MAXIMUM LIKELIHOOD DECODING SCHEME FOR CONVOLUTIONAL CODES

WAI-HUNG NG, FRANK M. H. KIM and SATORU TASHIRO
The Boeing Aerospace Company
Seattle, Washington

Summary. In recent years the application of coding techniques to enchance digital data transmission has become widely accepted. In general, one would assume that a maximum likelihood decoding of convolutional codes would be impractical for long constraint length codes because the general approach of sequential decoding algorithms utilize very few properties of the code and hence require a considerable effort to decode the received data sequence.

In this paper, several structure and distance properties of the convolutional codes for different constraint lengths are derived and utilized in developing an efficient maximum likelihood decoding scheme. Under the proposed decoding threshold conditions, which are functions of the distance properties of the utilized codes, the required number of decoding operations can be reduced markedly. The analysis has been supported by computer simulations and by the development and testing of a prototype decoder. Key results are presented and discussed.

Introduction. The performance of convolutional codes has been recognized as being superior to block codes for real communication channels. Today, two primary decoding techniques are commonly used: optimum nonsequential decoding and sequential decoding. Nonsequential decoding is inefficient in obtaining the maximum coding gain because it is limited to relatively short constraint lengths. Sequential decoding, on the other hand, is subject to the variability in the number of computations required to successfully complete the decoding. The result is that buffer overflow occurs frequently, especially in the soft-decision sequential decoding process, unless extremely large buffers are utilized.

We are interested in the maximum likelihood decoding of the convolutional codes. Normally, one might consider that maximum likelihood decoding would require the construction of the entire branch value code tree followed by the selection of the path which has the minimum difference from the received code sequence. This procedure would be highly impractical for long constraint length codes because of the size of the code tree. However, we have found that the distance properties of the utilized codes are extremely useful in establishing the threshold conditions to eliminate unnecessary back-up searches and to locate the area where the decoding errors might be tentatively accepted. By applying the proposed threshold conditions, the decoding effort appears to be reduced remarkably.

In order to investigate the relationship between the coding gain and search length for various bit-error-rates (BER), discussions based on our limited computer simulation will be given. Also, a simple soft-decision permissible path decoder configuration, which is the basis of our prototype suboptimum docoder design, will be presented.

An Upper Bound on the Back-up Distance. The present study is an extension of the previous work [1]. The basic decoding strategy is that we always seek a code path at the minimum distance from the received sequence. We believe that when optimum threshold conditions are derived, maximum likelihood decoding will always require less decoding operations than general sequential decoding algorithms. The reason is that it has the highest probability of eliminating most of the unnecessary back-up searches by means of the minimum back-up distance requirement.

Let us define the distance function d(k) to be the minimum distance between two half-trees of length k branches of the selected code. For soft-decision decoding, each of the received code bits is quantized into Q levels and therefore the distance function d(k) could be considered to be extended to (Q-1)d(k). To simplify the exposition, we adopt the following notation in this paper:

- $\underset{\sim}{u}$ = transmitted sequence, it is a path in the code tree that is encoded from the message sequence.
- $\underset{\sim}{v}$ = received sequence, which differs from $\underset{\sim}{u}$ because of channel noise.
- $\underset{\sim}{w}$ = tentatively decoded sequence, a path in the code tree being compared with the received sequence $\underset{\sim}{v}$.
- $\underset{\sim}{t} = \underset{\sim}{w} \oplus \underset{\sim}{v}$ = test-error sequence, $|\underset{\sim}{t}|$ is the weight of $\underset{\sim}{t}$.

For any sequence, say $\underset{\sim}{w}$, we let $\underset{\sim}{w}_1$ be the last segment of $\underset{\sim}{w}$.

Now, we are going to derive an upper bound on the back-up distance for both hard-decision and soft-decision maximum likelihood decoding. Assume that a tentatively decoded sequence $\underset{\sim}{w}$ is obtained by a single branch forward extension of a decoded sequence which is at a minimum distance from the received sequence $\underset{\sim}{v}$; if there is another code path $\underset{\sim}{w}'$ of the same length whose distance to $\underset{\sim}{v}$ is a minimum for paths of its length, then $\underset{\sim}{w}'$ is diverging from $\underset{\sim}{w}$ at most b_t branches back where b_t is the maximum positive integer i such that $(Q-1)d(i) \leq 2\,|\underset{\sim}{t}|$. By applying this boundary condition with additional constraints [2], we can significantly reduce the decoding operation, especially for soft-decision decoding of convolutional codes having long constraint lengths.

