the encoder is a hardware device in the space vehicle itself. This search led to the finding of the code described below which provides extremely low decoder error probability and can be encoded by a device of remarkable simplicity, requiring fewer modulo-two adders than the presently used systematic code of the same constraint length. The code has the further desirable feature that, although non-systematic, the information stream can be easily obtained from the encoded digits without the use of a decoder. This latter feature permits quick "look in" at engineering data by ground stations without decoding equipment.

For a general, rate 1/2, binary, convolutional code, the information sequence

$$I(D) = i_0 + i_1 D + i_2 D^2 + \ldots \tag{1}$$

is used to form two encoded sequences, $T_1(D)$ and $T_2(D)$, by the rules

$$T_1(D) = G_1(D)I(D) \tag{2}$$

$$T_2(D) = G_2(D)I(D).$$

The code is systematic if $G_1(D) = 1$, i.e. if $I(D)$ is itself the first encoded sequence.

The search for a good non-systematic code was limited to codes such that

$$G_1(D) = D + G_2(D). \tag{3}$$

With this constraint, we see from (2) that

$$T_1(D) + T_2(D) = DI(D) \tag{4}$$

so that simply by adding (modulo-two) the two encoded sequences together, one obtains the information sequence unaltered except for a delay of one time instant.

It has been observed from experience, that generators (i.e. the coefficients in the polynomials $G_1(D)$ and $G_2(D)$) with a high density of "ones" generally result in low error probability. As will be seen later, a density of "ones" well above one-half also leads to a simple encoder. For these reasons, a search was made to find a good code using the following algorithm:

Algorithm: (1) Set the first two coefficients in $G_1(D)$ equal to "ones" and set $k = 3$.

(2) Set the kth digit in $G_1(D)$ equal to "one"

unless setting to "zero" gives a greater minimum distance over the first k branches of the code tree.

(3) Increase k by 1 and go to (2).

Application of this algorithm in a computer program up to k = 48 yielded the following generators (coefficients shown in the usual octal form):

$$G_1 = (733,\ 533,\ 676,\ 737,\ 355,\ 3)_8$$
$$G_2 = (533,\ 533,\ 676,\ 737,\ 355,\ 3)_8 \tag{5}$$

The minimum distance of the full code is 16. Since the algorithm is "nested", truncation of the two generators at any k, $k \leq 48$, will yield a good code at that constraint length.

For purposes of testing, the code was truncated at k = 36 since this is a likely figure to be used in some application. Hence, the generators used in the test were:

$$G_1 = (733,\ 533,\ 676,\ 737)_8$$
$$G_2 = (533,\ 533,\ 676,\ 737)_8 \tag{6}$$

By computer search, it was determined that this code had a "minimum distance" (measured over the constraint length of 36 branches) of 11 and a "free distance" (minimum distance over the full code tree) of at least 17. The free distance has proved to be a better predictor of error probability that the minimum distance and this code has a high value of this parameter. The exact free distance, however, is not yet known. It should be noted that G_1 contains 28 "ones" out of 36 digits, an exceptionally high density of "ones."

(b) Implementation of the Encoder

The "trick" used to reduce modulo-two adder connections in the encoder when the generators have a high density of "ones" is to implement the complement of the generator plus adding a circuit whose effect is to complement the transfer function preceding it. This latter circuit can be simply built as shown in Fig. 1. At the output of the adder where Y(D) is formed, we have the equation

$$Y(D) = I(D) + D[Y(D) + D^M I(D)]$$

$$\text{or} \quad \frac{Y(D)}{I(D)} = \frac{1 + D^{M+1}}{1 + D} = 1 + D + D^2 + \ldots + D^M . \qquad (7)$$

Hence, if G(D) is a polynomial transfer function of degree M, a circuit whose transfer function is the complement of G(D), i.e.

$$G(D) + 1 + D + D^2 + \ldots + D^M$$

can be obtained by adding the output of the circuit in Fig. 1 to the output of the circuit whose transfer function is G(D). The M memory cells used to realize G(D) can be the same as those used in the circuit of Fig. 1 so that the total circuit can be built simply as shown in Fig. 2.