Permissible Path Decoding. In our study, permissible path decoding will be utilized to eliminate all the shorter back-up searches. Here, we will briefly review the basic concept, and use a simple example to show how it could be applied.

In order to simplify the explanation, the discussion will be limited to the hard-decision decoding of a rate 1/3 code. We may define a permissible path $\underset{\sim}{P}$ to be a truncated path selected from the lower-half initial code tree such that it satisfies the following conditions: (i). $|\underset{\sim}{P}|$ is odd, (ii). $|\underset{\sim}{P}_1| = 1$, (iii). others (refer to [1]). Let us refer to an initial code tree as given in Figure 1. By examining the first 3 branches in the lower-half tree we note that there is only one path which satisfies the conditions just mentioned above. We call this path a permissible path $\underset{\sim}{P}$ and put it in the memory, where $\underset{\sim}{P}$ = 111 010 001. Now, let $\underset{\sim}{u}$ be the transmitted sequence and $\underset{\sim}{v}$ be the received sequence which contains a burst of two channel errors due to noise. By using a simple branch-by-branch forward search to determine a tentatively decoded sequence $\underset{\sim}{w}$, we note that there is a decoding error which results in a wrong decoded sequence. Thereafter, $|\underset{\sim}{t}|$ starts to increase, as shown in Figure 1. After we have decoded the first 4 segments and obtained $|\underset{\sim}{t}| = 3$, we question that a $\underset{\sim}{w}'$ exists that has a $|\underset{\sim}{t}'| = |\underset{\sim}{w}' \oplus \underset{\sim}{v}| < 3$. This can be easily examined by applying a test to the permissible path $\underset{\sim}{P}$, which has been previously stored in the memory. By using the modulo-2 addition, we have $|\underset{\sim}{t}'| = |\underset{\sim}{t} \oplus \underset{\sim}{P}| = 2 < |\underset{\sim}{t}|$, which implies that there is a $\underset{\sim}{w}'$ which has a smaller test-error weight than $\underset{\sim}{w}$. Hence, by applying the same $\underset{\sim}{P}$ we can derive that $\underset{\sim}{w}' = \underset{\sim}{w} \oplus \underset{\sim}{P} = \underset{\sim}{u}$, and recover to the correct path. This simple example represents the basic operation of the permissible path decoding; further detailed discussion should be directly referred to [1].

Distribution of Test-error Weight for Suboptimum Decoding. - From the discussion described in the last section, we will note that if there is a $\underset{\sim}{t}'$ such that $|\underset{\sim}{t}'| < |\underset{\sim}{t}|$, it implies that the maximum value of $|\underset{\sim}{P}|$ should be $|\underset{\sim}{P}|_{max} < 2|\underset{\sim}{t}|$ since $|\underset{\sim}{t}| > |\underset{\sim}{t}'| = |\underset{\sim}{t} \oplus \underset{\sim}{P}| \geqq |\underset{\sim}{P}| - |\underset{\sim}{t}|$. When the decoding is based only on the relative search length (bounded maximum back-up distance to make a decoding decision), the distribution of the test-error weight must be analyzed in order to reduce the number of required permissible paths that are a direct function of $|\underset{\sim}{P}|$.

We have carried out a computer simulation study of the distribution of the test-error weight for suboptimum decoding. The utilized code parameters are: V = 3, K = L = 16, where 1/V is the code rate, and K and L are constraint length and search length in segments, respectively. The code is generated by a single-generator sequence $\underset{\sim}{g}$ = 7112122201101011 (in octal). The decoding strategy is to search a path in the corresponding truncated code tree such that it is at minimum distance from the received sequence over the search length. We know that under the branch-by-branch search routine, the maximum test-error weight of each decoded branch for hard-decision decoding of a rate 1/3 code will always be less than or equal to 1. Therefore, the maximum value of $|\underset{\sim}{t}|$ will be $|\underset{\sim}{t}|_{max} = L = 16$.