These considerations can now be used to develop an encoder for the code whose generators are given in (6). Taking G(D) as the complement of $G_1(D)$, we have in octal form

$$G = (044, 244, 101, 040)_8$$

$$\text{or} \quad G(D) = D^3 + D^6 + D^{10} + D^{12} + D^{15} + D^{20} + D^{26} + D^{30}. \qquad (8)$$

Upon taking $M = 35$ and taking $G(D)$ as in (8), it follows from the analysis of Fig. 2 that $G_1(D)$ as in (6) is the transfer function relating $T_1(D)$ to $I(D)$ in Fig. 3. Moreover, the transfer function relating $T_2(D)$ to $I(D)$ in Fig. 3 is just

$$D + G_1(D) = G_2(D)$$

so that the circuit in Fig. 3 is a complete encoder for the binary, $R = 1/2$, code with constraint length 36 branches as specified by the generators in (6).

The complete encoder uses only 11 two-input modulo-two adders, compared to 21 two-input modulo-two adders required for a tapped shift-register to implement the systematic code with the same constraint length that is presently utilized in the NASA GSFC convolutional coding systems. This code has the generators:

$$G1 = (400,000,000,000)_8$$
$$G2 = (715,\ 473,\ 701,\ 317)_8. \tag{9}$$

Since G2 has 22 "ones" among its 36 coefficients, the encoder for the systematic code could profitably be instrumented in the manner shown in Fig. 2. This would lead to an encoder with 16 two-input modulo-two adders, a considerable savings over the single tapped shift-register implementation but still considerably more than the 11 adders required for the non-systematic code. This advantage of the non-systematic code is quite surprising since one "intuitively" expects that a good non-systematic code would be harder to encode than a good systematic code with the same constraint length.

(b) Performance of the Code

The error probability and computation performance of the non-systematic code of (6) relative to the systematic code of (9) is given in Tables I to IV. Table I gives the performance of the codes on the additive Gaussian noise channel with an E_b/N_o (Energy per information bit to single-side noise power per Hertz ratio) of 2.0 (3 db). The performance is nearly identical for the two codes, with the systematic code having a very slight computational advantage. As will be seen, this advantage derives from the fact that the systematic code often decodes a frame in reasonably few computations when "prudence" demands a closer examination, i.e. the systematic code is considerably more prone to decoding errors.

This latter fact is brought out clearly in Tables II, III and IV which shows performance on successively worse binary symmetric channels (BSC's.) These BSC's are chosen so that the code rate $R = \frac{1}{2}$ represents 90%, 100% and 110% respectively of the computational cutoff rate (R_{comp}) of the channel. For the worst channel (Table IV), there were no decoding errors over 1000 decoded frames whereas nearly 10% of the same frames were decoded incorrectly when the systematic code was used.

Allowing a rather large (50,000 computations--a computation being defined as a "forward look" and requiring about 100 μsec on the UNIVAC 1107 computer) amount of computation before the attempt to decode each frame of 256 information bits is abandoned, it is remarkable that no decoding error has yet been made in any of the sequential decoding simulations using the non-systematic code of (6).

A good qualitative comparison of the non-systematic code of (6) to the systematic code of (9) can be obtained from Table IV which gives their performance on a very noisy BSC. There are 141 more frames out of 1000 frames which fail to decode (in 50,000 computations or less) for the non-systematic code. However, 87 frames are erroneously decoded with the systematic code compared to <u>none</u> for the non-systematic code. The conclusion is that the decoding terminated on the extra 141 frames with the systematic code by "decoding" when the decoded frame error probability was near 50%. Without trying to be flippant, one could term this a "Fools rush in where angels fear to tread" phenomenon that accounts for the computational advantage of the systematic code as a consequence of its greater proneness to decoding error. The non-systematic code emerges a clear winner in system performance as well as system complexity.

2. Free Distance of Convolutional Codes.

Prior work done under this grant has established the importance of the free distance, d_{free}, of convolutional codes when used with sequential decoding as a determiner of the decoder error probability.

Recent work by D. Costello has resulted in a "Gilbert-like" lower bound on the free distance attainable with periodic, time-varying convolutional codes. This work shows that surprisingly large free distances are attainable. For example, at $R = 1/2$, a free distance to constraint length ratio of at least 0.39 can be obtained. This compares to an ordinary minimum distance to constraint length ratio of at least 0.11 guaranteed by the usual Gilbert bound. As $R \to 1$, the ratio between these two bounds becomes infinite.

Costello's lower bound on d_{tree} has also been used to obtain an asymptotically tight bound on the error probability attainable with low rate codes on the BSC.

A technical report, now in preparation, will give complete details of this work.

3. Simulation of the Jelinek Sequential Decoding Algorithm

J. Geist has just completed the programming of the UNIVAC 1107 computer in the Univ. of Notre Dame Computing Center to simulate a sequential decoder employing the Jelinek decoding algorithm. This facility will be used in the next quarter to obtain detailed performance comparisons with the Fano algorithm. Preliminary results indicate that:

(a) The two algorithms require about the same decoding time when the code rate R is about 90% of R_{comp}. For lower rates, the Jelinek algorithm is slightly superior. For higher rates, the Fano algorithm becomes much superior.