A computer simulation study was performed on the Univac 1108 computer [3], with crossover probabilities of the binary symmetric channel equal to $p = 7 \times 10^{-2}$ and 9.5×10^{-2}, respectively. It is interesting to find that the maximum value of $|\underset{\sim}{t}|$ is less than 13. The detailed distribution of $|\underset{\sim}{t}|$ is given in Table 1 for reference. This simulation study proves that the number of permissible paths can be reduced by simply limiting the weight condition on $\underset{\sim}{P}$. Further discussion is given in the next section.

General Suboptimum Decoding Procedure. Today, communication engineers are also interested in the relatively simple decoder with medium coding performance. One way of simplifying the decoder design is to reduce the search length requirement for the suboptimum decoding of the convolutional codes. Let us consider that the total number of decoding errors N to be the product of the number of decoding error bursts N_B and the average number of decoding errors N_A in one burst. In order to minimize N_B , we should maximize the minimum distance between two-half trees. Minimizing N_A could be interpreted as minimizing the recovery errors after the first decoding error is being accepted. According to [4], the value of N_A could be minimized by either utilizing the systematic code with minimum weight generator sequence or by using nonsystematic code with a longer search length. For a relative short search length, it is almost impossible to minimize the values of N_A and N_B at the same time. Hence, it is necessary to select the code parameters for the specified BER region where the maximum coding gain is desired.

The general procedure, logical statements, and conditions for sub-optimum permissible decoding are discussed as follows:

(a). Whenever the decoder finds a $\underset{\sim}{w}$ which has minimum $|\underset{\sim}{t}|$ from $\underset{\sim}{v}$ over L segments of the truncated tree, the first segment of $\underset{\sim}{w}$ will be shifted out and a new $\underset{\sim}{w}_1$ will be shifted into the decoder. Hence, the first (L-1) segments of $\underset{\sim}{w}$ will always have a minimum test-error weight from the corresponding (L-1) segments of $\underset{\sim}{v}$.

(b). If newly determined $\underset{\sim}{w}_1$ results in a zero test-error weight, $|\underset{\sim}{t}_1| = 0$, then $\underset{\sim}{w}$ is still at minimum distance from $\underset{\sim}{v}$.

(c). For the V = 3 code with hard-decision decoding: If $|\underset{\sim}{t}_1| \neq 0$, it implies $|\underset{\sim}{t}_1| = 1$ (or $\underset{\sim}{t}_1$ has to be either equal to 001, 010, or 100). From (a), if there is a $\underset{\sim}{t}'$ such that $|\underset{\sim}{t}'| = |\underset{\sim}{t} \oplus \underset{\sim}{P}| < |\underset{\sim}{t}|$, it implies $|\underset{\sim}{t}'| = |\underset{\sim}{t}| - 1$ and $|\underset{\sim}{t}'_1| = 0$. The reason is that $\underset{\sim}{t}' = \underset{\sim}{t} \oplus \underset{\sim}{P}$ means $\underset{\sim}{t}'_1 = \underset{\sim}{t}_1 \oplus \underset{\sim}{P}_1$. Then $|\underset{\sim}{t}'| < |\underset{\sim}{t}|$ if and only if $\underset{\sim}{t}_1 = \underset{\sim}{P}_1$ such that $|\underset{\sim}{t}'_1| = 0$ and also $|\underset{\sim}{t}'| = |\underset{\sim}{t}| - 1$. Hence, when $|\underset{\sim}{t}_1| \neq 0$, we only need to use those $\underset{\sim}{P}$ with $\underset{\sim}{P}_1 = \underset{\sim}{t}_1$ to search for a $\underset{\sim}{t}'$.

(d). For $V = 3$ with soft-decision ($Q > 2$) decoding: Consider the $Q = 8$ case. The branch-by-branch search could result in $|\underset{\sim}{t}_1| \leqq 10$ since the $V = 3$ cases could be considered as $V = 3x(Q - 1) = 21$ code for the $Q = 8$ soft-decision case. The possible values of $\underset{\sim}{P}_1$ for $|\underset{\sim}{t}_1| \neq 0$ will extend to 001, 010, 100, 011, 101, 110 with the hard-decision ($Q = 2$) version. In order to simplify the computer simulation, whenever $|\underset{\sim}{t}_1| \neq 0$, we will apply those $\underset{\sim}{P}$ with $\underset{\sim}{P}_1 = \underset{\sim}{t}_1$ in hard-decision sense to search for a $|\underset{\sim}{t}'| < |\underset{\sim}{t} \oplus \underset{\sim}{P}|$ in the soft-decision sense. For example: If $\underset{\sim}{t}_1$ = $\tilde{0}$xx $\underset{\sim}{1}$xx $\tilde{0}$xx (in octal), the conditions on $\underset{\sim}{P}$ for searching $\underset{\sim}{t}'$ will be (i). $|\underset{\sim}{P}| < 2\ |\underset{\sim}{t}|$ in soft-decision sense, (ii). $\underset{\sim}{P}_1 = 010$ in hard-decision sense.

The distribution of permissible path $\underset{\sim}{P}$ for an initial tree generated by the generator sequence $\underset{\sim}{g} = 7112111101101$ is given in Table 2. After studying the simulation results on the distribution of the test-error weight presented in the last section, we wondered if it was possible to achieve further reductions in the number of permissible paths by simply limiting the maximum weight on $\underset{\sim}{P}$. Simulation results are given in Table 3, 4, and 5. They show that the effect on permissible path reduction is quite negligible even when the maximum weight of $\underset{\sim}{P}$ to be $|\underset{\sim}{P}|_{max} = d(L)$, especially in the lower BER region. Further simulations of search length effect on $V = 3$, $Q = 8$ suboptimum decoding are summarized as follows:

(1) Effect of different constraint lengths: two decoding simulations were performed on systematic codes generated by $\underset{\sim}{g}_1 = 7112122$ (K=7) and $\underset{\sim}{g}_2 = 7112122011$ (K=10) with search length $L = 10$ and $|\underset{\sim}{P}|_{max} = d(L)$. Simulation results showed that the code generated by $\underset{\sim}{g}_1$ performed better at BER $\cong 5x10^{-3}$ (approximately 0.5 dB coding gain difference) than $\underset{\sim}{g}_2$, that the crossover point occurred at BER $\cong 2.5x10^{-4}$, and that the code generated by $\underset{\sim}{g}_1$ appeared worse than $\underset{\sim}{g}_2$ at BER $\cong 10^{-5}$ (approximately 0.5 dB coding gain difference). All BER refers to decoded BER.

(2) Comparison of systematic and nonsystematic codes: three decoding processes were simulated on codes generated by $\underset{\sim}{g}_3 = 7125$ (non-systematic code, K = 4, $|\underset{\sim}{g}| = 7$), $\underset{\sim}{g}_4 = 71221$ (systematic code, K = 5, $|\underset{\sim}{g}| = 7$) and $\underset{\sim}{g}_5 = 71226$ (non-systematic code, K = 5, $|\underset{\sim}{g}| = 8$), with search length $L = 7$ and no path reduction. Simulation results showed that $\underset{\sim}{g}_3$ produces the best results, $\underset{\sim}{g}_4$ was second, and $\underset{\sim}{g}_5$ was the poorest (the computer simulation study was limited to BER $\geqq 10^{-4}$ region).

(3) Effect of search length on non-systematic codes: two decoding processes were computed for the code generated by $\underset{\sim}{g}_3 = 7125$ with search lengths $L = 7$ and $L = 10$, respectively. Simulation results showed that the $L = 10$ decoding performance was better than that for $L = 7$ by 0.5 dB in the BER region between $3x10^{-3}$ to $5x10^{-5}$. This indicates that when the non-systematic code is applied to suboptimum decoding, the minimun search length should be at least twice the constraint length.