(b) The time per computation of the Jelinek algorithm grows quadratically with the total number of computations required to decode the frame. The time per computation is fixed with the Fano algorithm.

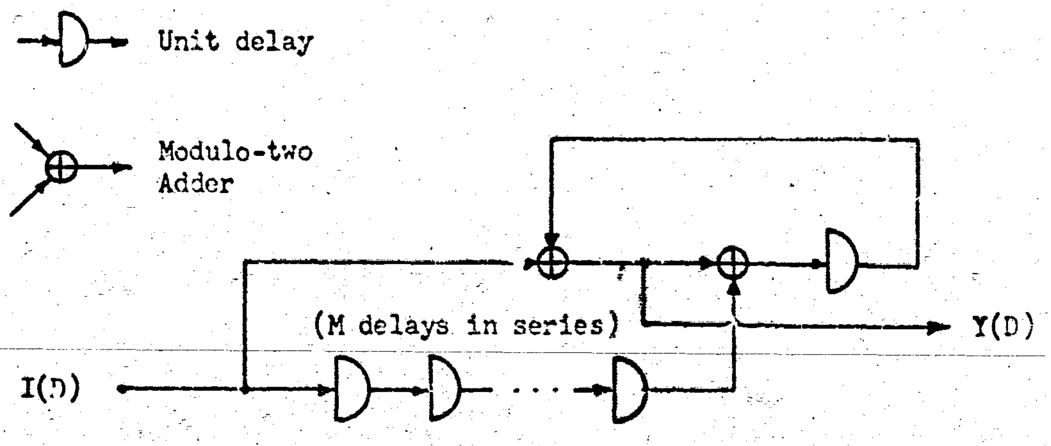

Fig. 1 Binary Linear Sequential Circuit with Transfer Function $1 + D + \ldots + D^M$

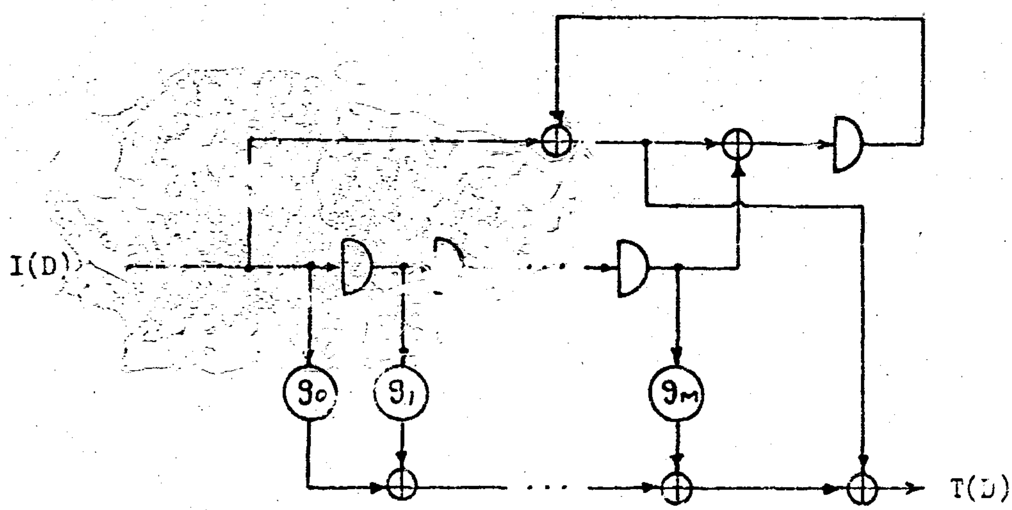

Fig. 2 Binary Linear Sequential Circuit with Transfer Function $G(D) + 1 + D + \ldots + D^M$ where $G(D) = g_0 + g_1 D + \ldots + g_M D^M$