Prototype Soft-Decision Permissible Path Decoder. An engineering model of a rate 1/3, soft-decision Q = 8, constraint length K = 10, search length L = 10, systematic permissible path convolutional decoder has been built and tested [5,6]. The purpose of the development was to determine the coding performance and to use the hardware as a design tool as well as a demonstration device. During the development, emphasis was placed on design simplicity, checkout convenience, and the use of readily available low cost IC. Secondary factors were parts count minimization and data rate.

The resultant design utilizes standard 14 and 16 pin TTL IC mounted on a 180 socket Scanby circuit board which in turn was mounted on a standard laboratory chassis, as shown in Figure 2. In addition to the decoder, the circuit board contains the transmitting terminal with its data generator and convolutional encoder. It also contains circuits for testing the decoder both functionally and in terms of BER performance. The front panel has I/O terminals, control switches, and a set of eight LED octal number readouts for monitoring key operations in the decoder during the functional (checkout) tests. The emphasis on design simplicity, plus the algorithm's inherent simplicity, has resulted in the use of extremely simple routines and timing architecture. All wiring on the Scanby board, with the exception of test circuits, have been wirewrapped on a semi-automatic machine. The time required to program the machine was estimated to be far less than that required to hand wirewrap and to correct wiring errors.

Coding efficiency tests were conducted with a setup as shown in Figure 3. The noisy channel is represented by a video summer, located external to the chassis, which adds Gaussian noise to the transmitted encoded data. The noisy signal is 3 bit quantized in the chassis-mounted A/D converter. The tests measure the decoded BER as a function of the error rate of the most significant bit of the A/D output. The theoretical coherent PSK BER performance curve was then used to relate channel BER, decoded BER, and signal to noise ratios. The coding gain at 10^{-3}, 10^{-4}, and 10^{-5} BER has been determined to be 2.5, 3.4, and 4.1 dB, respectively.

Other performance characteristics of the decoder include a maximum data rate of 150 Kb/s and a through-put delay of only 11 data bits. The maximum data rate can be readily increased by a factor of 10 or more by including a data buffer to take advantage of the infrequency of the back-up search as well as the fact that most of back-searches are far shorter than the upper bound (refer to the back-up reduction operation in [1] and threshold conditions in [2]). The through-put delay, however, will be lengthened by the use of a data buffer. A detailed discussion of the prototype design and its modifications will be presented in a later paper.

Conclusion. In this paper we have presented the basic concept of maximum likelihood decoding for convolutional codes. Some limited computer simulation results and a prototype permissible path decoder using suboptimum decoding were included and discussed. The development of the prototype has demonstrated that high coding gains can be achieved in a relatively simple decoder. It should be noted that the decoder can be made with significantly fewer parts through better parts selection, by deletion of many of the built-in test equipment, and by further optimization of the code parameters. At present, computer simulation studies are underway with emphasis on rate 1/2 codes. Further results are anticipated in the near future.

Acknowledgement. The authors deeply appreciate the important contributions of H. Peter Thomas and Frank J. Kato to the prototype development.

References.

[1] W-H. Ng and P. E. Pfeiffer, “Minimum-Hamming-Distance Decoding of Single Generator Binary Convolutional Codes”, Information and Control, vol. 13-4, pp. 295-315, October 1968.

[2] W-H. Ng, “An Upper Bound on the Back-Up Depth for Maximum Likelihood Decoding of Convolutional Codes”, IEEE Trans. on Information Theory, (to be published May 1976).

[3] W-H. Ng and Y. S. Kuo, “Soft-Decision and Permissible Path Reduction for the Ng-Pfeiffer Decoding Algorithm”, Lockheed Electronics, Inc., LEC-642D-823245, November 1970.

[4] W-H. Ng, “Study on Decoding Recovery Behavior for Convolutional Codes”, IEEE Trans. on Information Theory, vol. IT-16, pp 795-797, November 1970.

[5] F.M.H. Kim, “Development and Testing of Convolutional Decoders”, Boeing Aerospace Company No. 2-5130-0000-133, April 1975.

[6] F.M.H. Kim, “SID TDMA Error Correction Coding/Decoding Design Criteria, Techniques, and Implementation”, Boeing Aerospace Company, No. 2-5130-10-091, November 1973.

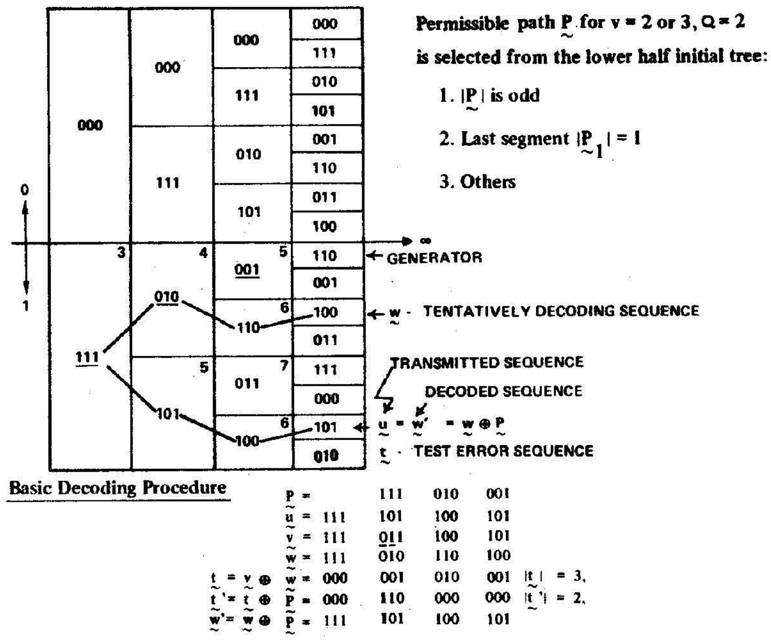

FIGURE 1 - INITIAL CODE TREE AND PERMISSIBLE PATH DECODING

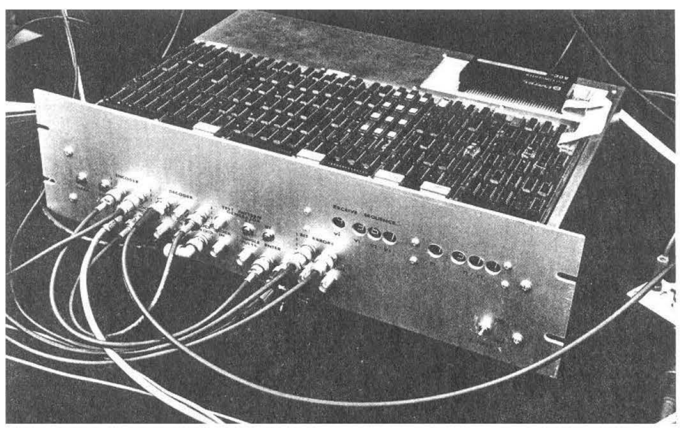

FIGURE 2 - PROTOTYPE PERMISSIBLE PATH DECODER

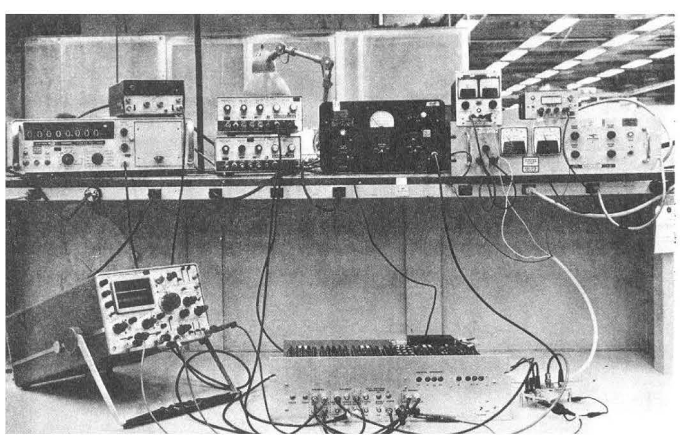

FIGURE 3 - DECODER TEST SET-UP

Number of $|\underset{\sim}{t}|$ in each record for crossover probabilities p

Case 1. $p = 7\times10^{-2}$

Record No.	$\lvert\underset{\sim}{t}\rvert = 8$	9	10	11	12	≥ 13
1	80	24	5	1	0	0
2	68	11	3	0	0	0
3	51	10	4	0	0	0
4	99	32	14	0	0	0
5	82	10	1	0	0	0
6	117	33	5	0	0	0
7	85	23	13	3	1	0
8	113	33	19	2	0	0
9	85	33	12	2	0	0
10	55	8	0	0	0	0
Total	835	217	76	8	1	0