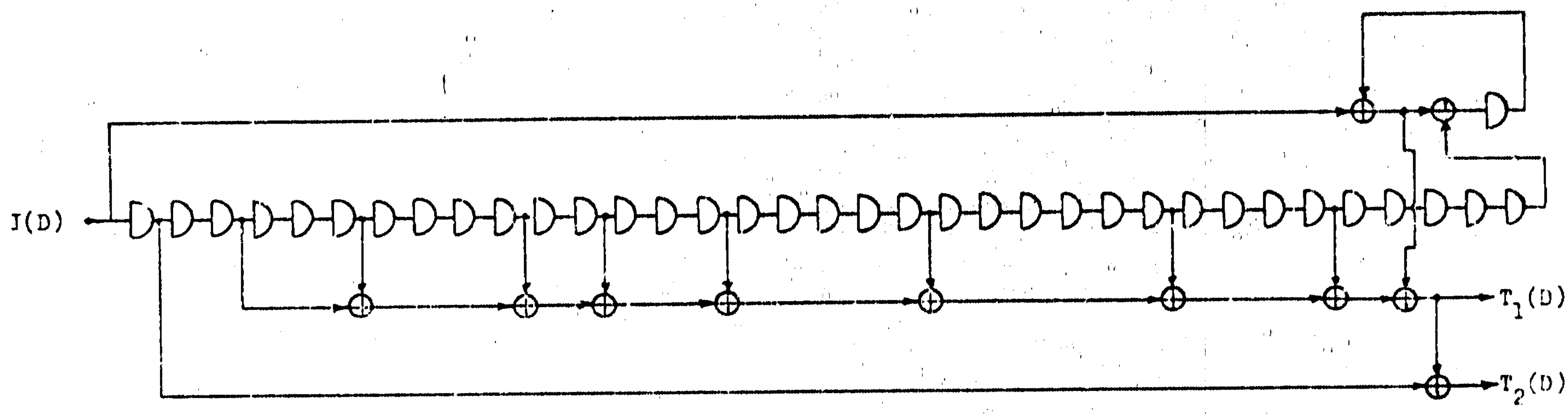

Fig. 3 Complete Encoder for the Non-Systematic Code Having

$$G_1 = (733,\ 533,\ 676,\ 737)_8$$

$$G_2 = (533,\ 533,\ 676,\ 737)_8$$

TABLE I: Performance on the Additive Gaussian Noise Channel with $E_b/N_o = 2.0$ (3 db). Results of Decoding [illegible] Frames of 256 Information Bits Each.

		Non-Systematic Code of Eq. ([illegible])	Systematic Code of Eq. ([illegible])
No. of frames with computation equal to or greater than the number shown in the first column	292	1000	1000
	400	968	969
	450	[illegible]	900
	500	835	810
	600	676	652
	700	567	523
	850	445	404
	1000	358	327
	1200	292	254
	1500	225	188
	4000	70	60
	10,000	17	19
	25,000	9	9
No. of erased frames (computations exceeding 50,000)		5	4
No. of frames resulting in decoding errors		0	0

TABLE II: Performance on the Binary Symmetric Channel with Crossover Probability 0.033 ($R = .9\ R_{comp}$). Results of Decoding 1000 Frames of 256 Information Bits Each

		Non-Systematic Code of Eq. (6)	Systematic Code of Eq. (9)
No. of frames with computation equal to or greater than the number shown in the first column	292	1000	1000
	400	883	870
	550	405	399
	700	223	195
	850	135	105
	1000	92	69
	1500	47	29
	2000	26	17
	2500	18	13
	5000	5	6
	10,000	2	2
	20,000	0	0
No. of erased frames (computation exceeding 50,000)		0	0
No. of frames resulting in decoding errors		0	0

TABLE III: Performance on the Binary Symmetric Channel with Crossover Probability 0.045 ($R = R_{comp}$).

Results of Decoding 1000 Frames of 256 Information Bits Each

		Non-Systematic Code of Eq. (6)	Systematic Code of Eq. (9)
No. of frames with computation equal to or greater than the number shown in the first column	292	1000	1000
	400	991	991
	550	785	756
	700	581	510
	850	477	403
	1000	382	320
	1500	240	187
	2000	167	138
	2500	134	104
	5000	63	48
	10,000	36	31
	20,000	23	11
No. of erased frames (computation exceeding 50,000)		8	4
No. of frames resulting in decoding errors		0	2

TABLE IV: Performance on the Binary Symmetric Channel with Crossover Probability 0.045 ($R = 1.1\ R_{comp}$).

Results of Decoding 1000 Frames of 256 Information Bits Each

		Non-Systematic Code of Eq. (6)	Systematic Code of Eq. (9)
No. of frames with computation equal to or greater than the number shown in the first column	292	1000	1000
	400	1000	1000
	550	949	932
	700	863	817
	850	802	734
	1000	753	673
	1500	640	532
	2000	585	455
	2500	543	412
	5000	440	319
	10,000	358	237
	20,000	303	181
No. of erased frames (computation exceeding 50,000)		249	108
No. of frames resulting in decoding errors		0	87