Case 2. $p = 9.5\times10^{-2}$

Record No.	$\lvert\underset{\sim}{t}\rvert = 8$	9	10	11	12	≥ 13
1	414	148	46	6	0	0
2	391	131	34	0	0	0
3	361	190	66	17	1	0
4	438	181	46	5	0	0
5	405	172	40	0	0	0
6	462	236	87	6	0	0
7	414	163	39	0	0	0
8	477	228	76	8	0	0
9	502	237	97	11	0	0
10	414	209	93	8	0	0
Total	4278	1895	624	61	1	0

Table 1. Weight distribution of test-error sequence $|\underset{\sim}{t}| \geq 8$ for $V = 3$, $K = L = 16$, $Q = 2$ suboptimum decoding where $\underset{\sim}{g} = 7112122201101011$. Each record used to prepare the tabulation above consisted of 10,800 message digits.

Last segment $\underset{\sim}{P}_1$ in the permissible path $\underset{\sim}{P}$							
$\|\underset{\sim}{P}\|$	Total Number of Paths	001	010	011	100	101	110
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	1	1	0	0	0	0	0
6	4	0	1	0	1	0	1
7	9	1	1	1	1	1	0
8	17	1	2	1	2	1	1
9	31	2	4	2	3	2	1
10	54	4	5	3	5	4	2
11	92	7	8	5	9	5	4
12	156	19	8	9	13	8	7
13	263	22	25	13	20	8	19
14	441	29	44	20	38	25	22
15	716	55	51	38	58	44	29
16	1108	77	75	58	76	51	55
17	1636	109	93	76	98	75	77
18	2281	121	114	98	110	93	109
19	3000	117	133	110	124	114	121
20	3739	118	125	124	122	133	117
21	4417	96	107	122	110	125	118
22	4985	87	79	110	89	107	96
23	5425	72	56	89	57	79	87
24	5733	38	43	57	42	56	72
25	5931	25	27	42	23	43	38
26	6047	15	13	23	13	27	25
27	6106	4	7	13	7	13	15
28	6130	2	3	7	1	7	4
29	6139	1	0	1	2	3	2
30	6142	0	0	2	0	0	1
31	6142	0	0	0	0	0	0

Table 2. Weight distribution of permissible paths $\underset{\sim}{P}$ for $V = 3$, $K = 13$, $Q = 8$ where $\underset{\sim}{g} = 711212220\bar{1}101$

P_e In - Channel	P_e Out - Information	Remark
3.991×10^{-2}	4.62×10^{-5}	A
3.991×10^{-2}	4.62×10^{-5}	B
3.991×10^{-2}	4.62×10^{-5}	C
3.991×10^{-2}	4.62×10^{-5}	D
4.479×10^{-2}	1.02×10^{-4}	A
4.479×10^{-2}	1.02×10^{-4}	B
4.479×10^{-2}	1.02×10^{-4}	C
4.479×10^{-2}	1.02×10^{-4}	D
4.953×10^{-2}	2.51×10^{-4}	A
4.953×10^{-2}	1.39×10^{-4}	B
4.953×10^{-2}	1.39×10^{-4}	C
4.953×10^{-2}	1.39×10^{-4}	D
5.954×10^{-2}	5.82×10^{-4}	A
5.954×10^{-2}	3.89×10^{-4}	B
5.954×10^{-2}	3.62×10^{-4}	C
5.954×10^{-2}	3.62×10^{-4}	D
6.958×10^{-2}	1.48×10^{-3}	A
6.958×10^{-2}	1.46×10^{-3}	B
6.958×10^{-2}	9.53×10^{-4}	C
6.958×10^{-2}	9.53×10^{-4}	D
7.964×10^{-2}	5.01×10^{-3}	A
7.964×10^{-2}	4.02×10^{-3}	B
7.964×10^{-2}	3.34×10^{-3}	C
7.964×10^{-2}	3.31×10^{-3}	D
8.963×10^{-2}	9.90×10^{-3}	A
8.963×10^{-2}	8.47×10^{-3}	B
8.963×10^{-2}	7.09×10^{-3}	C
8.963×10^{-2}	7.04×10^{-3}	D
9.965×10^{-2}	2.02×10^{-2}	A
9.965×10^{-2}	1.48×10^{-2}	B
9.965×10^{-2}	1.37×10^{-2}	C
9.965×10^{-2}	1.27×10^{-2}	D

Each run includes 108,000 information digits

A : $|\underset{\sim}{P}| \leq 15$, total 284 paths
B : $|\underset{\sim}{P}| \leq 17$, total 764 paths
C : $|\underset{\sim}{P}| \leq 19$, total 1846 paths
D : $|\underset{\sim}{P}| < \infty$, total 9420 paths

Table 3. Simulation results of permissible path decoding with path reduction , where the code parameters are : V = 3, K = L = 16, Q = 2, $\underset{\sim}{g}$ = 7112122201101011

Each run includes 108,000 information digits

P_e In - Channel	P_e Out - Information	No. of Decoding Errors	Remark
5.985×10^{-2}	0	0	A
5.985×10^{-2}	1.89×10^{-5}	2	B,C,D,E
6.982×10^{-2}	1.89×10^{-5}	2	A
6.982×10^{-2}	7.41×10^{-5}	8	B
6.982×10^{-2}	4.63×10^{-5}	5	C,D,E
8.987×10^{-2}	8.06×10^{-4}	87	A
8.987×10^{-2}	5.83×10^{-4}	63	B
8.987×10^{-2}	5.28×10^{-4}	57	C
8.987×10^{-2}	5.09×10^{-4}	55	D,E
10.926×10^{-2}	4.18×10^{-3}	451	A
10.926×10^{-2}	3.66×10^{-3}	395	B
10.926×10^{-2}	3.93×10^{-3}	424	C
10.926×10^{-2}	3.94×10^{-3}	425	D,E

A : $|\underset{\sim}{P}| \leqq 11$, total 92 paths
B : $|\underset{\sim}{P}| \leqq 13$, total 229 paths
C : $|\underset{\sim}{P}| \leqq 15$, total 422 paths
D : $|\underset{\sim}{P}| \leqq 17$, total 602 paths
E : $|\underset{\sim}{P}| < \infty$, total 766 paths

Table 4. Simulation results of permissible path decoding with path reduction. The code parameters are : V = 3, K = L = 10, Q = 8, $\underset{\sim}{g}$ = 7112122201

Each run includes 108,000 information digits

V/K	Q	$\underset{\sim}{g}$	P_e - In Channel	P_e - Out Information	No. of Decoding Errors	Remark
3/13	8	$\underset{\sim}{g}_a$	7.978×10^{-2}	0	0	A
3/13	8	$\underset{\sim}{g}_a$	8.987×10^{-2}	2.50×10^{-4}	27	A
3/13	8	$\underset{\sim}{g}_a$	10.926×10^{-2}	3.85×10^{-3}	416	A
3/13	8	$\underset{\sim}{g}_a$	10.926×10^{-2}	1.80×10^{-3}	194	B
3/16	8	$\underset{\sim}{g}_b$	9.971×10^{-2}	7.89×10^{-4}	85	C
3/16	8	$\underset{\sim}{g}_b$	11.974×10^{-2}	3.72×10^{-3}	375	C

A : $|\underset{\sim}{P}| \leqq 13$, total 263 paths
B : $|\underset{\sim}{P}| \leqq 15$, total 716 paths
C : $|\underset{\sim}{P}| \leqq 15$, total 759 paths

Table 5. Simulation results of permissible path decoding (K = L) where $\underset{\sim}{g}_a$ = 7112122201101
$\underset{\sim}{g}_b$ = 7112122201101011